

1

Elementos, estructura y funciones de un sistema operativo

OBJETIVOS DEL CAPÍTULO

- ✓ Conocer qué es un sistema informático.
- ✓ Trabajar con los distintos sistemas de representación.
- ✓ Saber qué es un sistema operativo.
- ✓ Ver los elementos y estructura de un sistema operativo.
- ✓ Conocer las funciones de un sistema operativo.
- ✓ Identificar los procesos y sus estados.
- ✓ Ver los distintos tipos de sistemas operativos.
- ✓ Conocer algunos de los sistemas operativos actuales para microordenadores.
- ✓ Conocer cómo arrancar y parar el sistema operativo.

1.1 EL SISTEMA INFORMÁTICO

La palabra **informática** proviene de la contracción de los vocablos **INFOR**mación y auto**MÁTICA**. El diccionario de la Real Academia Española la define como “*el conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de ordenadores*”.

La palabra **ordenador**, **computador** o **computadora**, que comúnmente se utiliza para referirse a las máquinas informáticas, está definida en el diccionario de esta forma: “*máquina electrónica dotada de una memoria de gran capacidad y de métodos de tratamiento de la información, capaz de resolver problemas aritméticos y lógicos gracias a la utilización automática de programas registrados en ella*”.

Un ordenador transforma los datos de entrada en una salida conteniendo la información requerida. Los datos son conjuntos de símbolos utilizados para representar un valor numérico, un objeto o una idea en la forma adecuada para ser objeto de tratamiento. La representación de los datos se hace en base a un determinado código que le confiere su significado. Así, por ejemplo, para representar números se suelen utilizar los dígitos decimales, aunque se estudiarán muchos otros códigos o formas de representación.

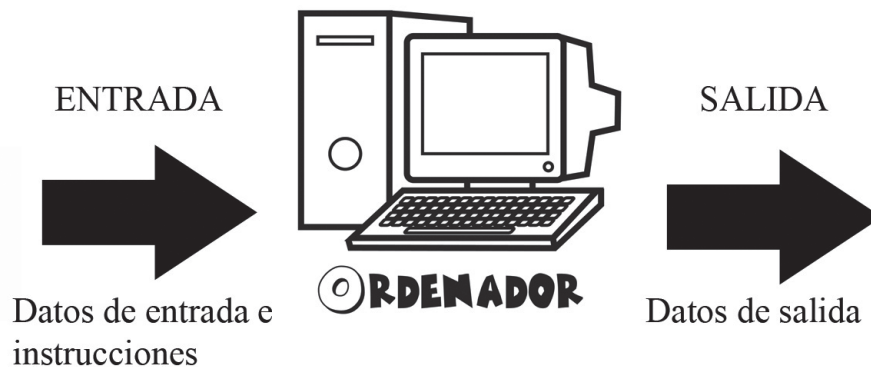


Figura 1.1. Representación esquemática del funcionamiento de un ordenador

El vertiginoso desarrollo de la informática en las últimas décadas ha impulsado, a su vez, multitud de áreas de la sociedad (tecnología, economía, ciencias, investigación...), de manera que hoy en día parece imposible pensar en la mayoría de las actividades de nuestra sociedad sin que aparezca algún sistema informático.

La mayoría de las personas utilizan diariamente un ordenador, ya sea directa y voluntariamente (escribir una carta, consultar el correo electrónico, escuchar música, navegar por Internet, ver una película, realizar cálculos complejos...) o indirectamente (sacar un billete de tren, utilizar un cajero automático, hacer la declaración de la renta, etc.).

Un **sistema informático** puede definirse como un conjunto de partes interrelacionadas. Un sistema informático típico emplea un ordenador que usa dispositivos programables para capturar, almacenar y procesar datos. Dicho ordenador, junto con la persona que lo maneja y los periféricos que lo envuelven, resultan de por sí un ejemplo de un sistema informático.

Estructuralmente, un sistema informático se puede dividir en partes, pero funcionalmente es indivisible, en el sentido de que si se divide, pierde alguna de sus propiedades esenciales. Por eso, un sistema informático sin alguna de sus partes no funcionaría.

Todo sistema informático está compuesto por tres elementos básicos:

- Un **componente físico** (hardware): incluye las placas, circuitos integrados, conectores, cables y sistema de comunicaciones.
- Un **componente lógico** (software): permite disponer de un lenguaje lógico para comunicarse con el hardware y controlarlo. Hay dos tipos de software:
 - **Software de base**: es el conjunto de programas necesarios para que el hardware tenga capacidad de trabajar (hacen posible que la pantalla funcione, que represente lo que se escribe desde el teclado, comunicarse con los periféricos, etc.). Recibe también el nombre de **sistema operativo**.



¿SABÍAS QUE...?

El software de base hace posible que la pantalla funcione, que represente lo que se escribe desde el teclado, que el equipo se comunique con los periféricos, etc.

- **Software de aplicación**: son los programas que maneja el usuario (tratamiento de textos, bases de datos, hojas de cálculo...).
- **Componente humano**: está constituido por las personas que participan en la dirección, diseño, desarrollo, implantación y explotación de un sistema informático.

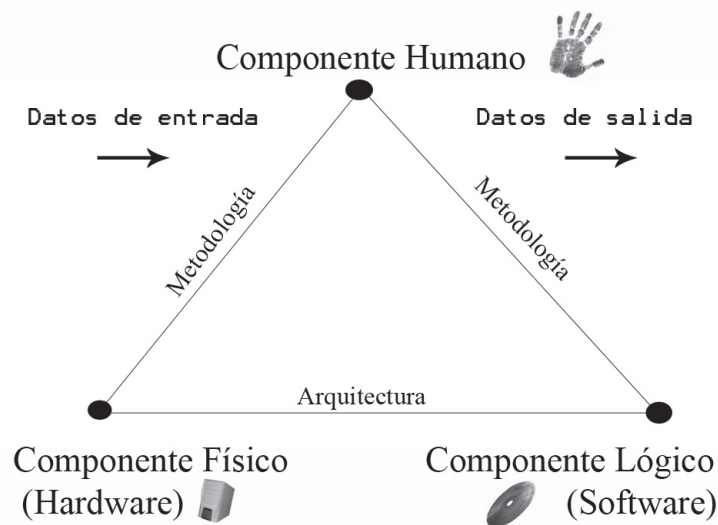


Figura 1.2. Representación de la estructura de un sistema informático

1.1.1 TIPOS DE SISTEMAS INFORMÁTICOS

Los sistemas informáticos se pueden clasificar de dos maneras:

1. Por su **uso** se pueden clasificar en:

- **Sistemas Informáticos de uso general:** son los que se utilizan para varios tipos de aplicaciones (es el caso de los ordenadores personales).
- **Sistemas Informáticos de uso específico:** son los que se caracterizan por ejecutar uno o unos pocos programas (es el caso de los robots industriales o de los videojuegos).

2. Por sus **prestaciones** se pueden clasificar en:

- **Supercomputadores:** son equipos con gran capacidad de cálculo. Se utilizan en el entorno técnico-científico y en la realización de simulaciones, por ejemplo, la *Roadrunner* (Correcaminos), que se utiliza para simular explosiones atómicas, ya que es capaz de realizar 1.000 billones de cálculos por segundo (1 **petaflops** o 1.000 **teraflops**).
- **Sistemas grandes, computadoras centrales o mainframes:** son equipos utilizados para dar soporte a grandes redes de comunicaciones con cientos e incluso miles de usuarios. Por ejemplo, para el procesamiento de las transacciones bancarias.

La distinción entre supercomputadores y computadoras centrales no es sencilla, pero se puede decir que las supercomputadoras se centran en los problemas limitados por la velocidad de cálculo, mientras que las computadoras centrales se centran en problemas limitados por los dispositivos de E/S y la fiabilidad de las transacciones.

- **Sistemas medios o miniordenadores:** son equipos con capacidad para soportar cientos de usuarios con un coste y unas prestaciones inferiores a los grandes sistemas. Son también conocidos como servidores de redes con **terminales tontos** sin capacidad de cálculo propia. Actualmente están en desuso, siendo remplazados por microordenadores interconectados entre sí (denominados **estaciones de trabajo**) y con un **servidor**. Es preciso distinguir estas estaciones de trabajo de las referenciadas en el punto siguiente.
- **Estaciones de trabajo:** son equipos monousuarios muy potentes y especializados, algunos con tecnología RISC, que permiten conseguir un aumento de prestaciones (por ejemplo, para diseño y CAD). Siguiendo las tendencias de rendimiento de las computadoras en general, los microordenadores actuales son más potentes que las mejores estaciones de trabajo de anteriores generaciones. Como resultado, el mercado de las estaciones de trabajo se está volviendo cada vez más especializado, ya que muchas operaciones complejas que antes requerían sistemas de alto rendimiento pueden ser ahora realizadas por computadores de propósito general. Sin embargo, el hardware de las estaciones de trabajo está optimizado para situaciones que requieren un alto rendimiento y fiabilidad, donde un microordenador podría dejar de responder.
- **Microordenadores:** son equipos monousuarios menos potentes que se pueden clasificar de varias maneras. Entre ellas se encuentran: ordenadores profesionales y personales, ordenadores de oficina y domésticos, y ordenadores de escritorio y portátiles.

ACTIVIDADES 1.1



- Busque información sobre varias supercomputadoras.

1.2 EL ORDENADOR

Un ordenador puede definirse como “*el dispositivo, o conjunto de dispositivos programables, compuesto por una o varias unidades de procesamiento, recursos y equipos periféricos destinados al procesamiento de grandes cantidades de datos, capaz de efectuar distintos tipos de cálculos, incluyendo operaciones aritméticas y lógicas a gran velocidad*”.

La primera “máquina” diseñada para realizar cálculos fue el ábaco. Siglos después (alrededor del año 1620), aparecieron las calculadoras mecánicas basadas en engranajes, capaces de realizar operaciones elementales. Estas no eran máquinas automáticas, ya que requerían la intervención humana durante el proceso de cálculo.

Estos dispositivos mecánicos fueron evolucionando hasta que, en 1890, Herman Hollerith creó una máquina para realizar el censo de EE.UU. Esta máquina utilizaba un sistema electrónico para la lectura de las tarjetas perforadas y un sistema mecánico para calcular. En 1924, la compañía fundada por Hollerith cambia de nombre para denominarse **International Business Machines (IBM)**. Comienza la época de los computadores.

1.2.1 ARQUITECTURA DEL ORDENADOR

En 1944, John **von Neumann** describió en su famoso modelo, un computador con programa almacenado en memoria eléctrica. Este modelo se utilizó en la construcción del **EDVAC** (*Electronic Discrete Variable Automatic Computer*) en 1952 y es la que se utiliza en la mayoría de los ordenadores actuales.

Según la arquitectura de von Neumann, un computador está formado por:

- Una **Unidad Aritmético-Lógica (ALU)**, que realiza cálculos y comparaciones y toma decisiones lógicas (determina si una afirmación es cierta o falsa mediante las reglas del Álgebra de Boole).



A mediados del siglo XIX, **George Boole** desarrolló la idea de que las proposiciones lógicas podían ser tratadas mediante herramientas matemáticas. Las proposiciones lógicas (asertos, frases o predicados de la lógica clásica) son aquellas que únicamente pueden tomar valores Verdadero/Falso, o preguntas cuyas únicas respuestas posibles sean Sí/No. Según Boole, estas proposiciones pueden ser representadas mediante símbolos y la teoría que permite trabajar con estos símbolos, sus entradas (variables) y sus salidas (respuestas), a través de la **Lógica Simbólica** desarrollada por él. Dicha lógica simbólica cuenta con operaciones lógicas que siguen el comportamiento de reglas algebraicas. Por ello, al conjunto de reglas de la Lógica Simbólica se le denomina **ÁLGEBRA DE BOOLE**.

- Una **Unidad de Control (UC)**, que interpreta cada una de las instrucciones del programa en lenguaje máquina y, de acuerdo con su microprogramación, ir generando las señales lógicas para que se realicen las modificaciones sobre los registros y/o las posiciones de la memoria principal correspondientes a dicha instrucción de la máquina.
- La **Memoria**, que está formada por los elementos que permiten almacenar y recuperar la información y una serie de **Registros** donde se almacena información temporalmente.



En un sentido más amplio, el concepto de memoria puede referirse también a sistemas externos de almacenamiento, como las unidades de disco.

- Los sistemas de **Entrada/Salida**, que permiten la comunicación con los dispositivos periféricos.

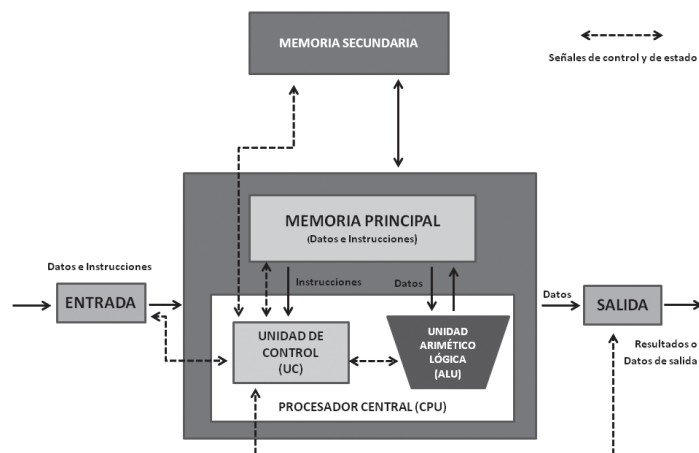


Figura 1.3. Arquitectura de un ordenador

- La **Unidad Central de Proceso (CPU)**, que no es más que un circuito secuencial que interpreta y ejecuta instrucciones. La CPU se compone de los dos primeros elementos del computador (la UC y la ALU) y de los registros. Habitualmente, la CPU es un microprocesador fabricado en un chip.



¿SABÍAS QUE...?

Un chip es un único trozo de silicio que contiene millones de componentes electrónicos.

ACTIVIDADES 1.2



- Busque información sobre von Neumann.

1.2.2 VISIÓN JERÁRQUICA DE UN COMPUTADOR

Para comprender el funcionamiento de un computador es necesario tener una visión global de su estructura jerárquica. Es decir, es importante darse cuenta de que el sistema completo está formado a su vez por subsistemas relacionados entre sí de forma escalonada.

- **Nivel 0 - Nivel de Lógica Digital.** Se corresponde con el hardware real de la máquina (dispositivos y circuitos electrónicos).
- **Nivel 1 - Microprogramación.** En este nivel se encuentran los microprogramas, cuya tarea consiste en interpretar las instrucciones de nivel superior.
- **Nivel 2 - Lenguaje Máquina.** Es el nivel inferior accesible por el usuario y se corresponde con el conjunto de instrucciones que forman el lenguaje directamente interpretable por el hardware, los modos de direccionamiento, los tipos de datos, organización del subsistema de memoria, etc.
- **Nivel 3 - Sistema Operativo.** Es el conjunto de programas que proporcionan facilidades a los niveles superiores en la gestión de los recursos del sistema. De esta manera, se crea un entorno favorable para que el usuario interactúe con los niveles inferiores de la máquina.
- **Nivel 4 - Lenguajes de Alto Nivel.** Son lenguajes de programación con alta abstracción respecto al hardware. El programador encuentra un entorno amigable para la codificación de algoritmos.
- **Nivel 5 - Nivel de Aplicación.** Es el nivel más alejado de la realidad física en el cual el usuario no tiene en cuenta los niveles inferiores.

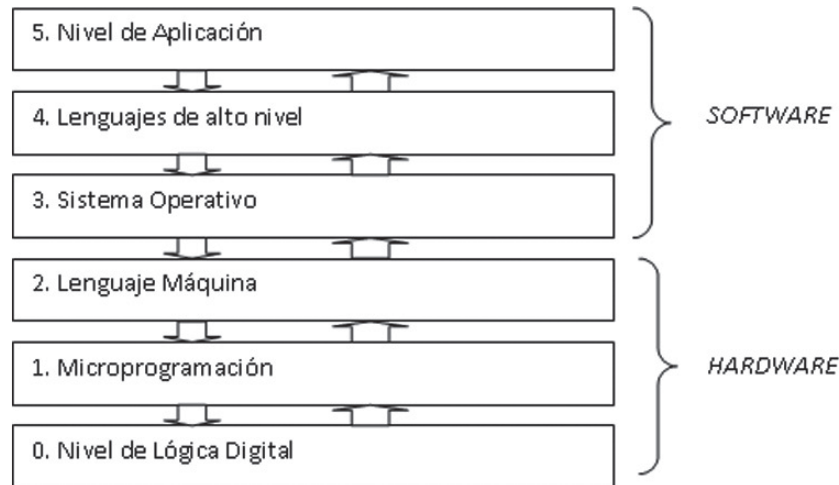


Figura 1.4. Niveles de una computadora

1.3 NORMATIVA LEGAL RELATIVA A LA INFORMÁTICA

Los **derechos de autor** son un conjunto de normas y principios que regulan los derechos morales y patrimoniales que la ley concede a los autores, por el hecho de haber creado una obra literaria, artística, científica o didáctica, esté publicada o inédita.

Los **derechos morales** incluyen dos aspectos específicos: el derecho al reconocimiento de la paternidad de la obra (autoría) y el derecho de un autor a preservar la integridad de la obra (es decir, a negarse a la realización de modificaciones u obras derivadas de la misma).

Los **derechos patrimoniales** son aquellos que se reflejan sobre el patrimonio y permiten satisfacer necesidades valorables en dinero.

Se suele utilizar la denominación de **copyright** (derecho de copia) para especificar la parte patrimonial de los derechos de autor.

Una obra pasa al dominio público cuando los derechos patrimoniales han expirado. Esto sucede trascurrido un plazo desde la muerte del autor. Por ejemplo, en el derecho europeo, 70 años desde la muerte del autor. Dicha obra entonces puede ser utilizada de forma libre, respetando los derechos morales.

En España se desarrollan los derechos de autor en la Ley de la Propiedad Intelectual.

1.3.1 TIPOS DE LICENCIA DE SOFTWARE COMERCIAL

Una **licencia de software** es una autorización formal con carácter contractual que un autor de un software da a un interesado para ejercer actos de explotación cumpliendo una serie de términos y condiciones establecidas dentro de sus cláusulas.

Desde el punto de vista del software comercial, existen distintos grupos de licencias:

- **Licencias OEM:** se trata de un tipo de licencia que supedita su venta a que forme parte de un equipo nuevo, estando prohibido venderlo si no es bajo esta condición. Si bien afecta más que nada a sistemas operativos, también puede afectar a otro tipo de software. Aunque el software comprado bajo este tipo de licencia implica la propiedad del mismo por parte del que lo compra, los fabricantes pueden poner ciertas limitaciones a su uso, como el número máximo de veces que se puede reinstalar.

Los programas adquiridos bajo este tipo de licencia NO se pueden vender ni ceder a terceros, salvo en las mismas condiciones en las que se compraron (es decir, como parte de un equipo).

- **Licencias Retail:** son las versiones de venta de software. En este caso, el programa es de la entera propiedad del usuario, pudiendo éste cederlo libremente a terceros o venderlo.
- **Licencias por volumen:** es un tipo de licencia de software destinado a grandes usuarios (empresas), normalmente bajo unas condiciones similares a las de las licencias OEM, aunque sin estar supeditadas a equipos nuevos.

Básicamente, se trata de estipular un determinado número de equipos que pueden utilizar el mismo código de licencia, quedando el fabricante de dicho software autorizado para hacer las comprobaciones que considere oportunas para ver que las licencias que se están utilizando son las adquiridas.

Normalmente, estas licencias se venden en paquetes de x número de licencias (por ejemplo, en paquetes de 25 licencias como mínimo).

Este tipo de licencia NO se puede ceder a terceros, ni total ni parcialmente.

ACTIVIDADES 1.3



- Si cuenta con una versión de Windows en su ordenador, averigüe si es una versión *OEM*, *retail* o si dispone de una *licencia por volumen*.

1.3.2 TIPOS DE LICENCIA DE SOFTWARE LIBRE

Desde el punto de vista del software libre, existen distintos grupos de licencias:

- **Licencias BSD** (*Berkeley Software Distribution*). Este tipo de licencias se utilizan en el software distribuido junto a los sistemas operativos BSD (sistemas operativos derivados del sistema Unix desarrollados por la Universidad de California en Berkeley). Este tipo de licencias mantienen la protección de los derechos de autor únicamente para la renuncia de garantía y para requerir la adecuada atribución de la autoría en trabajos derivados, pero permite la libre redistribución y modificación, incluso si dichos trabajos tienen propietario. Son muy permisivas y son fácilmente absorbidas al mezclarse con las licencias GNU GPL, con las cuales son compatibles.
- **Licencias GNU GPL** (Licencia Pública General de GNU). Este tipo de licencias conservan los derechos de autor y permiten la redistribución y modificación bajo términos diseñados para asegurarse de que todas las versiones modificadas del software permanecen bajo los términos más restrictivos de la propia GNU GPL. Esto hace que no se pueda crear un producto con partes no licenciadas GPL. Por lo tanto, el conjunto tiene que ser GPL.

Es decir, esta licencia obliga a que si se reutiliza en un mismo programa código “A” licenciado bajo licencia GNU GPL y código “B” licenciado bajo otro tipo de licencia libre, el código final “C”, independientemente de la cantidad y calidad de cada uno de los códigos “A” y “B”, deberá estar bajo licencia GNU GPL.

Aproximadamente el 60% del software licenciado como software libre emplea una licencia GPL.



¿SABÍAS QUE...?

GNU es un acrónimo recursivo que significa *GNU No es Unix*.

- **Licencias GNU LGPL** (Licencia Pública General Limitada de GNU). La licencia GPL ofrece grandes beneficios pero hay algunas veces en las que ofrece ciertas restricciones. Por ejemplo, cuando se utilicen partes o bibliotecas de software GPL en un software propietario o distribuido bajo otra licencia, estaría obligada a distribuir todo su software bajo GPL, lo que, seguramente, no les interesaría.

Sin embargo, con este tipo de licencias, que es una versión reducida de la licencia GNU GPL LGPL, no es obligatorio que el software derivado tenga que ser libre.

Otras actividades que no sean copia, distribución o modificación no están cubiertas en esta licencia y están fuera de su alcance.

Como ejemplo de este tipo de licencia se puede citar el proyecto OpenOffice.org de Sun Microsystems.

- **Licencias MPL** (Licencia Pública de Mozilla). Este tipo de licencia evita el efecto de la GPL (es decir, que si se utiliza código licenciado GPL, el desarrollo final tiene que ser obligatoriamente licenciado GPL). No obstante, la licencia MPL no es tan excesivamente permisiva como las licencias BSD.

Tiene un gran valor porque fue el instrumento que empleó Netscape Communications Corp. para liberar su Netscape Communicator 4.0 y empezar con el proyecto Mozilla.

Se utilizan en gran cantidad de productos de software libre de uso cotidiano en todo tipo de sistemas operativos, como por ejemplo, Mozilla Firefox.

- **Licencias Copyleft** ☺. Este tipo de licencia permite ejercer el derecho de autor permitiendo la libre distribución de copias y versiones modificadas de una obra u otro trabajo, exigiendo que los mismos derechos sean preservados en las versiones modificadas.

Hay que hacer constar que el titular de los derechos de autor (*copyright*) de un software bajo licencia *copyleft* puede también realizar una versión modificada bajo su *copyright* original y venderla bajo cualquier licencia que desee, además de distribuir la versión original como software libre.

Esta técnica ha sido usada por una serie de empresas que realizan software libre (por ejemplo MySQL).

- **Multilicencia.** Con este tipo de licencias, una obra se puede publicar bajo los términos de varias licencias. De esta manera, terceras personas pueden reproducir, distribuir, comunicar públicamente, interpretar y transformar, por cualquier medio, la obra, en cualquier momento o lugar, licenciando o multilicenciando, según sea el caso, la obra original o la obra derivada, bajo los términos de una de las licencias o bajo los términos de un subconjunto de ellas, sujeto a las condiciones particulares que recojan las definiciones de cada una de las licencias.

1.4 CÓMO MEDIR LA INFORMACIÓN

La cantidad mínima de información es la contenida en un **bit** (acrónimo procedente del inglés: *BI*nary *di*git), que representa el 0 ó el 1, el sí o el no, el blanco o el negro (cuando se utiliza para codificar un archivo gráfico), el verdadero o el falso, etc. Es, por ejemplo, la cantidad de información que necesitaba un padre a la puerta del paritorio para saber si se trataba de niño o de niña; actualmente esta pregunta se desvela en las primeras ecografías, pero sigue conteniendo 1 bit de información (código: 0 \Rightarrow niño, 1 \Rightarrow niña, o viceversa).

Un **byte** es el número de bits necesarios para almacenar un carácter y, como generalmente los códigos más usuales utilizan 8 bits para representar cada carácter, siempre que no se indique lo contrario, se considera que 1 byte es equivalente a 8 bits.

Tanto los bits (**b**) como los bytes (**B**) tienen toda una serie de múltiplos que se forman de manera sistemática añadiendo el prefijo correspondiente:

Tabla 1.1 Equivalencia entre las distintas unidades

1 Kilobyte (KB)	$=2^{10}$ Bytes	$=1.024$ Bytes	$\approx 10^3$ Bytes
1 Megabyte (MB)	$=2^{20}$ Bytes	$=1.048.576$ Bytes	$\approx 10^6$ Bytes
1 Gigabyte (GB)	$=2^{30}$ Bytes	$=1.073.741.824$ Bytes	$\approx 10^9$ Bytes
1 Terabyte (TB)	$=2^{40}$ Bytes	$=1.099.511.627.776$ Bytes	$\approx 10^{12}$ Bytes
1 Petabyte (PB)	$=2^{50}$ Bytes	$=1.125.899.906.842.624$ Bytes	$\approx 10^{15}$ Bytes
1 Exabyte (EB)	$=2^{60}$ Bytes	$=1.152.921.504.606.846.976$ Bytes	$\approx 10^{18}$ Bytes

Para tener una idea de lo que representan estas cantidades de información, pensemos que el fichero del diccionario de la Real Academia incluido en la Biblioteca Encarta ocupa algo más de 12 MB; una imagen de 800 por 600 píxeles usada para el escritorio de Windows en color verdadero ocupa 1,3 MB y, por ejemplo, cada minuto de música estéreo en calidad CD viene a suponer 10 MB en formato *wav*, mientras que solo 1 MB en formato *mp3*.

ACTIVIDADES 1.4



➤ ¿Cuántos bits hay en 64 KB?, ¿y en 16 TB?, ¿y en 40 GB?

1.5 REPRESENTACIÓN DE LA INFORMACIÓN

Internamente, los computadores solo reconocen dos símbolos: el 0 y el 1. En cambio, la información manejada por el hombre se representa mediante cifras decimales y letras mayoritariamente, pero no podemos olvidar las imágenes (gráficos, fotografías y vídeo) y los sonidos.

Los factores a tener en cuenta a la hora de seleccionar la codificación binaria más conveniente para cada tipo de datos son:

- ✓ Coste de la traducción.
- ✓ Coste del almacenamiento de la información.
- ✓ Coste del tratamiento de la información.

En el caso de la información textual, la traducción es muy sencilla, pues se hace letra a letra y no se escatima en el coste de almacenamiento pero, para datos numéricos, la situación es variable y, en las distintas formas de representación, hay que analizar el coste de almacenamiento, la complejidad en la traducción, la facilidad para realizar las operaciones numéricas y los rangos de representación.

Otro aspecto en la representación de la información es la tolerancia a fallos en las operaciones de lectura y escritura y en las transmisiones, lo que se consigue incorporando redundancia en la codificación para detectar y corregir los posibles errores mediante chequeos. Además, cuando la cantidad de información a representar es muy grande, como en el caso de un vídeo, se han buscado métodos de compactación para facilitar su almacenamiento y retransmisión a través de la red.

1.5.1 SISTEMAS DE NUMERACIÓN USUALES

Un **sistema de numeración** es una forma de representar cualquier cantidad numérica. Casi todos los sistemas de numeración utilizados en la actualidad son de tipo polinomial. Todo **sistema polinomial** cumple las siguientes características:

- ✓ Todo número es una expresión formada por un conjunto de símbolos, llamados dígitos, cada uno con un valor fijo y diferente a los demás.
- ✓ El número de símbolos distintos que se pueden usar en un determinado sistema de numeración constituye su “base”, es decir, en **base 10 (sistema decimal)** los números que se pueden representar son {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, en base 2 son {0, 1}, en base 8 son {0, 1, 2, 3, 4, 5, 6, 7} y en base 16 {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}.
- ✓ El valor numérico que expresa una determinada combinación de dígitos en una base de numeración dada depende de dos factores: del **valor** de los dígitos y de la **posición** de cada uno de ellos en el polinomio.
- ✓ Cada posición del dígito tiene un valor intrínseco que aumenta de derecha a izquierda según potencias sucesivas de la base del sistema de numeración empleado. El dígito que aparece en el extremo izquierdo es el de más valor o más peso y el colocado en el extremo derecho es el de menos valor o menor peso.

De forma general, en un sistema de numeración de **base b** (b entero y mayor que la unidad), un número N cualquiera (representado como N_b) se puede expresar mediante un polinomio de potencias de la base, multiplicadas por un símbolo perteneciente al sistema de numeración. En el siguiente ejemplo se representan algunos números en forma polinomial:

EJEMPLO 1.1

$$458 \text{ (base 10)} = 458_{10} = 4 * 10^2 + 5 * 10^1 + 8 * 10^0$$

458₁₀: el número 4 es el de mayor peso y el 8 el de menor peso.

$$1110000 \text{ (base 2)} = 1 * 2^6 + 1 * 2^5 + 1 * 2^4 + 0 * 2^3 + 0 * 2^2 + 0 * 2^1 + 0 * 2^0$$

SISTEMA DE NUMERACIÓN EN BASE DOS

El **sistema binario** o en **base 2** es el utilizado internamente por el computador. Cada dígito se denomina **bit** y sus valores posibles son 0 y 1. Así, se puede asociar (aunque pudiera ser al contrario):

0 = “Ausencia de Tensión”

1 = “Presencia de Tensión”

Como se vio de manera general en el punto anterior, un número binario estará formado por un conjunto ordenado de bits. El valor de cada posición del número aumenta de derecha a izquierda según potencias de 2. Así, el valor decimal equivalente de las primeras potencias enteras de 2 son las siguientes:

2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰
128	64	32	16	8	4	2	1

La conversión de un número en base 2 a decimal se realiza fácilmente representando el número mediante su polinomio equivalente y operando éste en base 10.

EJEMPLO 1.2**CONVERTIR NÚMEROS DE BINARIO A DECIMAL**

$$1011_2 = 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0 = \mathbf{11}_{10}$$

$$110,11_2 = 1 * 2^2 + 1 * 2^1 + 0 * 2^0 + 1 * 2^{-1} + 1 * 2^{-2} = \mathbf{6,75}_{10}$$

Tal y como se comentó anteriormente, los bits suelen representarse agrupados en paquetes de ocho, formando lo que se conoce como un **byte**.

Los números con signo

La forma más habitual para representar los números binarios negativos es escribirlos como si fuesen positivos y añadirle un nuevo símbolo (el signo -) a la izquierda del dígito más significativo. En el sistema de numeración utilizado en Electrónica digital (que es el binario) supondría trabajar con tres niveles lógicos de tensión: uno para el **0**, otro para el **1** y otro para el signo (-), lo que sin duda complicaría los sistemas digitales. Para evitar el tener que añadir el símbolo (-), cuando se tienen que codificar tanto números positivos como negativos, se usa un bit para indicar el signo del número. Existen varios tipos de codificación del signo; el más sencillo de todos es añadir un bit extra a la izquierda del número en valor absoluto: si el bit es **1**, el número es negativo; si es **0**, el número es positivo.

EJEMPLO 1.3

-3 → 1011	-4 → 1100	-7 → 1111
+3 → 0011	+4 → 0100	+7 → 0111

SISTEMAS OCTAL Y HEXADECIMAL

El sistema **octal**, o en **base 8**, utiliza los dígitos del 1 al 7.

El sistema **hexadecimal**, o en **base 16**, supone la utilización de 16 dígitos con valores equivalentes a los decimales del 0 al 15; como no se dispone más que de 10 guarismos arábigos, se utilizan a continuación las letras de la A a la F.

1.5.2 CONVERSIONES ENTRE SISTEMAS DE NUMERACIÓN

Si se quieren realizar operaciones con números en distintas bases, suele ser necesario hacer cambios de base, esto es, dado un N_{b_1} , queremos obtener N_{b_2} .

Para llevar a cabo este cambio, existen distintos métodos, según sea el par de bases de trabajo. Como método general se utiliza el siguiente:

$$N_{b_1} \rightarrow N_{10}$$

$$N_{10} \rightarrow N_{b_2}$$

El primer paso es muy sencillo, simplemente hay que descomponer el número en su representación polinomial y sumar sus elementos.

EJEMPLO 1.4**CONVERTIR A DECIMAL**

$$5.562,7_8 = 5 * 8^3 + 5 * 8^2 + 6 * 8^1 + 2 * 8^0 + 7 * 8^{-1} = \mathbf{2.930,125}_{10}$$

$$11.001,1_2 = 1 * 2^4 + 1 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0 + 1 * 2^{-1} = \mathbf{25,5}_{10}$$

Para realizar el segundo paso de la conversión, se trata de forma separada la parte fraccionaria y la parte entera del número en base 10 a convertir.

- **Parte entera.** Para realizar la conversión de la parte entera, es necesario calcular los coeficientes sobre la nueva base **b2**.

Para ello se divide el número en base 10 por la base b2 sucesivamente, obteniendo los coeficientes del resto de cada división hasta que se obtenga un cociente menor que b2; ese cociente será el último coeficiente.

EJEMPLO 1.5**CONVERTIR EL NÚMERO 5.270_{10} A BASE 8**

División	Cociente	Resto (coeficiente)
$5.270 : 8$	658	$a_0 = 7$
$658 : 8$	82	$a_1 = 2$
$82 : 8$	10	$a_2 = 2$
$10 : 8$	1	$a_3 = 2$
		$a_4 = 1$

Resultado: $5.270_{10} = 12.227_8$

EJEMPLO 1.6**CONVERTIR EL NÚMERO 517_{10} A BASE 2**

División	Cociente	Resto (coeficiente)
$517 : 2$	258	$a_0 = 1$
$258 : 2$	129	$a_1 = 0$
$129 : 2$	64	$a_2 = 1$
$64 : 2$	32	$a_3 = 0$
$32 : 2$	16	$a_4 = 0$
$16 : 2$	8	$a_5 = 0$
$8 : 2$	4	$a_6 = 0$
$4 : 2$	2	$a_7 = 0$
$2 : 2$	1	$a_8 = 0$
		$a_9 = 1$

Resultado: $517_{10} = 1.000.000.101_2$

- **Parte fraccionaria.** Abordaremos ahora la conversión de la parte fraccionaria de un número en base 10.

El proceso es similar al anterior, pero multiplicando la parte fraccionaria por la nueva base **b2**; del resultado obtenido en cada multiplicación, la parte entera será el coeficiente y la parte fraccionaria se multiplicará de nuevo por **b2**.

EJEMPLO 1.7**CONVERTIR EL NÚMERO $0,543_{10}$ A BASE 8**

Multiplicación	Entero (coeficiente)	Resto
$0,543 * 8 = 4,344$	$a_{-1} = 4$	0,344
$0,344 * 8 = 2,752$	$a_{-2} = 2$	0,752
$0,752 * 8 = 6,016$	$a_{-3} = 6$	0,016
$0,016 * 8 = 0,128$	$a_{-4} = 0$	0,128
$0,128 * 8 = 1,024$	$a_{-5} = 1$	0,024
		...

Resultado: $0,543_{10} = 0,42601..._8$

Cuando se utilizan los sistemas binarios, octal y hexadecimal, puede emplearse el método general de conversión entre bases, aunque existen métodos más rápidos que no requieren el paso intermedio por base 10, ya que sus bases son potencias de 2. De esta forma, a cada dígito octal le corresponden tres dígitos binarios y a cada dígito hexadecimal, cuatro binarios.

Hexadecimal	2		8				A			
Binario	1	0	1	0	0	0	1	0	1	0
Octal	1	2		1		1		2		

EJEMPLO 1.8**CONVERSIÓN ENTRE BASES**

$$FA3F_{16} = 1111\ 1010\ 0011\ 1111_2$$

$$3.456_8 = 011\ 100\ 101\ 110_2$$

ACTIVIDADES 1.5

➤ Convierta a binario los siguientes números: 525_{10} , 372_8 y $37F_{16}$

➤ Pase a decimal las siguientes cantidades: 111.000_2 , 372_8 y $37F_{16}$

1.5.3 CÓDIGOS ALFANUMÉRICOS

Son aquellos que sirven para representar caracteres alfanuméricos: cifras decimales, letras, caracteres especiales y caracteres de control.

Uno de los códigos más utilizados para este fin es el código **ASCII** (*American Standard Code for Information Interchange*), que utiliza 7 bits para la codificación de letras mayúsculas, minúsculas, cifras numéricas y una serie de signos de puntuación y símbolos gráficos.

Actualmente, se utiliza el código ASCII extendido de 8 bits que codifica todo el alfabeto internacional y permite la compatibilidad con los registros de 8 bits dentro de los ordenadores, además de servir para detectar posibles errores de transmisión.

Tabla 1.2 Códigos ASCII estándar

*	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	TAB	LF	VT	FF	CR	SO	SI
1	DEL	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	P	q	r	s	t	u	v	w	x	y	z	{		}	~	

Posteriormente, han aparecido nuevos sistemas de codificación que permiten representar un mayor número de símbolos alfanuméricos empleando un mayor número de bits en la codificación. Se debe destacar el código *Unicode*, que emplea 16 bits en la codificación.

ACTIVIDADES 1.6



➤ Codifique la palabra "Hola" en ASCII.

1.6 TIPOS DE SOFTWARE

Se define software como el “conjunto de instrucciones o programas usados por una computadora para hacer una determinada tarea”. Tienen carácter virtual, intangible (al contrario que el hardware) y están almacenadas en los diferentes sistemas de almacenamiento.

Para comprender la relación entre hardware y software, es necesario tener una visión global de la estructura de un ordenador; el sistema completo está formado, a su vez, por subsistemas relacionados entre sí de forma escalonada. En los niveles inferiores, la frontera entre el hardware y el software es bastante difusa (Figura 1.4).

- **Nivel 0 - Nivel de Lógica Digital.** Se corresponde con el hardware real de la máquina (dispositivos y circuitos electrónicos).
- **Nivel 1 - Microprogramación.** Son los comandos que intercambian entre sí los circuitos electrónicos. En este nivel, se encuentran los microprogramas, cuya tarea consiste en interpretar las instrucciones de nivel superior. Podemos considerar este nivel como la frontera entre el hardware y el software.
- **Nivel 2 - Lenguaje máquina.** Es el nivel inferior accesible por el usuario y se corresponde con el conjunto de instrucciones que forman el lenguaje directamente interpretable por el hardware, los modos de direccionamiento, los tipos de datos, organización del subsistema de memoria, etc.
- **Nivel 3 - Sistema operativo.** Es el conjunto de programas que proporcionan facilidades a los niveles superiores en la gestión de los recursos del sistema. De esta manera, se crea un entorno favorable para que el usuario interaccione con los niveles inferiores de la máquina.
- **Nivel 4 - Lenguajes de alto nivel.** Son lenguajes de programación con alta abstracción respecto al hardware. El programador encuentra un entorno amigable para la codificación de algoritmos.
- **Nivel 5 - Nivel de aplicación.** Es el nivel más alejado de la realidad física en el cual el usuario no tiene en cuenta los niveles inferiores.

1.6.1 LOS LENGUAJES DE PROGRAMACIÓN

Antes de hablar de los distintos lenguajes de programación, conviene dejar claros varios conceptos:

- **Instrucciones.** Son las distintas órdenes que recibe la computadora para operar con o sobre datos.
- **Programa.** Un programa consiste en una cadena de instrucciones y datos. Estas instrucciones son normalmente ejecutadas en secuencia, con eventuales cambios de flujo causados por el propio programa o eventos externos.
- **Lenguaje de programación.** Es el conjunto de símbolos y reglas empleados para codificar las instrucciones que componen cada programa.

El lenguaje que entienden las computadoras es el **lenguaje máquina**, en el que las instrucciones están formadas por dos campos:

- **Código de operación.** Indica a la máquina la operación a realizar.
- **Operandos.** Indica la dirección de memoria donde se encuentran los operandos o el valor de dichos operandos directamente.

El lenguaje máquina está totalmente ligado al hardware en donde se vaya a ejecutar, es decir, un programa en lenguaje máquina realizado para un determinado procesador no funcionará en otro distinto.

Además, el lenguaje máquina solo nos permite realizar operaciones muy simples, tales como saltos, sumas y restas, lo que implica una programación demasiado laboriosa para implementar cualquier algoritmo, siendo necesario, además, conocer las peculiaridades del hardware que se pretende manejar.

Para evitar estos problemas y facilitar la tarea de programación, surgen los **lenguajes de alto nivel**, que no dependen de la plataforma hardware en la que se vayan a ejecutar y, además, presentan instrucciones más complejas y potentes.

Ejemplos de lenguajes de alto nivel: BASIC, Visual BASIC, C, C++ y Visual C.

La utilización de lenguajes de alto nivel facilita enormemente la tarea de escribir programas pero, para que esas instrucciones sean comprensibles para el procesador, deben ser convertidas antes a lenguaje máquina. Esa conversión se realiza mediante programas traductores, llamados compiladores.

A lo que escribe el programador se le denomina “código fuente”. Al resultado de la “conversión” (compilación) en lenguaje máquina, se le denomina “código objeto”, “binarios” o “ficheros ejecutables”.

1.6.2 LAS APLICACIONES INFORMÁTICAS

Son programas diseñados para resolver las necesidades habituales de los usuarios. Suelen disponer de interfaces que agilizan y facilitan el trabajo al usuario:

- ✓ Gestores de bases de datos.
- ✓ Hojas de cálculo.
- ✓ Procesadores de texto.
- ✓ Correo electrónico.
- ✓ Creación de presentaciones.
- ✓ Diseño.
- ✓ Tratamiento fotográfico.
- ✓ Etc.

Las empresas desarrolladoras de software suelen comercializar paquetes con las aplicaciones más utilizadas por los usuarios como, por ejemplo, Microsoft Office o Lotus SmartSuite.

1.7 EL SISTEMA OPERATIVO

Un **sistema operativo** es un programa o conjunto de programas que actúa como intermediario entre el usuario y el hardware del ordenador, gestionando los recursos del sistema y optimizando su uso.

El sistema operativo es en sí mismo un programa, pero un programa muy especial y quizás el más complejo e importante. Cuando se conecta un ordenador se carga parte del sistema operativo en la memoria y se ejecuta. El sistema operativo *despierta* al ordenador y hace que reconozca a la CPU, la memoria, las unidades de disco y cualquier otro dispositivo conectado a ella, como el teclado, el ratón, la impresora, etc., verificando así que no existan errores de conexión y que todos los dispositivos se han reconocido y trabajan correctamente. A este primer diagnóstico se le denomina **POST**.

El sistema operativo presenta al usuario la máquina de una forma más fácil de manejar y programar que el hardware que está por debajo, es decir, un usuario normal, simplemente abre los ficheros que grabó en un disco, sin preocuparse por la disposición de los bits en el medio físico, los tiempos de espera del motor del disco, la posición de un cabezal, el acceso de otros usuarios, etc.

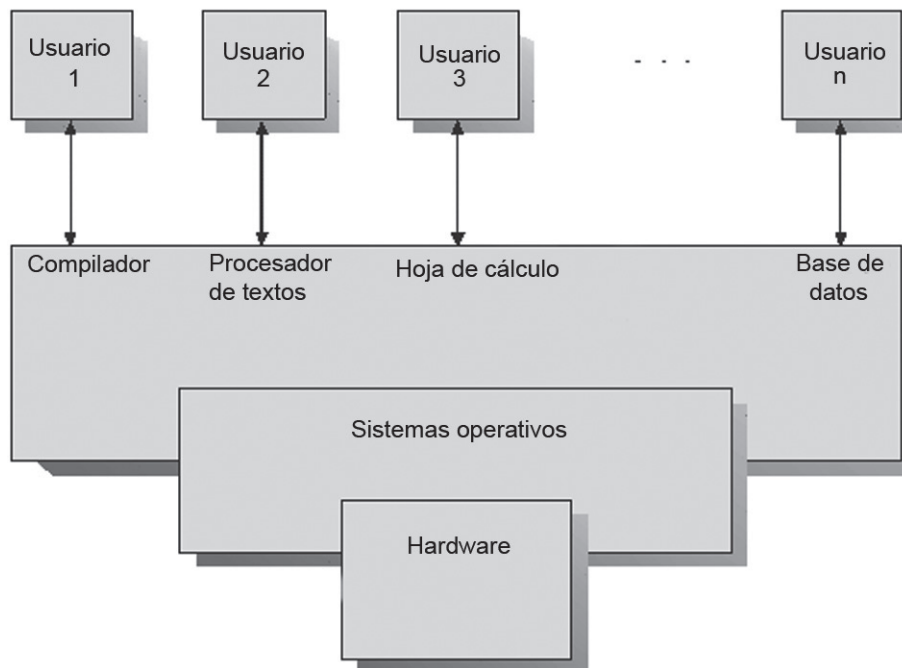


Figura 1.5. El SO como intermediario entre los usuarios y el hardware

1.7.1 FUNCIONES DEL SISTEMA OPERATIVO

A continuación se muestran las funciones principales que realiza todo sistema operativo:

- **Control de la ejecución de los programas.** Para ello, acepta los trabajos, administra la manera en que se realizan, les asigna los recursos y los conserva hasta su finalización.
- **Administración de periféricos.** Coordinando y manipulando los dispositivos conectados al ordenador.
- **Gestión de permisos y de usuarios.** Adjudica los permisos de acceso a los usuarios y evita que las acciones de uno afecten el trabajo que está realizando otro.
- **Control de concurrencia.** Establece prioridades cuando diferentes procesos solicitan el mismo recurso.
- **Control de errores.** Gestiona los errores de hardware y la pérdida de datos.
- **Administración de memoria.** Asigna memoria a los procesos y gestiona su uso.
- **Control de seguridad.** Debe proporcionar seguridad tanto para los usuarios como para el software y la información almacenada en los sistemas.

En concordancia con estas funciones principales, es posible analizar la estructura de un sistema operativo en cinco niveles. Los primeros dos niveles entrarían dentro de la parte del sistema operativo dependiente del hardware, el resto de los niveles pertenecen a la parte portable del mismo (ver Figura 1.6).

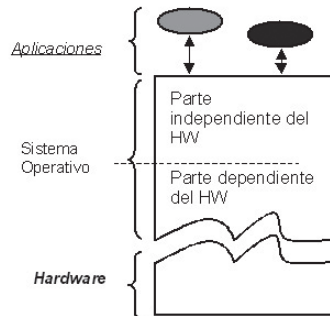


Figura 1.6. Esquema de relación entre partes software y hardware

Cada uno de los niveles se comunica con el inmediatamente inferior y superior coordinando sus funciones.

- **Nivel 1 - Gestión del procesador.** En este nivel se encuentra la parte del sistema operativo encargada de la gestión de la CPU. En los sistemas operativos multiproceso (es decir, que pueden ejecutar varios procesos a la vez), este nivel se encarga de compartir la CPU entre los distintos procesos realizando funciones de sincronización, conmutación de la CPU y gestión de interrupciones.
- **Nivel 2 - Gestión de memoria.** Este nivel es el encargado de repartir la memoria disponible entre los procesos. Se realizan funciones de asignación y liberación de memoria, y el control de violación de acceso a zonas de memoria no permitidas.
- **Nivel 3 - Gestión de procesos.** Este nivel es el encargado de la creación y destrucción de los procesos, intercambio de mensajes y detección y arranque de los mismos.

- **Nivel 4 - Gestión de dispositivos.** En este nivel se realiza la gestión de las entradas/salidas (E/S) en función de los dispositivos existentes. Entre otras, se encarga de las funciones de creación de procesos de E/S, asignación y liberación de dispositivos E/S, y planificación de la E/S.
- **Nivel 5 - Gestión de la información.** El objetivo de este nivel es el de gestionar el espacio de nombres lógicos, utilizados para simplificar el acceso a los recursos, ya que mediante estos se sustituyen rutas de acceso que pueden ser muy largas y difíciles de recordar por un solo nombre, encargándose el sistema operativo, de forma totalmente transparente para el usuario, de realizar esa búsqueda de ruta. Otro de sus cometidos es la protección de la información realizando funciones de creación y destrucción de ficheros y directorios, apertura y cierre de ficheros, lectura y escritura de ficheros y protección de acceso.

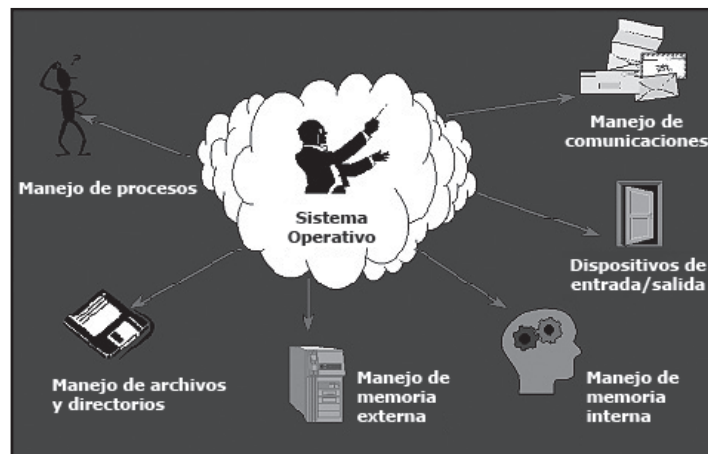


Figura 1.7. El SO como director de operaciones sobre el hardware

Es importante destacar que un mismo sistema operativo puede trabajar en múltiples plataformas hardware, por lo que debe poder adaptarse a las peculiaridades de cada una de ellas.

Imagínese un programador que pretenda realizar una aplicación de gestión de archivos, teóricamente debería conocer las peculiaridades del hardware donde correrá su aplicación a la hora de manipular archivos, pero gracias a la existencia del sistema operativo el programador puede abstraerse de las peculiaridades de la máquina y su aplicación funcionará correctamente, independientemente del hardware que esté por debajo.

1.7.2 EL KERNEL

El **Kernel** es el núcleo del sistema operativo. Cuando arranca el ordenador, se carga en memoria y permanece allí, realizando funciones básicas como:

- ✓ Manejo de la memoria.
- ✓ Comunicación entre procesos.
- ✓ Control de interrupciones.
- ✓ Manejo de condiciones de error.
- ✓ Traslado de control de un proceso a otro.
- ✓ Control de periféricos.

1.7.3 PROGRAMAS Y PROCESOS

Un **proceso** es un programa en ejecución. Cada proceso se compone de un código que se ejecuta y una estructura de datos, estando ambos cargados en memoria.

No hay que confundir *procesos* con *archivos* o *programas*. Por ejemplo, un compilador C no es un proceso, pero un compilador C ejecutándose será un proceso para el sistema operativo y, por tanto, le asignará recursos (CPU, memoria, etc.) y controlará su ejecución.

La estructura de datos sirve para identificar unívocamente cada proceso y controlar todos los aspectos de su ejecución. Cada estructura contiene básicamente los siguientes datos:

- **Estado actual del proceso:** puede ser *En ejecución*, *Esperando*, *Detenido*, etc.
- **Identificación unívoca del proceso:** a cada proceso se le asigna un PID (código identificador de proceso) que se utilizará para identificarlo ante el sistema operativo.
- **Prioridad del proceso:** cada proceso tiene asignada una prioridad de forma que, en cualquier instante, el proceso que mayor prioridad tiene asignada, de entre los que están en estado *Esperando*, es el que se ejecutará.
- **Zona de memoria asignada:** cada proceso debe tener una zona de memoria independiente que no puede ser interferida por otros procesos.
- **Recursos asociados al proceso:** como ficheros, semáforos, etc.

PLANIFICACIÓN DE CPU-MULTIPROCESAMIENTO

Existen ordenadores que disponen de varios procesadores, por lo que son capaces de ejecutar al mismo tiempo varias tareas, estos computadores disponen de sistemas operativos especiales denominados **Sistemas multiprocesador**. En los ordenadores convencionales, normalmente solo se dispone de un procesador, por lo que en cada momento solo se estará procesando una tarea (actualmente, los procesadores disponen de varios núcleos que, en determinadas actuaciones, pueden actuar como si fueran varios procesadores). No obstante, es posible crear la ilusión de multiprocesamiento por medio de interrupciones, cambiando el proceso activo cada cierto tiempo aunque no se haya finalizado la tarea en ejecución.

El sistema operativo se encarga de compartir la CPU entre los distintos procesos de la manera más eficiente posible, persiguiendo los siguientes objetivos:

- **Equidad.**
- **Eficiencia.**
- **Bajo tiempo de respuesta.**
- **Rendimiento alto.**
- **Minimización del tiempo de espera.**

Desgraciadamente, todos estos objetivos no se pueden conseguir simultáneamente.

Existen dos formas básicas de realizar esta conmutación entre tareas.

- **Cooperative switching.** Las tareas cooperan unas con otras suspendiéndose voluntariamente cada cierto tiempo en puntos determinados de su ejecución. Windows 3.x es un ejemplo de este método de conmutación.
- **Pre-emptive switching.** El sistema operativo interrumpe la tarea que se está ejecutando según cierto algoritmo de prioridades. Linux y todas las versiones desde Windows 95 son ejemplos de este método de conmutación.

Cuando un proceso se suspende temporalmente, deberá reiniciarse después, exactamente en el mismo estado en que se encontraba cuando se detuvo. El sistema operativo se encarga de guardar el estado en el que se encuentra la tarea al suspenderse, guardando el contenido de registros del procesador y reemplazándolos con valores de la próxima tarea a ejecutar.

Este cambio de contenido de registros se denomina **cambio de contexto**. Este cambio requiere un tiempo muy pequeño en comparación al tiempo que se le asigna a cada tarea para su ejecución.

ALGORITMOS DE PLANIFICACIÓN

Existen distintos algoritmos empleados para realizar la planificación. El más sencillo de ellos es el **PEPS** (Primero en Entrar, Primero en Salir o **FIFO**, en inglés *First Input First Output*), en el cual no se libera la CPU hasta que no termina el proceso en ejecución y los tiempos de respuesta varían fuertemente de un momento a otro, según el orden de llegada a la CPU de los procesos.

EJEMPLO 1.9

Tenemos tres procesos con distintos tiempos de procesamiento:

Proceso 1: 18 Unidades

Proceso 2: 6 Unidades

Proceso 3: 6 Unidades

Los ejecutamos atendiendo al algoritmo FIFO sin tener en cuenta la prioridad de los procesos.

En el primer caso:

PROCESO 1-----PROCESO 2-----PROCESO 3

Estudiando el tiempo medio de retorno, es decir, el tiempo medio que se tarda en finalizar un proceso, tendremos: $(18 + 24 + 30)/3 = 24$

En el segundo caso, el orden de llegada de los procesos es otro:

PROCESO 2-----PROCESO 3-----PROCESO 1

El tiempo medio de retorno en este caso será: $(6 + 12 + 30)/3 = 16$. Al hacer la media se aprecia que los procesos están menos tiempo esperando.

Se observa que, empleando el algoritmo FIFO, si apareciera una tarea crítica que requiriese atención inmediata, ésta debería esperar a la finalización de las tareas en cola.

Otro tipo de algoritmo de planificación es el **SJF** (*Shortest Job First*), es decir, el trabajo más corto se ejecuta primero. Este algoritmo asigna la CPU al trabajo que requiere menor tiempo de proceso.

La dificultad reside en saber cuál de los procesos en espera de ser ejecutados tendrá menor tiempo de proceso, para ello se emplean algoritmos de predicción que calculan el siguiente tiempo de ejecución de un proceso como una media exponencial de los tiempos de las últimas ejecuciones de esa parte de código.

El problema que puede presentarse es que vayan entrando a la cola de espera de ejecución los procesos cortos y los procesos largos que estén esperando no se ejecuten nunca. Para evitar esto, se emplean técnicas de envejecimiento de procesos, aumentando la prioridad de los procesos que llevan mucho tiempo inactivos.

Uno de los algoritmos más sencillos, justo y de uso más amplio es el **Round Robin**. En su utilización no se establecen prioridades, cada proceso tiene asignado un tiempo de ejecución denominado **quantum**; si se cumple ese tiempo y la tarea no ha concluido, se da paso al siguiente proceso y el proceso no finalizado pasa al final de la lista de procesos en espera.

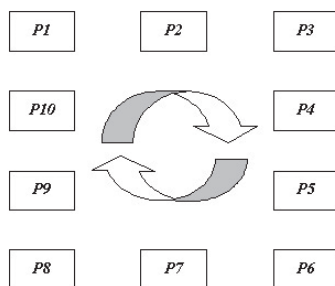


Figura 1.8. Planificación Round Robin

Tal y como se muestra en la Figura 1.8, si se está ejecutando el proceso 1 y se agota su cantidad de *quantum*, se desalojaría la CPU y pasaría a ejecutarse el proceso 2. Cuando termine el *quantum* del proceso 10 se pasará de nuevo al proceso 1.

Otros tipos de algoritmos son los basados en **prioridades**. En ellos se asocia una prioridad a cada proceso y la CPU se asigna al trabajo con prioridad más alta en cada momento. Normalmente, si se está ejecutando un proceso de prioridad media y entra un proceso de prioridad mayor, se requisa la CPU al primer proceso y se le entrega al proceso de mayor prioridad.

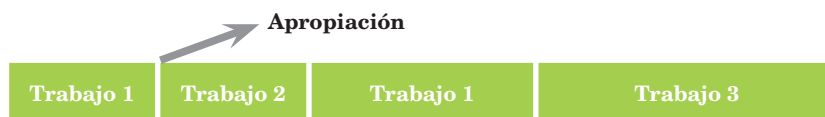


Figura 1.9. Esquema de utilización de la CPU con prioridades

En este ejemplo se observa que la CPU está ejecutando el trabajo 1 cuando se recibe la petición del trabajo 2 que tiene alta prioridad; entonces, se desaloja la CPU y se atiende la petición del trabajo 2 y, cuando termine, se recuperará el trabajo 1.

Normalmente, en los sistemas operativos suelen aparecer híbridos de varios algoritmos de planificación.

Por ejemplo, se puede trabajar con un algoritmo basado en **prioridades**, aplicando **SJF** o **Round Robin**, cuando existan varios procesos con la misma prioridad.

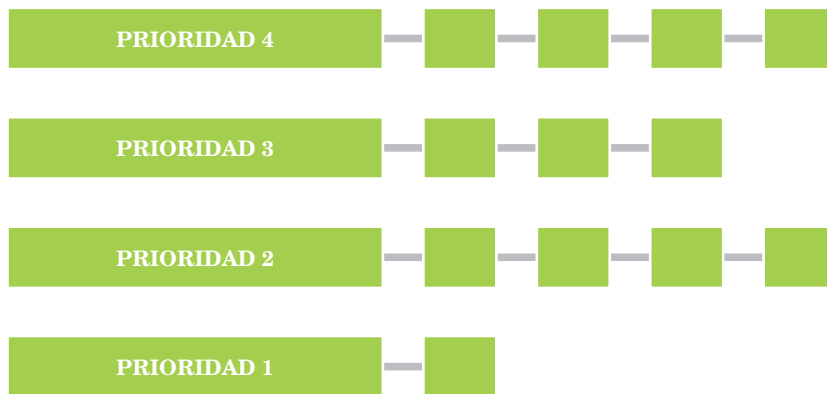


Figura 1.10. Algoritmo de planificación con 4 clases de prioridad

Los procesos de prioridad más alta (prioridad 4) se ejecutarán primero y, como hay 4 procesos, la CPU se repartirá utilizando el algoritmo *Round Robin*. Cuando estos procesos finalicen, se pasará a los de prioridad 3 y así sucesivamente.

SINCRONIZACIÓN DE PROCESOS

Cuando se ejecutan varios procesos a la vez, es posible que estos compartan uno o varios recursos del sistema (monitor, memoria...). El objetivo del sistema operativo es permitir que varios procesos compartan recursos sin que se produzcan problemas. Los procesos pueden ser **independientes** o **cooperantes**.

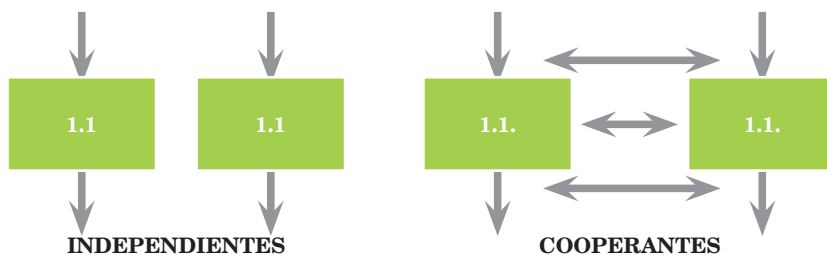


Figura 1.11. Procesos independientes y cooperantes

Los **procesos independientes** pueden ejecutarse a la vez, detenerse y reanunciarse sin ningún efecto negativo, son procesos deterministas y reproducibles. Por ejemplo, un programa que calcule números primos.

Los **procesos cooperantes** comparten su estado, su funcionamiento no es determinista (pueden darse situaciones inesperadas) y puede ser irreproducible. Por ejemplo, dos procesos que se ejecutan a la vez y comparten la pantalla, uno escribiendo «abc» y el otro «cba». Según sea la sincronización de los procesos por el sistema operativo, en la pantalla podrá verse «abccba» o «acbbca» y otras combinaciones.

En el caso de los procesos cooperantes, es posible que se produzcan situaciones peligrosas en las que se pierda el control.

Por ejemplo, dos personas que viven juntas:

Tabla 1.3 Ejemplo práctico de procesos cooperantes

Persona A	Persona B
09:50 Llega a casa	10:10 Llega a casa
10:00 Mira la nevera. No hay leche	10:20 Mira la nevera. No hay leche
10:10 Va a la tienda	10:30 Va a la tienda
10:20 Llega a la tienda	10:40 Llega a la tienda
10:30 Deja la tienda	10:50 Deja la tienda
10:40 Llega a casa y guarda la leche	11:00 Llega a casa y, ¡¡ya hay leche!!

Este problema práctico puede extrapolarse a múltiples casos informáticos.

Los dos procesos comparten un recurso y no existe ningún tipo de sincronización entre ellos. Es necesario que exista una forma de prohibir que más de algún proceso lea o escriba datos compartidos a la vez.

Lo que hay que conseguir es:

- **Exclusión mutua.** Es un mecanismo por el que se asegura que solo una persona o proceso estará haciendo algo en un instante determinado y los otros estarán excluidos (por ejemplo, dos procesos que comparten una sección de memoria).
- **Sección crítica.** Es una sección de código en la que se realizan operaciones que pueden dar lugar a interferencias con otros procesos. Cuando un proceso esté ejecutando su sección crítica, ningún otro proceso puede estar en ella. Es decir, la sección crítica debe ejecutarse en exclusión mutua. Hay que tener en cuenta que un proceso no puede *dormirse* en su sección crítica porque estará bloqueando la sección crítica de otros procesos en ejecución.

1.7.4 BLOQUEOS

La mayoría de los recursos que existen en cualquier ordenador solo pueden ser utilizados por un proceso a la vez. Basta pensar, por ejemplo, en una impresora: no es posible que dos procesos hagan uso al mismo tiempo de este dispositivo, el resultado sería catastrófico.

Sin embargo, en un sistema operativo multitarea, la situación anteriormente comentada puede darse en multitud de ocasiones y el sistema operativo deberá encargarse de que esto no ocurra, otorgando acceso exclusivo a los recursos del sistema susceptibles de ser accedidos por más de un proceso al mismo tiempo.

No obstante, la apropiación de un dispositivo E/S para uso exclusivo puede provocar, a su vez, problemas graves que el sistema operativo también deberá encargarse de resolver.

Imagínese dos procesos que desean imprimir un gran archivo situado en una cinta al mismo tiempo. El proceso **A** solicita el uso exclusivo la impresora mientras que el proceso **B** solicita el uso exclusivo de la cinta. El proceso **A** no puede comenzar la impresión porque no tiene acceso a la cinta, pero no libera la impresora. Mientras tanto, el proceso **B** no inicia la impresión, ya que no tiene acceso a la impresora, pero no libera la apropiación de la cinta. Se dice que los procesos se han bloqueado y permanecerán así para siempre si el sistema operativo no se encarga de solucionarlo.

El **bloqueo** de un conjunto de procesos se produce cuando un proceso espera un evento que solo puede ser provocado por otro proceso del conjunto. Los procesos permanecerán esperando, ninguno de ellos realizará ninguna acción hasta que otro libere algún recurso, por lo que se entra en un bucle de espera infinito.

El bloqueo se denomina también de manera muy ilustrativa como **abrazo mortal**.

Los sistemas operativos emplean distintas técnicas para enfrentarse al problema de los bloqueos:

- **Algoritmo del avestruz.** Es decir, meter la cabeza bajo tierra e ignorar el problema. Si después de estudiar la probabilidad de bloqueos en el sistema, se llega a la conclusión de que ésta es muy baja (por ejemplo, un bloqueo cada seis años) es posible que los programadores decidan no perder el tiempo en implementar técnicas de detección y corrección de bloqueos. Linux funciona con este algoritmo.
- **Evitarlos.** Mediante un estudio cuidadoso de la asignación de recursos, el sistema operativo puede conseguir que nunca se produzcan bloqueos.
- **Detección y recuperación.** Los bloqueos se producen pero el sistema operativo se encarga de detectarlos y recuperarse de ellos.

1.7.5 GESTIÓN DE MEMORIA

Desde un punto de vista computacional, la memoria de un ordenador puede verse como una tabla de palabras, cada una de ellas con su propia dirección.

Tabla 1.4 Estructura de la memoria

Dirección	Dato
F0h	CCh
F1h	ABh
F2h	ACh
F3h	AAh
...	...

Como se ha comentado anteriormente, una de las funciones básicas del sistema operativo es la de hacer ver al usuario que se ejecutan varios procesos *al mismo tiempo* en un sistema que realmente es monoproceso. Para ello, deberá compartir todos los dispositivos entre los distintos procesos buscando la máxima eficiencia sin que existan problemas de coordinación.

Cada proceso en ejecución necesitará un área de memoria para ubicar tres tipos de datos: el **código** del proceso (las instrucciones), los **datos** (variables inicializadas) y la **pila** (zona de trabajo durante la ejecución).

El **gestor de memoria** es la parte del sistema operativo encargada de asignar la memoria a los procesos y gestionar su uso. Debe proporcionar protección y uso compartido, es decir, debe proporcionar un espacio de memoria para cada proceso que lo necesite y controlar que ningún proceso trabaje en zonas de memoria que no le correspondan.

Cabe señalar que dos procesos pueden tener una zona de memoria de trabajo compartida a la que los dos puedan tener acceso. Por ejemplo, imagínese un sistema de control de temperatura de una habitación con dos procesos. Uno de ellos se encarga de leer la temperatura de un sensor; el otro, a partir de dicha temperatura y del valor deseado, controla un generador de frío/calor. En función de estas condiciones, el primer proceso leerá la temperatura y la depositará en una posición de la memoria a la que el segundo proceso deberá tener también acceso.

En este caso, el gestor de memoria deberá controlar el acceso a esas zonas compartidas para que los datos escritos y leídos sean correctos y no se produzcan colisiones o lecturas erróneas (por ejemplo, que el segundo proceso lea un dato a medio escribir por el primer proceso).

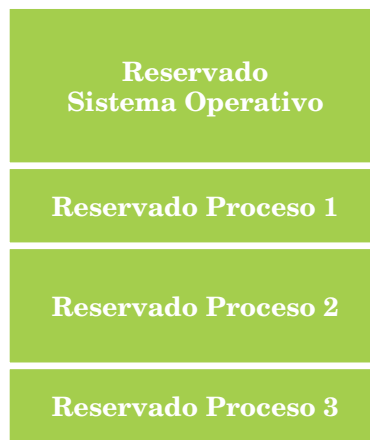


Figura 1.12. Esquema de uso de memoria en un sistema multiproceso

Los criterios empleados para evaluar un gestor de memoria son:

- **Memoria desaprovechada.** Se corresponde con la memoria que se pierde en el proceso de asignación.
- **Complejidad en el tiempo.** Es el tiempo perdido en el proceso de acceso a la zona de memoria asignada.
- **Procesos suplementarios de accesos a memoria.** Son los pasos a realizar para acceder a un dato en la memoria.

Como en el caso del uso de CPU, existen diferentes algoritmos para la gestión de memoria.

La opción más sencilla para realizar las asignaciones es dividir la memoria en m partes, del mismo o distinto tamaño. Cuando se cargue un trabajo, se le asignará la partición de memoria más pequeña disponible que pueda contenerlo. El problema de este tipo de asignación por particiones fijas es el desaprovechamiento de memoria que lógicamente se produce.

FRAGMENTACIÓN

La **fragmentación** puede definirse como la cantidad de memoria desaprovechada por el gestor de memoria. Puede ser de dos tipos:

- **Fragmentación interna.** Se debe a la diferencia de tamaño entre la partición de memoria y el objeto residente dentro de ella. Una posible solución a la fragmentación interna sería la asignación dinámica de memoria. En ella, los bloques no tienen un tamaño predefinido y el gestor de memoria asignará un espacio de memoria según las características del proceso a ejecutar.

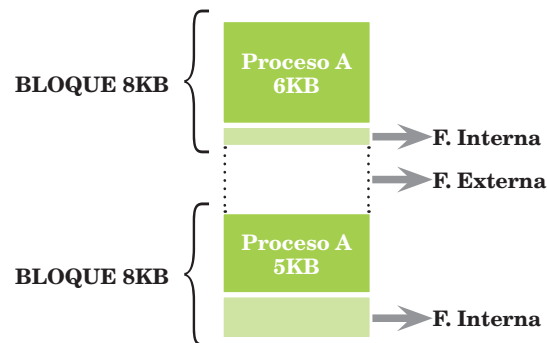


Figura 1.13. Asignación de bloques y tipos de fragmentación

- **Fragmentación externa.** Se debe al desaprovechamiento de memoria entre particiones no contiguas. Para solucionar la fragmentación externa sería necesario poder reubicar programas en tiempo de ejecución, lo cual es conocido como **compactación de memoria**.

En la práctica, las particiones fijas no son muy atractivas, sobre todo si se dispone de poca memoria, puesto que la mayor parte de ésta se desperdicia con programas menores que sus particiones. En vez de esto, se utiliza un algoritmo de gestión de memoria basado en **particiones variables**.

Con las particiones variables el número, posición y tamaño de las mismas varía de forma dinámica. Con ello, la gestión de memoria es mucho más eficiente ya que el tamaño de cada bloque se adapta a las necesidades de cada proceso, eliminándose la fragmentación interna.

Lógicamente, trabajar con particiones variables implica un gestor de memoria mucho más complejo.

SEGMENTACIÓN

La **segmentación** es un esquema de gestión de memoria evolucionado que se basa en particiones variables y presenta varias ventajas:

- ✓ Los bloques de un proceso pueden estar situados en áreas de memoria no contiguas (datos, pila, código).
- ✓ Suministra compactación de memoria.
- ✓ Permite protección y uso compartido.

La **compactación de memoria** es un proceso por el cual se elimina la fragmentación externa. El bloque de memoria asignado a un proceso puede reubicarse dinámicamente para aprovechar mejor la memoria.

1.7.6 MEMORIA VIRTUAL

En el año 1961, Fotheringham desarrolló una nueva técnica de gestión de memoria que denominó **memoria virtual**. La idea básica es que el tamaño del programa, los datos y la pila, combinados, pueden ser mayores que la memoria disponible para ese proceso. El sistema operativo guarda aquellas partes del programa de uso corriente en la memoria principal y el resto, en disco.

También es lógico pensar que no todas las partes de un programa se utilizan al mismo tiempo, por lo que pueden cargarse en memoria únicamente los datos que se vayan a utilizar inicialmente y, posteriormente, ir cogiendo el resto del disco.

Por ejemplo, si un proceso necesita 1 MB de memoria para su ejecución, se le puede asignar únicamente 512 KB. Cuando el sistema operativo necesite un bloque de datos que no se encuentre en la memoria, tiene que pasar un bloque de memoria al disco y traerse del disco el bloque de datos que necesita. A este proceso se le conoce como **swapping**.

Además, la memoria virtual es útil, ya que mientras un programa hace *swapping*, otro puede tener al procesador realizando un uso más eficiente de la CPU.

El sistema operativo decide qué partes del programa puede cargar en memoria, cuándo cargarlas y dónde ubicarlas; de esta manera, existe el riesgo de perder mucho tiempo en accesos al disco si la elección es errónea.

Las principales ventajas que ofrece la utilización de memoria virtual son:

- ✓ Permite trabajar con programas de mayor tamaño que la memoria física.
- ✓ Permite tener más programas cargados a la vez.
- ✓ Permite que los programas empiecen a ejecutarse antes.
- ✓ Reduce la frecuencia de intercambio de procesos.
- ✓ Hay menor trabajo de E/S durante el intercambio de procesos.

PAGINACIÓN

La paginación consiste en dividir el proceso en **páginas** (bloques de datos). El sistema operativo divide la memoria en bloques (marcos de página) y solo carga en memoria las páginas que se vayan a utilizar. Si la predicción se realiza correctamente y se cargan en memoria solo las páginas que realmente se van a utilizar, el proceso se ejecutará exactamente igual que si se hubieran traído todas las páginas.

Cuando el proceso solicita un dato que se encuentra en una página que no está cargada en memoria, se produce un **fallo de página** (*page fault*) y el sistema operativo carga la página solicitada. Por este motivo, a este tipo de paginación se le conoce también como **paginación a demanda**.

La ventaja de la paginación reside en la eliminación de fragmentación interna y la posibilidad de almacenar las páginas de un proceso de manera no contigua.

EJEMPLO 1.10

En un instante determinado existirán varios bloques (marcos de página) de la memoria cargados con páginas (bloques de datos); el resto de marcos pueden estar vacíos o utilizados para otros fines.

Imaginemos un proceso que tiene un tamaño de 4 páginas. En el ejemplo se observa que únicamente las páginas 2 y 3 están cargadas en memoria, las otras dos, en caso de necesitarse, provocarían fallo de página y se traerían desde el disco.

Además, las páginas 2 y 3 están cargadas desordenadamente: la página 2 está cargada en el marco 0 de memoria y la página 3 en el marco 2.

Marcos de página	Bloques cargados
0	2
1	--
2	3
3	--

Si accedemos a la primera dirección virtual de memoria (primer marco de página), observamos que físicamente estamos accediendo a la primera dirección de la página 2 del proceso.

Si empleamos páginas de 1 K (1.024 palabras), el marco 0 se direcciona con las direcciones virtuales 0 a 1.023, el marco 1 con las direcciones 1.024 a 2.047, y así sucesivamente.

Dirección virtual	Marco de página	Dirección física
0 – 1.023	0	2.048 – 3.071
1.024 – 2.047	1	Fallo de página
2.048 – 3.071	2	3.072 – 4.095
3.072 – 4.095	3	Fallo de página

ALGORITMOS DE REEMPLAZO DE PÁGINAS

Cuando ocurre un fallo de página, se hace necesario cargar en memoria la página solicitada, pero puede ocurrir que no haya marcos libres. Entonces se debe elegir una *víctima* entre las páginas que están en memoria para descargarla y darle su lugar a la nueva página requerida.

Si la página *víctima* fue modificada mientras estuvo en memoria, se escribirá su contenido al disco y se cargará la página referida en el marco recién liberado, se cambiará la tabla de páginas y se reiniciará la ejecución.

Existen diferentes algoritmos de reemplazo, algunos de los cuales se describen a continuación:

- **FIFO** (*First Input First Output*). Es el más simple de ellos, tanto para entender como para programar. Consiste en elegir la página que lleve más tiempo en memoria, sin considerar cuándo se utilizó por última vez.
- **LRU** (Usado Menos Recientemente). Este algoritmo consiste en considerar que un acceso a página reciente puede implicar un acceso futuro próximo. El algoritmo elige como víctima aquella página, de entre las que se encuentran en memoria, que hace más tiempo que no ha sido referenciada.
- **ÓPTIMO**. Es el que produce menor tasa de *page fault*. Es un algoritmo teórico, pues exige conocer las futuras referencias a página, eligiendo como víctima aquella que no volverá a ser referenciada nunca más o lo hará muy posteriormente. Sirve para estudios comparativos cuando se analizan nuevos algoritmos.

1.7.7 DISPOSITIVOS DE ENTRADA/SALIDA

Una de las funciones más importantes y complejas que realiza el sistema operativo es la destinada a manejar los diferentes periféricos existentes. Debe enviar comandos a los dispositivos, detectar interrupciones y controlar errores.

Los dispositivos de entrada/salida se dividen, en general, en dos tipos:

- **Dispositivos orientados a bloques:** trabajan con bloques de tamaño fijo, además tienen la propiedad de que se pueden direccionar, es decir, es posible escribir o leer cualquier bloque independientemente ya que cada uno tiene una dirección concreta. Entre ellos se encuentran los discos duros, la memoria y los discos compactos.
- **Dispositivos orientados a carácter:** trabajan con secuencias de bytes sin importar su longitud ni ninguna agrupación en especial. No son dispositivos direccionables. Entre ellos se encuentran el teclado, el ratón, la pantalla y las impresoras.

CONTROLADORES DE DISPOSITIVOS

Los dispositivos de entrada/salida (E/S) pueden dividirse en una parte mecánica y una parte electrónica. Cuando se accede a un dispositivo, siempre se hace a su parte electrónica, denominada **controlador del dispositivo**. El controlador se encargará de mostrar al sistema operativo una interfaz estándar adaptándose a las peculiaridades del hardware que está por debajo.

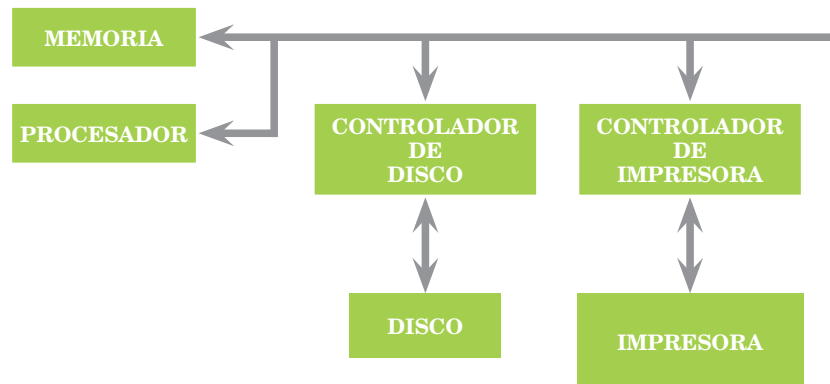


Figura 1.14. Acceso a dispositivos Entrada / Salida

ACCESO DIRECTO A MEMORIA

Muchos controladores, en particular los correspondientes a dispositivos de bloque, permiten el **DMA** (*Direct Memory Access*). El objetivo del DMA es solapar operaciones de CPU y E/S. La CPU proporciona al controlador información sobre la dirección de memoria para acceder y el número de datos a transferir. Una vez realizada la petición, la CPU se despreocupa momentáneamente de la transferencia y continúa realizando otras tareas. El controlador va leyendo los datos del disco y copiándolos en la memoria, una vez realizada la transferencia el controlador provoca una interrupción que hace que la CPU abandone el trabajo que estaba realizando.



Figura 1.15. Acceso a dispositivos de almacenamiento

MANEJADORES DE DISPOSITIVOS

Una vez comentado el hardware, es conveniente analizar el software de manejo de los dispositivos. Este software está organizado en capas. Las capas inferiores se encargan de ocultar las peculiaridades del hardware y las capas superiores de presentar una interfaz amigable a los usuarios.

Los manejadores de dispositivos (también conocidos como **drivers**) se encargan de aceptar las solicitudes abstractas que le hace el software independiente del dispositivo y ponerse en contacto con el controlador para realizar esa petición.

Si el dispositivo se encuentra ocupado atendiendo otra petición, el manejador se encargará de gestionar una cola de peticiones para darles paso tan pronto como sea posible.

1.7.8 UTILIZACIÓN DEL SISTEMA OPERATIVO

Se puede utilizar el sistema operativo de dos maneras distintas:

- El **modo orden** o **modo comando**: es la interacción del usuario y el sistema operativo a través de una línea de comandos (del tipo de la utilidad *Símbolo de sistema* de Windows o un terminal en Linux). El usuario tiene que teclear la orden que realiza la acción deseada y pulsar **[Intro]** para que el sistema operativo la ejecute, sin ratón, ventanas, iconos, escritorio...

MS-DOS y las primeras versiones de Linux funcionaban en modo orden. Sin embargo, y debido al auge que han tenido los sistemas operativos basados en una interfaz gráfica, se comenzaron a desarrollar entornos gráficos.

```
Ubuntu 8.10 jlr-desktop tty6
jlr-desktop login: _
```

Figura 1.16. Pantalla de terminal en Ubuntu

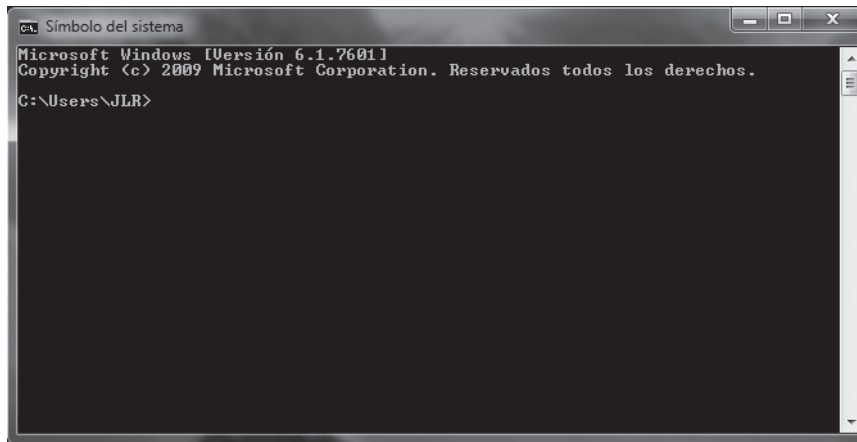


Figura 1.17. Pantalla de Símbolo del sistema en Windows 7

- El **modo gráfico**: es toda aquella interfaz que utilice ventanas, iconos y ratón. Al comienzo de la informática, los ordenadores utilizaban solo el modo orden, una vez que la tecnología lo permitió, la compañía Xerox desarrolló el ratón que en conjunción con un sistema de ventanas ha permitido que el usuario tenga una interacción más amigable con el sistema.

En gran medida, el desarrollo de los entornos gráficos y la facilidad de uso que ellos involucran, ha contribuido al *boom* de la informática. En la actualidad, la mayoría de los sistemas operativos incorporan la visualización de entornos gráficos.

Muchos usuarios avanzados, generalmente programadores, siguen usando el modo texto para todas o algunas de sus tareas, ya que, afirman que el trabajo en modo texto suele ser más rápido, por medio de atajos y complejas combinaciones de teclas para realizar operaciones sencillas como imprimir un documento.

El uso del modo comando en equipos servidores está muy extendido. De hecho, algunos administradores ni siquiera instalan un entorno gráfico en sus sistemas servidores con Linux o Windows Server 2008 (opción *Core*). El uso de un entorno gráfico en un equipo servidor se puede considerar un gasto innecesario de recursos.

1.8 CLASIFICACIONES DE LOS SISTEMAS OPERATIVOS

Según la perspectiva con la que se observen los sistemas operativos, pueden realizarse múltiples clasificaciones. Entre ellas se pueden incluir las siguientes:

1.8.1 POR LOS SERVICIOS OFRECIDOS

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

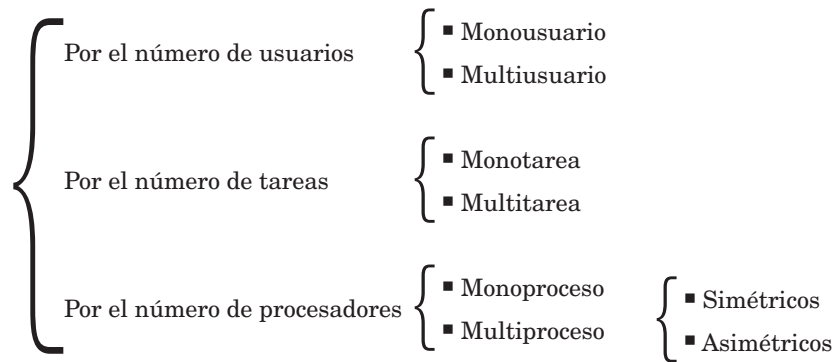


Figura 1.18

Los sistemas operativos **monousuario** o **monopuesto** son aquellos que únicamente soportan un usuario a la vez, sin importar las características de la máquina sobre la que está montado el sistema.

Los sistemas operativos **multiusuario** o **multipuesto** son capaces de dar servicio a más de un usuario a la vez, también independientemente de la plataforma hardware sobre la que esté montado.

Los sistemas **monotarea** son aquellos que solo permiten una tarea a la vez por usuario. Puede darse el caso de un sistema multiusuario y monotarea, en el cual se admiten varios usuarios al mismo tiempo, pero cada uno de ellos puede estar haciendo solo una tarea a la vez.

Un sistema operativo **multitarea** es aquel que permite al usuario estar realizando varios trabajos al mismo tiempo. Es común encontrar en ellos interfaces gráficas orientadas al uso de menús y al ratón, lo que permite un rápido intercambio entre las tareas para el usuario, mejorando su productividad.

Los sistemas **monoproceso** son los que únicamente permiten realizar un proceso a la vez. Sin embargo, permiten simular la multitarea haciendo que el sistema realice una tarea rotatoria con intercambio muy rápido.

Los sistemas operativos **multiproceso** son los que permiten realizar varios procesos simultáneamente y, por tanto, son capaces de ejecutar varias tareas al mismo tiempo.

Dentro de los sistemas multiproceso, se encuentran los sistemas **simétricos**, que son los que distribuyen la carga de procesamiento por igual entre todos los procesadores existentes. Sin embargo, los sistemas multiproceso **asimétricos** asignan una tarea por procesador existente, según su prioridad, y el resto de tareas (de baja prioridad) se ejecutan en un único procesador. Por ejemplo, un sistema biprocesador asimétrico ejecutaría una sola tarea en un procesador y el resto en el otro.

1.8.2 POR LA FORMA DE OFRECER LOS SERVICIOS

En esta clasificación se encuentran:

- **Sistemas centralizados.** Hasta que los computadores personales no tuvieron un precio accesible y suficiente potencia, la mayoría de los sistemas utilizaban el modelo de proceso centralizado. Con este tipo de modelo los computadores *mainframe* se encargaban de todo el procesamiento y los usuarios manejaban únicamente terminales *tontos* (es decir, no disponían de memoria, ni procesador).

Actualmente se siguen utilizando los sistemas centralizados (como los *Terminal Services* de Microsoft) pero los terminales dejan de ser tontos y pueden realizar otras muchas tareas por sí mismos.

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

Los Sistemas Operativos de red usados más ampliamente son: *Novell NetWare*, *Windows Server*, *Linux*, etc.

- **Sistemas distribuidos.** Los sistemas operativos distribuidos son sistemas cuasi-independientes que permiten distribuir los trabajos, tareas o procesos entre un conjunto de procesadores. Puede ocurrir que este conjunto de procesadores se encuentren en el mismo equipo o en equipos distintos (siendo, en este último caso, transparente para el usuario).

Los sistemas operativos distribuidos más extendidos son los siguientes: *Sprite*, *Solaris-MC*, *Mach*, *Chorus*, *Spring*, *Amoeba*, *Taos*, etc.

ACTIVIDADES 1.7



- Indique de qué tipo es el sistema operativo de su propio ordenador.

1.9 SISTEMAS OPERATIVOS ACTUALES

Los sistemas operativos actuales para microordenadores se dividen en dos grupos:

- **Sistemas operativos en red.** Funcionan siguiendo el esquema de un servidor principal que proporciona soporte a las estaciones de la red. Entre ellos se encuentran: *Windows Server 2008* y *Linux*.
- **Sistemas para equipos individuales.** Entre ellos se encuentran: *Windows XP Professional*, *Windows 7* y *Linux*.

1.9.1 WINDOWS SERVER 2008

La familia de servidores Windows 2008 está formada por varias versiones con las siguientes características:

Tabla 1.5 Versiones de Windows Server 2008

WINDOWS SERVER 2008 STANDARD	Incorpora capacidades de virtualización (tener un servidor virtual ejecutándose sobre el mismo ordenador físico) y webs mejoradas. Está disponible en versiones de 32 y 64 bits. Soporta hasta 4 procesadores y 4 GB de RAM (versión de 32 bits) o 32 GB de RAM (versión de 64 bits).
WINDOWS SERVER 2008 ENTERPRISE	Es una plataforma de nivel corporativo para aplicaciones críticas de negocio y virtualización. La disponibilidad del sistema se ve aumentada mediante funcionalidades como el cluster y la adición de memoria en caliente. Está disponible en versiones de 32 y 64 bits. Soporta hasta 8 procesadores y 64 GB de RAM (versión de 32 bits) o 2 TB de RAM (versión de 64 bits).
WINDOWS SERVER 2008 DATACENTER	Es una plataforma de nivel corporativo para aplicaciones críticas de negocio y virtualización a gran escala sobre servidores de todo tipo. La disponibilidad del sistema se ve incrementada con las posibilidades de configuración en cluster, particionamiento dinámico del hardware y adición de procesadores en caliente. Está disponible en versiones de 32 y 64 bits. Soporta hasta 32 procesadores y 64 GB de RAM (versión de 32 bits) y hasta 64 procesadores y 2 TB de RAM (versión de 64 bits).
WINDOWS WEB SERVER 2008	Esta versión es exclusivamente una plataforma Web integrada con los entornos IIS 7.0, ASP.NET y Microsoft .NET Framework, todos ellos rediseñados en profundidad. Está disponible en versiones de 32 y 64 bits. Soporta hasta 4 procesadores y 4 GB de RAM (versión de 32 bits) o 32 GB de RAM (versión de 64 bits).
WINDOWS SERVER 2008 PARA SISTEMAS BASADOS EN ITANIUM	Es la versión para sistemas basados en Intel Itanium optimizada para entornos a gran escala de bases de datos, aplicaciones de línea de negocio o de otro tipo. Está disponible en versión de 64 bits. Soporta hasta 64 procesadores y 2 TB de RAM.
WINDOWS SERVER 2008 STANDARD SIN HYPER-V	Este producto es igual que la versión Standard pero sin incluir la virtualización (Windows Server Hyper-V).
WINDOWS SERVER 2008 ENTERPRISE SIN HYPER-V	Este producto es igual que la versión Enterprise pero sin incluir la virtualización (Windows Server Hyper-V).
WINDOWS SERVER 2008 DATACENTER SIN HYPER-V	Este producto es igual que la versión Datacenter pero sin incluir la virtualización (Windows Server Hyper-V).

La configuración de este sistema operativo se basa en la elección de **roles** o **funciones** (es decir, servicios que la máquina ofrece a los usuarios, como el Servidor Web o el Servidor de archivos) y **características**, que son las prestaciones soportadas, como el Servicio WLAN o el Administrador de almacenamiento extraíble. La elección de un rol activa todo un conjunto de servicios necesarios, sin necesidad de tener que conocer sus detalles ni la necesidad de activarlos manualmente uno a uno.

Al activar cualquier función o característica, el programa de instalación o configuración señala automáticamente los servicios dependientes que deben ser activados. De esta forma, el administrador ve su tarea simplificada, pero, a la vez, tiene información clara y completa sobre los servicios que se van a instalar.

De forma similar, al desactivar o desinstalar un servicio se comprueban las dependencias y se elimina del equipo todos aquellos que ya no sean necesarios.

1.9.2 LINUX

Linux es un sistema operativo multiusuario con todas las características que necesita tener un sistema operativo moderno. De hecho, en los últimos años se ha convertido en una alternativa a los sistemas Windows para computadores basados en arquitecturas Intel y compatibles. Está basado en el sistema operativo Unix y es de libre distribución. Esto significa que Linux se distribuye bajo los términos de licencia GPL (*General Public License*), lo que implica que cualquiera puede libremente copiarlo, cambiarlo y distribuirlo pero sin posibilidad de aplicar restricciones en futuras distribuciones. Además, el código fuente de Linux (escrito principalmente en lenguaje C) es también público y de libre distribución.



¿SABÍAS QUE...?

La historia de Linux empieza en Finlandia, en el año 1991, cuando a un estudiante de la Universidad de Helsinki, llamado Linus B. Torvalds, se le ocurrió comprarse un PC con procesador Intel 386, después de observar que con el sistema operativo MS-DOS no aprovechaba los recursos de la máquina, sistema operativo de entonces: Minix. Sin embargo, debido a las limitaciones de Minix, Linus decidió reescribir algunas partes del sistema, añadiéndole mayor funcionalidad. Posteriormente, Linus decidió difundir el código fuente por Internet, de manera gratuita y con el nombre de Linux (contracción de Linus y Unix).

En los sistemas Linux se puede seleccionar más de un entorno gráfico para trabajar con él. Los entornos que más popularidad tienen son KDE y GNOME. Son los más utilizados debido a su similitud con el entorno de Windows, circunstancia que facilita el aprendizaje de usuarios que ya hayan trabajado con esos sistemas.

A diferencia de otros sistemas operativos, la mayoría de los entornos gráficos de Linux son capaces de gestionar varios escritorios a la vez, aunque en un momento dado solamente se pueda mostrar al usuario uno solo.



¿SABÍAS QUE...?

Unix es un sistema operativo multitarea y multiproceso desarrollado a principios de los años 70 y utilizado principalmente en sistemas servidores. Unix evolucionó durante 20 años tanto en ambientes académicos, como empresariales, lo que lo convirtió en un sistema operativo robusto y fiable.



¿SABÍAS QUE...?

Una distribución de Linux no es más que una recopilación de programas y ficheros (incluyendo la última versión estable del núcleo), organizados y preparados para su instalación. Estas distribuciones se pueden obtener a través de Internet o comprando los CD de las mismas. La mayoría del software que incluyen las distribuciones suele tener licencia GPL, lo que implica que cualquiera puede libremente copiarlo, cambiarlo y distribuirlo pero sin posibilidad de aplicar restricciones en futuras distribuciones.

ACTIVIDADES 1.8



- Busque información en Internet sobre las distribuciones de Linux.
- Indique qué Comunidades Autónomas han desarrollado su propia distribución de Linux y en qué distribución está basada cada una de ellas.

DISTRIBUCIONES

Linux es un sistema operativo de libre distribución. En los primeros tiempos se podían encontrar en multitud de servidores conectados a Internet todos los ficheros y programas necesarios para su funcionamiento. Sin embargo, y debido a que la tarea de reunir todos los ficheros necesarios e instalarlos correctamente podía ser bastante compleja, aparecieron empresas que se dedicaron a hacer este trabajo, y aquí surgió el término de distribución.

Una distribución no es más que una recopilación de programas y ficheros (incluyendo la última versión estable del núcleo), organizados y preparados para su instalación. Estas distribuciones se pueden obtener a través de Internet o comprando los CD de las mismas. La mayoría del software que incluyen las distribuciones suele tener licencia GPL.

Normalmente, la obtención de las distribuciones por Internet suele ser gratuita. Y si se obtiene la distribución mediante los CD, suele tener un coste en general bastante aceptable teniendo en cuenta la gran cantidad de software que incluyen, así como manuales e incluso soporte durante un período de tiempo (de 3 a 6 meses gratuitos).

Actualmente, sería difícil precisar el número de distribuciones existente de Linux, no obstante hay datos que apuntan a que son más de 300.

1.9.3 WINDOWS XP

Cuenta con las siguientes versiones:

- **Windows XP Home:** está destinada al mercado doméstico. Originalmente, no proporcionó soporte para SMP, aunque con los Service Pack se ha utilizado dicha función.
- **Windows XP Professional:** dispone de características adicionales diseñadas para entornos empresariales, como la autenticación por red y el soporte multiprocesador.
- **Windows XP Media Center Edition:** para equipos HP Media Center Computer y la serie Alienware Navigator. Esta versión debía venderse con estos ordenadores y no se podía encontrar en las tiendas por separado.
- **Windows XP Tablet PC Edition:** para ordenadores portátiles diseñados con una pantalla táctil que admitan escritura a mano y pantallas de pequeño tamaño.

- **Windows XP Corporate Edition:** que es similar a Windows XP Professional, pero diseñado especialmente para empresas. Esta edición no está bajo los métodos de ventas tradicionales.
- **Windows XP 64 Bit Edition:** para equipos con procesadores AMD 64 e Intel con extensiones de 64 bits.
- **Windows XP Starter Edition:** destinado a países con pocos recursos o con altos niveles de copia ilegal. Se puede considerar que es un Windows XP con características limitadas.

1.9.4 WINDOWS 7

Cuenta con las versiones siguientes:

Windows 7 Starter Edition

- ✓ Solo para los fabricantes de equipos originales (OEM) que vendan equipos nuevos.
- ✓ Sin interfaz de usuario ni Aero.
- ✓ Solo permitirá ejecutar 3 programas simultáneamente.

Windows 7 Home Basic

- ✓ Solo en mercados emergentes.
- ✓ Sin interfaz de usuario ni Aero.

Windows 7 Home Premium

- ✓ Para fabricantes de equipos originales (OEM) y tiendas.
- ✓ Incluye interfaz de usuario y Aero.
- ✓ Soporte para multitouch.
- ✓ Añade juegos “premium”.
- ✓ Multimedia (Media Center, Reproducción de DVD...).

Windows 7 Professional

- ✓ Para fabricantes de equipos originales (OEM) y tiendas.
- ✓ Incluye todas las características de Windows 7 Home Premium.
- ✓ Mejor capacidad para trabajar en red.
- ✓ Mayor protección de datos, con EFS.

Windows 7 Enterprise

- ✓ Solo para empresas.
- ✓ Incluye todas las características de Windows 7 Professional.
- ✓ Añade BitLocker.

Windows 7 Ultimate

- ✓ Disponibilidad limitada para fabricantes de equipos originales (OEM) y tiendas.
- ✓ Incluye todas las características de Windows 7 Enterprise, pero sin las restricciones de licenciamiento por volumen.

ACTIVIDADES 1.9



- Revise la versión del sistema operativo de su propio ordenador.

1.10 ENCENDER Y APAGAR EL EQUIPO

Encienda el ordenador pulsando el botón de encendido y compruebe que la pantalla se ilumina; si no lo está, pulse el botón de encendido de la pantalla.

Tras encender el ordenador y la pantalla, espere unos segundos para ver en la pantalla las primeras acciones que realiza el ordenador y su sistema operativo.

Una de las acciones que lleva a cabo el sistema operativo mientras el ordenador se pone en funcionamiento es la comprobación de todos los dispositivos internos y externos (funcionamiento y estado de los mismos).

Mientras se enciende el ordenador es muy aconsejable no utilizar ni el ratón ni el teclado y dejar que el ordenador haga las cosas a su ritmo. El puntero del ratón irá alternando su forma entre la de flecha y la de reloj de arena mientras el ordenador y el sistema operativo estén llevando a cabo sus acciones.

Sabrás que el ordenador se ha encendido completamente cuando el puntero del ratón tenga forma de flecha y ya no cambie su forma a reloj de arena.



Si durante la puesta en funcionamiento del ordenador el sistema operativo encuentra un problema de hardware o software, mostrará en pantalla el mensaje correspondiente o una sucesión de pitidos que indicarán el problema producido (deberá mirar el manual de la placa base). Si se indica cómo puede resolver el problema, intente resolverlo, pero si no, la mejor solución es pedir ayuda a alguien que sepa del tema.



El botón de reinicio solo se debe pulsar cuando el ordenador se bloquea y ya no responde a ninguna instrucción. La función del botón es reiniciar el sistema, es decir, cerrar el sistema y volverlo a poner en funcionamiento.

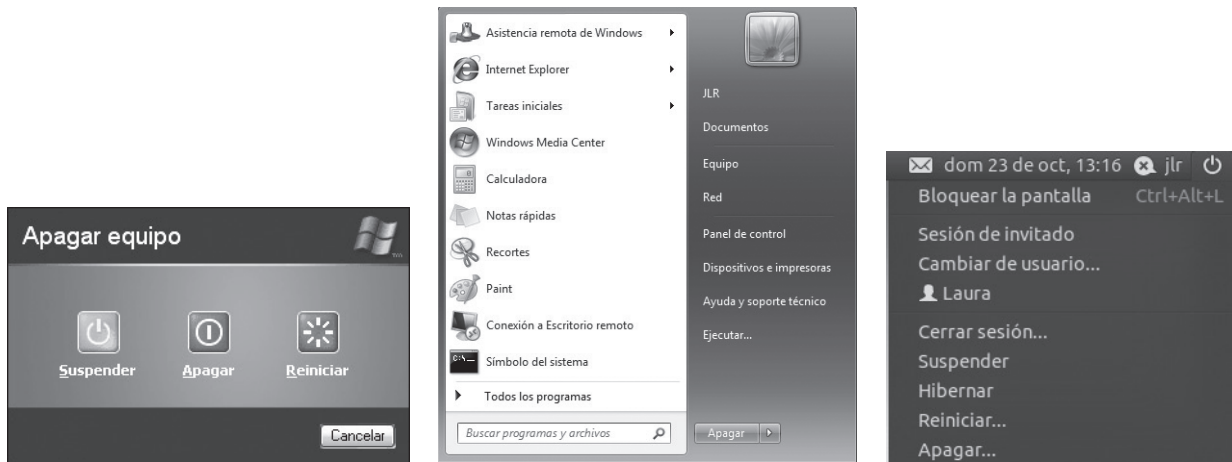


Para apagar el ordenador no utilice el botón de encendido ni lo desconecte de la red eléctrica.



Una posibilidad de apagar el equipo si no responde es pulsar durante unos segundos el botón de encendido. El sistema se apagará (recuerde que los datos que no estén guardados se perderán y, además, pueden originarse errores en el disco).

El proceso correcto para apagar el ordenador es cerrar todas las aplicaciones que se estén ejecutando. Cuando estén cerradas todas las aplicaciones, selecciona **Apagar** en la opción correspondiente del sistema operativo.



Figuras 1.19, 1.20 y 1.21. Apagar desde distintos sistemas operativos (Windows XP, Windows 7 y Ubuntu)

1.10.1 REINICIAR EL EQUIPO

Hay ocasiones en que se necesita reiniciar el equipo sin llegar a que se apague totalmente. Para ello, se puede utilizar la opción **Reiniciar** que se muestra en las Figuras 1.29 y 1.31.

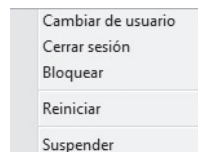


Figura 1.22. Pantalla que muestra Windows 7 al pulsar en el icono que hay a la derecha de Apagar

1.10.2 SUSPENDER O HIBERNAR EL EQUIPO

En los ordenadores actuales es posible, además, poner el equipo en dos estados:

- **Suspensión.** En este modo, la memoria RAM es el único componente que se mantiene alimentado eléctricamente. De esta forma, dado que el estado de los programas se mantiene en memoria, el usuario puede volver a lo que estaba haciendo rápidamente (ver Figuras 1.19, 1.21 y 1.22).
- **Hibernación.** En este modo, el contenido de la memoria se guarda en el disco duro, tras lo que el ordenador se apagará completamente. Al volver a iniciar el ordenador, el usuario se encontrará con todas las aplicaciones que tenía abiertas en el estado en el que se encontraban antes de hibernar (ver Figuras 1.21, 1.23 y 1.24).
- **Suspensión.** Debe utilizarse cuando se va a estar poco tiempo sin usar el ordenador, ya que podrá volver a lo que estaba en segundos, mientras que la **hibernación** se debe utilizar para periodos largos de inactividad, para consumir menos energía y para asegurarse de no perder los datos por algún corte de luz o porque el equipo se quede sin batería en el caso de un portátil.

Para habilitar la hibernación en Windows XP hay que marcar la casilla correspondiente en **Inicio → Panel de control → Rendimiento y mantenimiento** (si está en vista por Categorías) → **Opciones de energía → Hibernación**. Una vez habilitado, si se pulsa **[Mayúsculas]** en la ventana **Apagar equipo** de Windows, la opción de **Suspender** se verá sustituida por **Hibernar**.

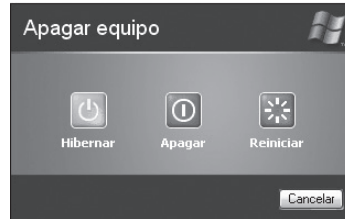


Figura 1.23. Hibernar desde Windows XP

Para habilitar la hibernación en Windows 7 debe ir a **Panel de Control → Sistema y seguridad** (si está en vista por Categorías) → **Opciones de energía → Cambiar la configuración del plan → Cambiar la configuración avanzada de energía → Suspender → Permitir suspensión híbrida**. Una vez habilitado, al pulsar en el icono que hay a la derecha de **Apagar**, verá la pantalla siguiente:

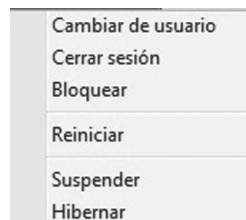


Figura 1.24. Hibernar desde Windows 7

1.11 LOS GESTORES DE ARRANQUE

En caso de que haya instalados varios sistemas operativos en un mismo ordenador, hay que utilizar un sistema para poder seleccionar qué sistema operativo se desea iniciar.

El **gestor de arranque** es un pequeño programa que se ejecuta una vez completado el inicio normal de la BIOS y que permite seleccionar el sistema operativo en caso de disponer de arranque múltiple.

Entre los gestores de arranque se encuentran los siguientes:

- **NTLDR** (*NT Loader*) es el archivo encargado del arranque de los sistemas operativos: Windows NT, Windows 2000, Windows XP y Windows Server 2003. Se encuentra, normalmente, en el directorio raíz del disco del sistema.

Requiere, como mínimo, que se encuentre en dicho directorio raíz: el archivo **boot.ini**, que contiene el menú de opciones de inicio, y el archivo **NTDETECT.COM**, que es el que se encarga de iniciar el sistema seleccionado.

- El **Administrador de arranque de Windows (Bootmgr)** es el archivo encargado del arranque de los sistemas operativos: Windows Vista, Windows 7 y Windows Server 2008. Se encuentra, normalmente, en el directorio raíz del disco del sistema.

Controla el proceso de arranque mostrando el menú multiarranque (si hubiera más de un sistema operativo instalado en el disco). Después, llama al archivo **WinLoad.exe** que es el cargador del sistema operativo Windows y dará paso al archivo **ntoskrnl.exe**, que se encargará del resto del arranque del sistema.



¿SABÍAS QUE...?

El archivo *WinLoad.exe* se encuentra en el directorio `\Windows\system32`.

- **Lilo** (*Linux Loader*) es un gestor de arranque de Linux que permite iniciar este sistema operativo junto con otras plataformas (como Windows) que haya en el mismo ordenador. Funciona en una variedad de sistemas de archivos y puede arrancar un sistema operativo desde el disco duro o desde un disco flexible externo.
- **Grub**. Es un gestor de arranque más moderno y flexible que Lilo, ya que permite que el administrador ejecute cualquier comando desde la línea de comando de Grub. Entre todas sus características hay que destacar la posibilidad de incluir múltiples formatos de ejecutables, el arranque de sistemas operativos no-multiarranque, una agradable interfaz de usuario y una interfaz de línea de comando muy flexible.

ACTIVIDADES 1.10



- Averigüe qué gestor de arranque utiliza el sistema operativo de su equipo.
- Busque información en Internet sobre los cuatro gestores de arranque descritos anteriormente.

1.12 GESTIÓN DE VARIOS SISTEMAS OPERATIVOS EN UN ORDENADOR

Hasta hace poco tiempo, la posibilidad de tener en un mismo ordenador dos sistemas operativos gestionados por un gestor de arranque permitía utilizar uno u otro y conseguir un ahorro en el hardware necesario. Sin embargo, esta utilización no era completa ya que no permitía disponer de ambos los sistemas operativos simultáneamente.

Actualmente, con las máquinas virtuales ha desaparecido esa limitación y es posible disponer de varios sistemas operativos instalados en máquinas virtuales de forma simultáneamente e, incluso, interactuando entre ellos de forma que un usuario no es capaz de distinguir entre un sistema operativo instalado en un ordenador y el que está instalado en una máquina virtual.



RESUMEN DEL CAPÍTULO

En este capítulo se ha indicado lo que es un sistema informático, indicando sus componentes.

Se ha hablado sobre los componentes físicos y lógicos de un ordenador, así como los tipos de licencias de software.

Se ha indicado la forma de representación de la información en los ordenadores, tanto a nivel numérico como textual.

Se ha tratado lo que es un sistema operativo y las funciones que realiza.

Se han descrito en profundidad sus partes, lo que es un bloqueo, la forma en la que gestiona la memoria (fragmentación y segmentación), indicando cómo maneja la memoria virtual (paginación y algoritmo de reemplazo de páginas) y los dispositivos de entrada/salida (controladores, manejadores y acceso directo a memoria).

Así mismo, se han tratado varias posibilidades para clasificarlos en función de los servicios ofrecidos y la forma de ofrecerlos.

También se han indicado algunos de los sistemas operativos actuales para microordenadores, explicando las versiones disponibles.



EJERCICIOS PROPUESTOS

1. Comente brevemente las funciones de un sistema operativo.
2. Explique la diferencia entre programas y procesos.
3. Explique las diferencias entre fragmentación interna y externa.
4. Convierta a binario los siguientes números en base decimal: $123,45_{10}$; $0,234_{10}$; $555,34_{10}$ y $123.432,79_{10}$.
5. Pase a decimal las cantidades: 111.000_2 ; $1.001,1001_2$; $1,01_2$ y $0,10101011_2$.
6. Indique las diferencias entre un sistema operativo monousuario y otro multiusuario.
7. Indique las diferencias que hay entre una licencia OEM y una Retail.
8. Indique lo que son los gestores de arranque.



TEST DE CONOCIMIENTOS



- 1** Indique qué afirmación es verdadera:
 - a) Es posible sumar un número binario y otro hexadecimal.
 - b) Para sumar un número binario y uno hexadecimal es imprescindible convertir previamente ambos números a decimal.
 - c) Es necesario convertir ambos números a una base común antes de operar con ellos.
 - d) Son correctas las respuestas a y b.
- 2** Indique qué afirmación es falsa:
 - a) Un sistema de numeración es una forma de representar cualquier cantidad numérica.
 - b) Cada posición del dígito tiene un valor intrínseco que aumenta de derecha a izquierda según potencias sucesivas de la base del sistema de numeración empleado.
 - c) El sistema de “base 2” utiliza solo dos símbolos, {1,2}, denominados bits.
 - d) El número de símbolos distintos que se pueden usar en un determinado sistema de numeración constituye su “base”, es decir, en base 10 los números que podemos representar son {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}.
- 3** Indique qué afirmación es verdadera:
 - a) Todo sistema informático está compuesto por dos elementos básicos (hardware y software).
 - b) Hay dos tipos de software (de sistema y de aplicación).
 - c) El software de base es un conjunto de programas necesarios para que el ordenador tenga capacidad de trabajar.
 - d) Todas las anteriores son correctas.
- 4** ¿Qué nivel jerárquico se corresponde con el hardware?
 - a) El nivel 0.
 - b) El nivel 1.
 - c) El nivel Lógica digital.
 - d) Son correctas la a y la c.
- 5** Indique qué afirmación es falsa:
 - a) Los sistemas operativos multiusuario son capaces de dar servicio a más de un usuario a la vez, independientemente de la plataforma hardware sobre la que esté montado.
 - b) Un sistema operativo multitarea es aquel que permite al usuario estar realizando varios trabajos al mismo tiempo.
 - c) Un sistema operativo en red es aquel que mantiene a dos o más computadoras unidas a través de algún medio de comunicación (físico o no), con el objetivo primordial de poder compartir los diferentes recursos y la información del sistema.
 - d) Los sistemas monoproceso son los que únicamente permiten realizar un proceso a la vez sin permitir simular la multitarea.
- 6** Indique qué afirmación es falsa:
 - a) Los sistemas operativos de red usados más ampliamente son: Windows Server y Linux Server.
 - b) Los sistemas operativos de escritorio usados más ampliamente son Windows XP, Windows Vista, Windows 7 y Linux Server.
 - c) Windows 7 es un sistema operativo multiusuario.
 - d) Linux es un sistema operativo libre.

7

Indique qué afirmación es falsa:

- a) Un software de dominio público es aquel que no requiere de licencia pues sus derechos de explotación pertenecen a todos por igual y cualquiera puede hacer uso de él, siempre con fines legales y consignando su autoría original.
- b) El software libre siempre es gratuito.
- c) NTLDR es el archivo encargado del arranque de los sistemas operativos: Windows NT, Windows 2000, Windows XP y Windows Server 2003.
- d) Grub es un gestor de arranque más moderno y flexible que Lilo, ya que permite que el administrador ejecute cualquier comando desde la línea de comando de Grub.