

## **Unidad 3. Componentes del software**

- 3.1. Características de un sistema operativo.
- 3.2. Estudio de los diferentes tipos de sistemas operativos.
- 3.3. Principales funciones de un sistema operativo.
- 3.4. Descripción de los sistemas operativos más conocidos (UNIX, DOS).
- 3.5. Programas del sistema (compiladores, cargadores, etcétera).
- 3.6. Proceso de arranque de una computadora personal.

## **Objetivos particulares de la unidad**

Al culminar el aprendizaje de la unidad, identificarás y manejarás los diferentes tipos y aplicaciones de los programas (software) que permiten al usuario realizar sus actividades mediante una computadora.

## Introducción

La presente unidad, al igual que las anteriores y posteriores, tienen como objetivo, ayudarte a reforzar los conocimientos adquiridos a lo largo de la revisión de las lecturas sugeridas en la guía de estudio. Por lo tanto, cuando escuches nuevamente en estos apuntes la palabra “Software”, seguramente te será ya muy familiar.

Los orígenes del Software, se remontan a los años cincuenta, época en que se crearon las primeras computadoras de la primera generación (la parte física o también llamada Hardware). Lo anterior, obligó a buscar una metodología para programar a los mismos, (parte lógica o también llamada Software). Grace Hopper, fue la primera en proponer el uso de subrutinas, que se crearon para el famoso equipo denominado MARK1. Ella misma desarrolló en 1952 un compilador para la UNIVAC. En 1952 el francés E. Holberton desarrolló una subrutina para la misma computadora que leía y almacenaba datos en unidades de cinta. En opinión de algunos, esta fue la primera manifestación de un Software importante para programar a la computadora.

Por los años de los 70's, se crearon programas más en forma como los conocemos en la actualidad, así, aparece el FORTRAN para la IBM 704. Casi al mismo tiempo, surge un nuevo programa para los negocios; el Cobol, convirtiéndose en un gran éxito debido a que se diseñó para fines comerciales y para funcionar en diferentes computadoras. En 1960 este programa pudo compilarse sin ningún problema en la UNIVAC II.

Por los años de los 70's, se crearon programas más en forma como los conocemos en la actualidad, así, aparece el FORTRAN para la IBM 704. Casi al mismo tiempo, surge un nuevo programa para los negocios; el Cobol, convirtiéndose en un gran éxito debido a que se diseñó para fines comerciales y para funcionar en diferentes computadoras. En 1960 este programa pudo compilarse sin ningún problema en la UNIVAC II.

Asimismo, los primeros Sistemas Operativos se desarrollaron por los 50's, y cada vez más han ido evolucionando hasta nuestros días. Al principio se les llamó monitores, posteriormente supervisores y actualmente sistemas operativos. Ejemplos de Sistemas operativos son el DOS, UNIX, LINUX, OS/360 etcétera, hasta los gráficos como los conocemos ahora (Windows).

Sin duda alguna, el Hardware y el Software siempre han estado unidos, ya que difícilmente uno podría operar sin el apoyo del otro.

Como lo vimos en la unidad 2, Los cuatro elementos básicos que componen un sistema de cómputo son: el hardware, el software, los datos y el usuario.



El software es el conjunto de instrucciones (programas) que le indican al hardware (la computadora) lo que debe hacer y que emplea para procesar datos. El Software es un conjunto de programas, documentos, procedimientos, y rutinas asociados con la operación de un sistema de cómputo.

El software se clasifica dependiendo de su aplicación. Hay *software* de sistemas (operativos o intérpretes, indispensables para el funcionamiento de una computadora) y de aplicaciones (Word, Excel, Access, etcétera).

### Software de sistemas



El software de sistemas se define como un conjunto de programas encargados de realizar una función de tipo lógica más que física, que permite la comunicación entre dos o más computadoras entre sí, pero que para ello, también requiere de un medio físico.

A través de la historia y al igual que los programas que estamos acostumbrados a utilizar (software de aplicaciones) han evolucionado, así también el software de sistemas. Ahora existe una amplia gama de los mismos, entre los más comunes encontramos:

- 📁 Ambiente Windows.
- 📁 Quick Link II.
- 📁 Ambiente DOS.
- 📁 UNIX
- 📁 Camaleon.
- 📁 SLIP
- 📁 Winsockets.

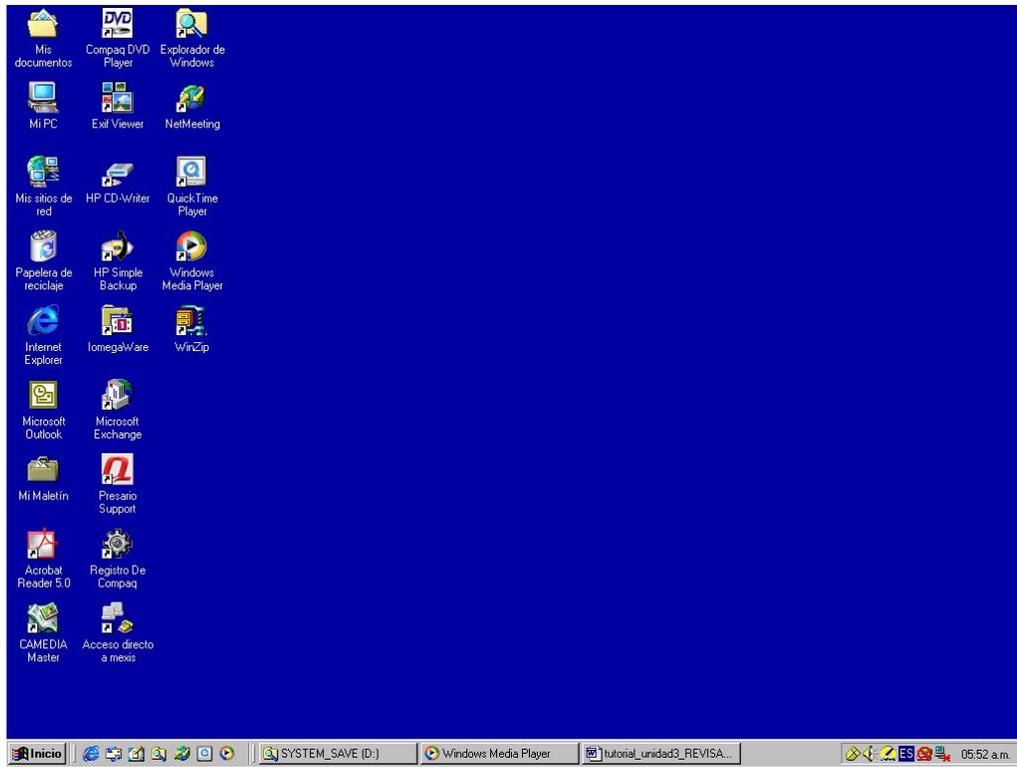


Figura 3.1 Ambiente Windows.

```
C:\>date
La fecha actual es: Vie 16/07/2004
Escriba la nueva fecha: <dd-mm-aa>

C:\>time
La hora actual es: 5:56:10.64
Escriba una nueva hora:

C:\>cd ..

C:\>_
```

Figura 3.2 Ambiente DOS.

## Software de aplicaciones



El software de aplicaciones está más relacionado con aquellos programas que utilizamos cotidianamente y que nos permiten realizar tareas rutinarias y repetitivas para simplificar el trabajo. Así, podemos recibir grandes beneficios a través de éstos programas (que son los más solicitados y utilizados): realizar un escrito, una carta, una nómina, presupuesto, lista de participantes, cálculos, presentaciones etcétera.

Asimismo, todos los programas y los datos que se generan son almacenados de manera separados ya sea en un disco, en una carpeta etcétera, en elementos denominados archivos, así, podemos deducir que existen archivos de programas y archivos de datos.

**Archivos de programas.** Son un conjunto de instrucciones ordenadas que al ejecutarlas, realizan una determinada acción. Los paquetes y aplicaciones especiales, son los archivos de programas, y su finalidad es poder manipular la información.

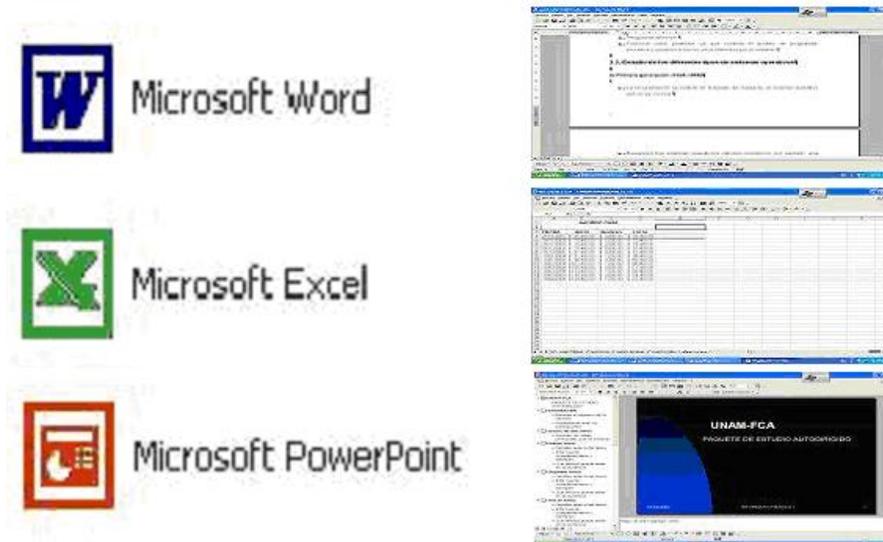


Figura 3.3 Archivos de programas de Office.

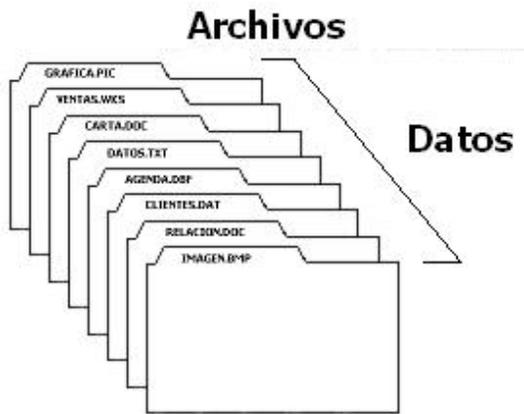


Figura 3.4 Archivos de datos.

**Archivos de datos.** Los archivos de datos es toda aquella información que es manipulada por los programas, no hay una forma directa de acceder a ellos, sino que tiene que ser a través del programa en el cual se generó.

Existen otros tipos de programas considerados también como software de aplicaciones con otras funciones por

ejemplo juegos por computadora. Son muy comunes sobre todo entre los adolescentes y niños, con la intención de brindar un momento de diversión y van desde los más sencillos hasta muy complicados.



Figura 3.5 Simulador de vuelo

Dentro del software de aplicaciones podemos encontrar:

- Edición de textos. Procesadores de Palabras o Texto
- Hojas electrónicas de cálculo
- Manejadores de bases de datos
- De presentación
- Graficadores

- Educativos
- Matemáticos
- Estadísticos
- Simuladores

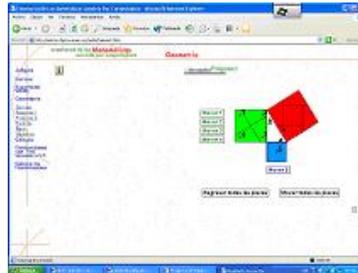
### EDUCATIVOS

<http://fcaenlinea.unam.mx>



### MATEMÁTICOS

<http://www.dgsca.unam.mx>



### SIMULADORES

<http://www.prodigyweb.net.mx/acme77/fisica>

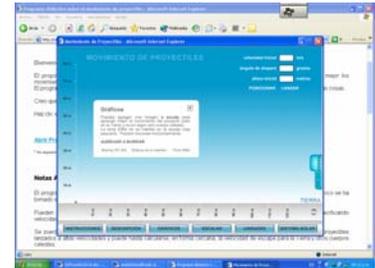


Figura 3.6 Software de aplicaciones.

## 3.1. Características de un sistema operativo



El sistema operativo, es un conjunto de programas que hacen posible la comunicación entre el usuario y la computadora (“interfaz: Se denomina así a todo aquel medio físico que conecta un dispositivo periférico con la computadora; también se le conoce así a todo el software que comunica al usuario con la misma”<sup>1</sup>).

Todas las computadoras tienen o cuentan con un sistema operativo que es el que hace posible la comunicación. Funciona como una especie de traductor que entiende a la máquina y el usuario. Su importancia es vital, pues se encargan de **controlar y administrar** todos los recursos (Es la actividad o capacidad de un componente físico de la computadora. También se denomina así a la actividad de un programa que puede ser utilizada por varios programas de modo concurrente o

<sup>1</sup> <http://www.dgsca.unam.mx>

simultáneo<sup>2</sup>) del sistema (hardware y software), su empleo es indispensable para manejar cualquier otro tipo de programa, simplemente, sin éste tipo de software, no se podría trabajar.

### **Características**

- Actúa como administrador de los recursos de la computadora.
- Controla la ejecución de los programas de usuarios para prevenir errores y uso inapropiado de la computadora.
- Ejecuta programas de usuario y resuelve problemas fácilmente.
- Da herramientas necesarias para que el sistema trabaje de forma eficiente.
- Es responsable de la creación y eliminación de procesos de sistema y de usuarios; detiene y continúa ejecutando un proceso; provee mecanismos para sincronizar, comunicar y proteger procesos.
- Mantiene historial de las partes de memoria a las que se puede tener acceso de manera concurrente.
- Decide qué procesos se cargarán en la memoria cuando haya espacio disponible en ésta.
- Asigna y quita espacio de memoria según las necesidades.
- Administra el espacio libre.
- Asigna espacios de almacenamiento.
- Organiza el disco.
- Proporciona una vista lógica uniforme de la información, independientemente de las características de cada dispositivo.
- Facilitar la entrada y salida de los diferentes dispositivos que se encuentran conectados a la computadora
- Permitir compartir entre usuarios los recursos de hardware con que cuenta una computadora
- Permite la creación y eliminación de archivos y directorios
- Proporciona el soporte para la manipulación de archivos y directorios.
- Facilita la búsqueda de archivos dentro de almacenamientos secundarios.

---

<sup>2</sup> Idem

- Resguarda archivos.
- Funciona como protector, ya que controla el acceso de programas, procesos y usuarios a los recursos definidos por el sistema.

### 3.2. Estudio de los diferentes tipos de sistemas operativos



Como ya se mencionó antes, los programas de sistemas y los de aplicaciones fueron evolucionando a través de la historia, los sistemas operativos de las primeras computadoras, eran muy difíciles de manejar, además de tener limitaciones. Esta evolución a través de la historia nos obliga a clasificarlos por generaciones.

#### A) Primera generación (1945-1955)

- La programación se realiza en **lenguaje de máquina**, el sistema operativo aún no se conoce.
- Resuelven (los sistemas operativos) cálculos numéricos, por ejemplo, una tabla con las funciones seno y coseno.
- Aparece el concepto de nombres de archivo del sistema para lograr independencia de información

#### B) Segunda generación (1955-1965)

- Los programadores escriben sus programas primero en papel (**fortran o assembler**) y lo pasan a tarjetas perforadas, luego le entregan la tarjeta perforada al operador.
- El operador le entrega, por impresora, al programador la salida.
- Surge el sistema por lotes, que consiste en juntar lotes de tareas similares que se leen con el lector de tarjetas perforadas y se almacenan en una cinta. Tarea que se realiza en una máquina económica, de baja capacidad de procesamiento, útil para leer tarjetas, copiar cintas e imprimir.

### C) Tercera generación (1956-1980)

- Surgen los sistemas de modos *múltiples*, que soportan simultáneamente (dos procesos son simultáneos cuando se ejecutan en el mismo instante; es decir, para que exista simultaneidad entre  $n$  procesos se debe contar forzosamente con  $n$  procesadores<sup>3</sup>) procesos por lotes, tiempo compartido, procesamiento en tiempo real y multiprocesamiento
- Aparecen nuevas técnicas como multiprogramación y sistemas de tiempo compartido. **La multiprogramación** consiste en dividir la memoria en sectores y cargar varios procesos, cada uno en un sector de memoria; cuando uno de ellos necesita esperar que una operación de entrada-salida finalice, puede utilizar el procesador otra operación. Si la cantidad de procesos que se cargan en memoria simultáneamente son suficientes, el procesador permanece ocupado casi el 100 por ciento del tiempo.
- En sistemas de tiempo compartido, cada usuario tiene una terminal *on-line*, donde puede enviar comandos y ver su salida en forma interactiva. Igual que en multiprogramación, se cargan varios procesos a la vez en memoria.

### D) Cuarta generación (1980-a la fecha)

- *Software* para computadoras personales.
- Sistemas multiprocesadores (multiprocesamiento simétrico y asimétrico). Los procesadores comparten memoria y hacen copia del sistema operativo. Cada uno de ellos tiene asignada una tarea específica. Además, hay un procesador *master* que da tareas a los procesadores esclavos (es decir, comparten memoria y periféricos).
- Se incrementan la capacidad de procesamiento y confiabilidad.
- Hay sistemas operativos para redes, donde el usuario puede dar órdenes a una computadora en forma remota o copiar archivos de una computadora a otra, etcétera.
- Sistemas distribuidos. Cada procesador tiene su memoria y reloj local, y se comunican con otros procesadores a través de una red (líneas telefónicas o

---

<sup>3</sup> Idem

buses de alta velocidad). Los procesos se ejecutan en distintos procesadores, unidos mediante una red, sin que el usuario se entere.

- Incursionan los sistemas en tiempo real (utilizados cuando las tareas deben realizarse dentro de un límite de tiempo). Hay dos tipos de sistemas en tiempo real: *hard* y *soft*. Los primeros requieren que el procesamiento se complete dentro de un límite de tiempo; si no se satisface, el sistema falla. Los segundos son menos estrictos con los límites de tiempo y ejecutan el proceso en lapso real.

Como podemos observar durante las diferentes generaciones de los sistemas operativos se dieron características que nos permiten hacer una clasificación:

**Sistemas operativos por lotes.** Los sistemas operativos por lotes requieren que la información esté reunida en bloque o "lote" (el programa, los datos, y las instrucciones). Los trabajos son procesados en el orden de admisión, según el modelo de "primero en llegar primero en ser atendido". En estos sistemas la memoria se divide en dos zonas. Una de ellas es ocupada por el sistema operativo, y la otra se usa para cargar programas transitorios para su ejecución. Cuando termina la ejecución de un programa se carga un nuevo programa en la misma zona de memoria.

**Sistemas operativos multiusuario.** Los sistemas operativos multiusuario permiten acceder simultáneamente a un sistema de computadoras a través de dos o más terminales. Este tipo de sistema operativo es fundamental en el manejo de redes de computadoras actualmente

**Sistemas operativos multiprogramación.** Los sistemas de multiprogramación son capaces de soportar dos o más procesos concurrentes múltiples, permiten que residan al mismo tiempo en la memoria primaria las instrucciones y los datos procedentes de dos o más procesos. Estos sistemas implican la operación de multiproceso, para el manejo de la información. Se caracterizan principalmente por

un gran número de programas activos simultáneamente que compiten por los recursos del sistema, como el procesador, la memoria, y los "dispositivos de E/S". Estos sistemas monitorean el estado de todos los programas activos y recursos del sistema

**Sistemas operativos de tiempo compartido.** Los sistemas operativos de tiempo compartido tratan de proporcionar un reparto equitativo de los recursos comunes para dar la impresión a los usuarios de que poseen una computadora independiente. En estos sistemas el administrador de memoria proporciona aislamiento y protección de los programas, ya que generalmente no tienen necesidad de comunicarse entre ellos. El control de E/S se encarga de proporcionar o retirar la asignación a los dispositivos de forma que se preserve la integridad del sistema y se proporcione servicio a todos los usuarios. El administrador de archivos proporciona protección y control en el acceso de la información, dada la posibilidad de concurrencia y conflictos al tratar de acceder a los archivos

**Sistemas operativos de tiempo real.** Estos sistemas tienen como objetivo proporcionar tiempos más rápidos de respuesta, procesar la información sin tiempos muertos. En estos sistemas el administrador de memoria es relativamente menos solicitado debido a que muchos procesos residen permanentemente en memoria. El administrador de archivos se encuentra normalmente en grandes sistemas de tiempo real y su objetivo principal es manejar la velocidad de acceso, más que la utilización eficaz del almacenamiento secundario.”<sup>4</sup>

**Por su nivel, los sistemas operativos pueden clasificarse en:**

Bos (*basic operating system*), Dos y Tos (*disk y tape operating system*), Os (*operating system*); por su tipo, en: Pcp (*primary control program*), Mft (*Multiprogramming with fixed number of tasks*) y Mvt (*multiprogramming with variable number of tasks*). El nivel se refiere al tipo de facilidades que ofrece el

---

<sup>4</sup> Idem

sistema. Así, Bos es el nivel más bajo y simple; aunque requiere de menos memoria y su optimización de recursos es baja. Os, por su parte, representa un nivel más avanzado y requiere mayor memoria, es útil en grandes sistemas y alcanza más optimización. El Pcp permite operar una sola tarea a la vez; y el Mft, hasta 15 programas diferentes simultáneamente. El Mvt es similar al Mft, pero las participaciones se definen y asignan en forma automática por el propio sistema operativo.

Algunos ejemplos de sistemas operativos más actuales son: Dos, Unix, Windows, Macintosh y OS/2.

### **3.3. Principales funciones de un sistema operativo**

- Funciona como interfaz para el usuario, *software*, *hardware* y para la administración de archivos.
- Administra todos los programas que se están ejecutando.
- Permite a los programas compartir información.
- Hace posible la multitarea (sobre todo en sistemas más recientes).
- Sigue la pista de todos los archivos de cada disco.
- Deja a los usuarios crear sus propios archivos y directorios.
- Mantiene organizada la comunicación con el CPU y las otras piezas que conforman el *hardware*.
- Proporciona el *software* necesario para enlazar computadoras y formar una red.
- Proporciona utilerías para llevar a cabo determinadas funciones como desfragmentar archivos, respaldos, recuperación de datos, protección contra virus y de pantalla.



Figura 3.7 Funciones del Sistema Operativo.

### 3.4. Descripción de los sistemas operativos más conocidos (Dos, Unix)

#### A) DOS (*Microsoft disk operating system*)

Conjunto de programas que hacen posible la comunicación entre el usuario y la computadora.

##### a) Características

- Sólo se puede cargar en la memoria un solo programa a la vez.
- Diseñado para manejar bajas cantidades de memoria (640 KB).
- Hecho para CPU de 8 y 16 *bits*.
- Obliga a la computadora de alto rendimiento a trabajar por debajo de su capacidad.
- Los nombres de los archivos se limitan a ocho caracteres más una extensión de tres después del punto (todos los archivos necesitan de un nombre para ser identificados, y este nombre consta de dos partes separadas por un punto).

Ejemplo:

| Nombre        | Extensión   |
|---------------|-------------|
| <b>NOMINA</b> | <b>.DOC</b> |

Figura 3.8 Las partes del nombre de un archivo.

El que los archivos cuenten con una extensión es de gran ayuda, ya que esta, nos permite identificar el tipo de archivo de que se trata y en qué programa fue creado (archivo de texto, de hoja de cálculo, de presentación y otros)

|  |      |                 |
|--|------|-----------------|
| Archivos de programas<br>(ejecutables)         | .EXE |                 |
|  | .COM |                 |
|  | .BAT |                 |
| Archivos de datos<br>(a través de un programa) | .DBF | Base de datos   |
|  | .WK1 | Hoja de cálculo |
|  | TXT  | Texto           |

Figura 3.9 Diferentes tipos de extensión del nombre de un archivo.

- La manera como el usuario puede comunicarse es a través de una serie de comandos para realizar acciones tales como copiar, borrar, crear un directorio, leer un disco, etcétera.

```
C:\>date
La fecha actual es: Vie 16/07/2004
Escriba la nueva fecha: <dd-mm-aa>

C:\>time
La hora actual es: 5:56:10.64
Escriba una nueva hora:

C:\>cd ..

C:\>_
```

Figura 3.10 Sistema operativo DOS.

## b) Comandos

Los comandos son órdenes se le da a la computadora y el sistema operativo tiene como tarea interpretarlos y realizar la acción correspondiente.

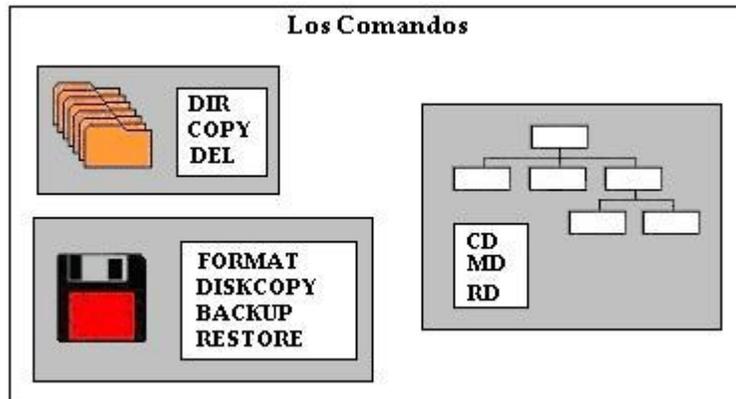


Figura 3.11 Comandos del sistema operativo.

- Internos: se encuentran en el archivo *COMMAND.COM* y al iniciar el sistema son cargados en memoria.
- Externos: se almacenan en disco, en archivos separados; es decir, cada comando externo es un archivo.



La diferencia entre estos dos tipos de comandos radica en que los internos están registrados en la memoria *ROM* de la máquina, y los externos se encuentran grabados en discos de almacenamiento secundarios (fijos o removibles). Asimismo, otra diferencia es la velocidad con la que se ejecutan los comandos de uno y otro.

Los comandos generales que maneja el Dos son los siguientes (cada cual desempeña una función específica y pueden ser comandos de procesamiento por lotes, para *config.sys*, internos o externos):



|  |  |  |
|--|--|--|
|  | configuración de teclado   | del DOS  |
| <u>LABEL.</u> Poner una etiqueta al disco                            | <u>FIND.</u> Busca y presenta todas las ocurrencias de una cadena de caracteres                              | <u>MEM.</u> Verifica la capacidad de la memoria                          |
| <u>MORE.</u> Presenta la información de salida pantalla por pantalla | <u>MKDIR.</u> Crea un subdirectorío por debajo del directorío actual   | <u>RESTORE.</u> Restaura archivos de discos creados con la opción Backup |
| <u>RESTORE</u>   | <u>REPLACE.</u> Reemplaza selectivamente archivos en el directorío destino por los archivos del mismo nombre | <u>RENAME.</u> Cambia el nombre de los archivos                          |
| <u>RMDIR.</u> Elimina directorios vacíos                             | <u>PROMPT.</u> Modifica el símbolo del sistema del DOS   |  |

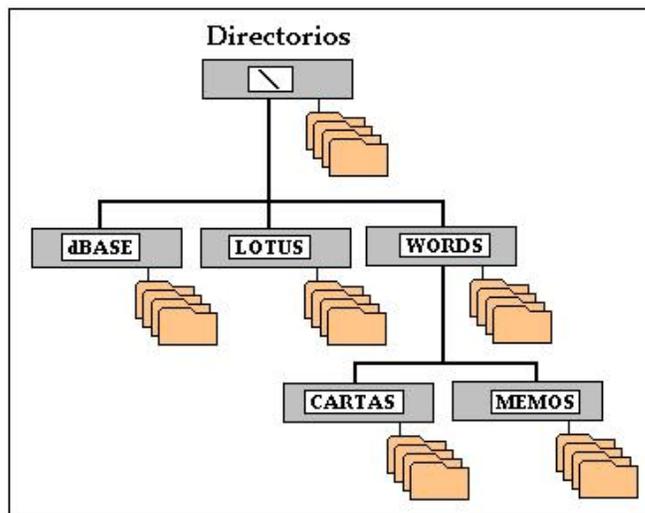


Figura 3.13 Estructura de directorios.

Uno de los comandos más conocidos del MS-DOS, es el **DIR**, que nos muestra el contenido de un disco, directorio o subdirectorio. Tiene la misma función que en Windows al crear una carpeta. Su función consiste en hacer una separación de los archivos de manera organizada.

Otro punto importante del Sistema operativo MS-DOS, es la ruta de acceso (paths). Éstos señalan la ruta de acceso a determinados archivos, dentro de un directorio. Ejemplo:

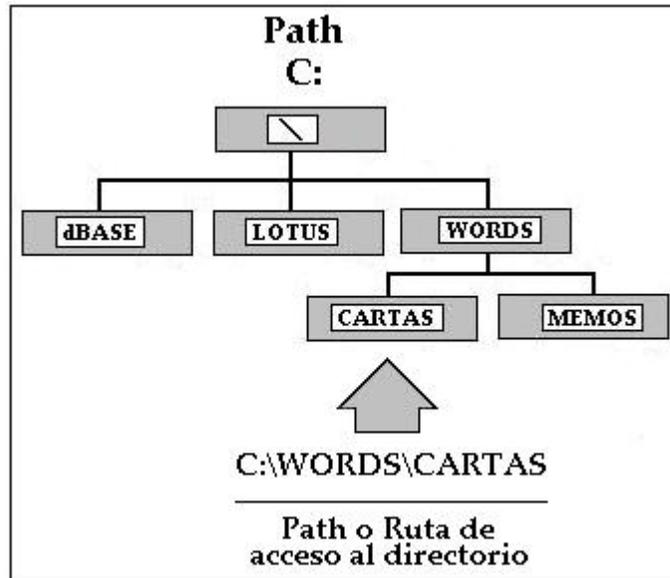


Figura 3.14 Ruta de acceso a los archivos.

La sintaxis de un comando se estructura de la siguiente manera:

Ejemplo:



Figura 3.15 Sintaxis del comando COPY.

### c) Teclas de edición de comandos

- F3. Al oprimir esta tecla una sola vez, el sistema operativo nos presentará el comando o tarea ordenada más recientemente.
- Ins. Si al teclear un comando cometemos el error de oprimir una o varias letras, esta tecla nos permite insertar las letras omitidas, si colocamos el cursor en la posición deseada y la pulsamos una sola vez.
- Del. Su función inversa a Ins; nos permite eliminar letras o caracteres, sólo que en este caso habrá de colocar el cursor en la posición del carácter a eliminar y pulsarla cada vez que sea necesario.

- → ← Al pulsar estas teclas se puede desplazar el cursor a la derecha o a la izquierda, siempre y cuando existan caracteres. El límite de la izquierda lo establece el *PROMPT* y el de la derecha, el último carácter del más reciente comando.

#### **d) Comodines**

Trabajando con MS-Dos, en algunos de sus comandos (*DIR*, *COPY*, *DEL*, entre otros), tenemos la alternativa de utilizar dos signos especiales con función de comodín (que pueden reemplazar uno o todos los caracteres de los nombres de archivos o directorios).

- ? Este signo se utiliza para reemplazar un carácter en los nombres. Por ejemplo, si damos la instrucción *DIR C?.TXT*, la respuesta que se nos presentará será el listado de todos los archivos que comiencen con la letra C y cuya extensión sea *TXT*.
- Se emplea para reemplazar uno o todos los caracteres anteriores o posteriores a la posición que ocupe dentro del nombre o en la extensión. Ejemplo: *DIR \*.DBF*, indica que nos muestra todos los archivos cuya extensión es *DBF*.

#### **e) Editor del sistema**

*EDIT*, es uno de los programas de MS-Dos –otro es *EDLIN*– que permite crear, abrir, editar, guardar o imprimir archivos de texto en código ASCII.

#### **f) Código ASCII**

Es una serie de símbolos (para tener la representación de cada uno de ellos, necesitamos conocer una especie de claves). Por ejemplo, para traer el símbolo de diagonal invertida \, el usuario tiene que oprimir la tecla de Alt y el número 92 del teclado numérico; para acentuar vocales oprimiremos Alt y el número 160, para á; 161, í; 62, ó; 163, ú; y 130, é. A través de código ASCII, también podemos traernos diferentes símbolos:

$\Sigma, \{, \varepsilon, \Pi, \infty, \geq, \Psi, \mathfrak{R}, \Omega, \Xi$

## B) Unix

Sistema operativo multitarea, desarrollado por AT&T en 1960. Es un sistema muy potente, se usa para operar supercomputadoras, mainframes y minicomputadoras, así como PC y estaciones de trabajo potentes. Generalmente, presenta una interfaz de línea de comandos, aunque también puede mostrar ambiente operativo gráfico. Debido a la capacidad de Unix para trabajar con tantas clases de *hardware*, se convirtió en columna vertebral de Internet. Además, requiere de muchos comandos y es más difícil de operar.

### a) Características

- Sistema de ficheros jerárquico en el que todo se encuentra anclado en la raíz.
- El sistema de ficheros está basado en la idea de volúmenes, que pueden montarse y desmontarse (por eso se les asigna un nodo del árbol como punto de anclaje).
- Realiza un riguroso control de acceso a ficheros. Cada uno está protegido por una secuencia de *bits*. Sólo se permite el acceso global al *root* o superusuario.
- Compatibilidad de todos los procesos de entrada y salida. Para Unix, el universo es un sistema de ficheros. De esta forma existe compatibilidad entre ficheros, dispositivos, procesos, *pipes* y *sockets*.
- Presenta comandos de usuario (a nivel de *shell*) para iniciar y manipular procesos asíncronos concurrentes. Un usuario puede ejecutar varios procesos, intercambiarlos e interconectarlos a través de *pipes* o tuberías
- Sistema operativo de red.
- Sistema multiusuario y de tiempo compartido.
- El usuario introduce comandos y recibe resultados en una terminal.
- Todo usuario dispone de un directorio privado llamado *home directory* sobre el que, exceptuando al *root*, sólo él tiene control.

- El sistema proporciona, también a nivel de *shell*, importantes facilidades para las comunicaciones entre usuarios y máquinas, dentro y fuera del propio sistema.
- Permite alto grado de customización o de particularización del entorno, según preferencias de cada usuario, a través de ficheros de configuración particulares.

## b) Comandos

- *Login*. Para entrar hay que disponer de una cuenta. En caso de ser un sistema recién instalado, no está creada aún ninguna cuenta excepto la de *root*, por lo que habrá que contestar tecleando *root*. En otro caso, se teclea el *login* o identificativo correspondiente. Tras ello y para comprobar la identidad del usuario, el sistema pregunta.
- *Password*. Esperando un *password* o palabra clave asociado al *login* y, si todo va bien, ya se está dentro. Mientras se teclea la palabra clave, no aparece eco en pantalla –para que nadie que esté en rededor del usuario pueda leerla–. Por ello, si se produce un error al teclear, no habrá otra oportunidad. Si una vez dentro se desea salir, bastará teclear *exit*, *logout* o simplemente CONTROL+D, según el sistema.
- Nada más terminar el proceso de *login*, una *shell* arranca automáticamente y advierte de su disposición a recibir comandos mostrando un *prompt*, que por defecto será uno de los símbolos #, %, > ó \$, en función de qué *shell*. Este *prompt* equivale al famoso C:> del sistema Dos.
- Ahora, ya desde dentro, puede empezarse a jugar con algunos comandos.
- *Echo*. Su finalidad es mostrar mensajes, es decir, presenta un eco de sus argumentos en pantalla de modo idéntico a como funciona el (eco) del Dos. La diferencia estriba en que, al igual que la mayoría de los comandos Unix, dispone de gran número de opciones.
- *Hostname*. Indica el nombre de la máquina que aloja el sistema. No se trata del nombre del *hardware*, sino del de todos los sistemas Unix. Esto se debe a que

Unix es un sistema en red, lo que obliga a tener bien identificadas las máquinas.

- *Who*. Muestra una lista de los usuarios que se encuentran conectados en ese momento en el sistema.
- *who am i*. Su función es informarle a uno quién es el usuario. En cuanto uno se mueve un poco por el sistema, siempre que disponga de más de una cuenta, descubrirá que es un comando realmente necesario.
- *Ls*. Abreviatura de *list*, muestra el contenido de un directorio. Su funcionamiento y sintaxis es análoga al *dir* del Dos.

En Unix, los nombres del fichero no están sujetos a las reglas del sistema Dos, que los limita a 8 caracteres y una extensión de 3 separados por un punto, sino que pueden tener una longitud entre 16 y 256 caracteres en función de la versión.

Por otra parte, Unix cuenta con los mismos comodines del Dos: \* y ?. Sin embargo, para Unix, hay un fichero cuyo nombre comienza por el carácter punto (.), como *.profile*, es un fichero oculto y únicamente será listado si se utiliza la opción -a.

*Cat*. Muestra el contenido de un fichero del mismo modo que el *type* del Dos. Equivale teclear *cat .login*, en Unix, y *type .login* en Dos (aunque *.login* no es nombre válido en Dos).

*Date*. Muestra la hora del sistema de forma análoga a como lo hace Dos.

*Passwd*. Cambia el *password* del usuario en la máquina en que lo ejecuta. Es importante utilizar este comando frecuente para proteger el sistema ante el potencial ataque de *hackers*.

*Man*. Es el comando más importante tanto para el usuario principiante como para el experimentado, incluso para el *root*. A través de él, se puede acceder al manual completo en línea del sistema.

### 3.5. Programas de sistema (compiladores, cargadores, etcétera)

Dentro de la clasificación de software de sistemas, se encuentran los programas de carga o inicio, los sistemas operativos, los compiladores, los intérpretes y los controladores de dispositivos.



odos los lenguajes de alto nivel requieren de un compilador o de un intérprete.

Un compilador es programa que convierte los lenguajes de alto nivel en instrucciones de lenguaje de máquina.

En otras palabras, traducen los programas que fueron escrito en algún lenguaje de programación de alto nivel, a lenguaje de máquina(que es el que entienden las computadoras). Esto es porque cuando el lenguaje entra a la computadora, el sistema operativo debe utilizar un compilador para traducirlo al lenguaje de máquina. El compilador, analiza el lenguaje utilizado y hace la traducción sentencia a sentencia del programa fuente y verifica que no haya errores. El resultado es un programa objeto que para que sea ejecutado por la computadora, debe pasar por otra etapa de procesamiento llamado

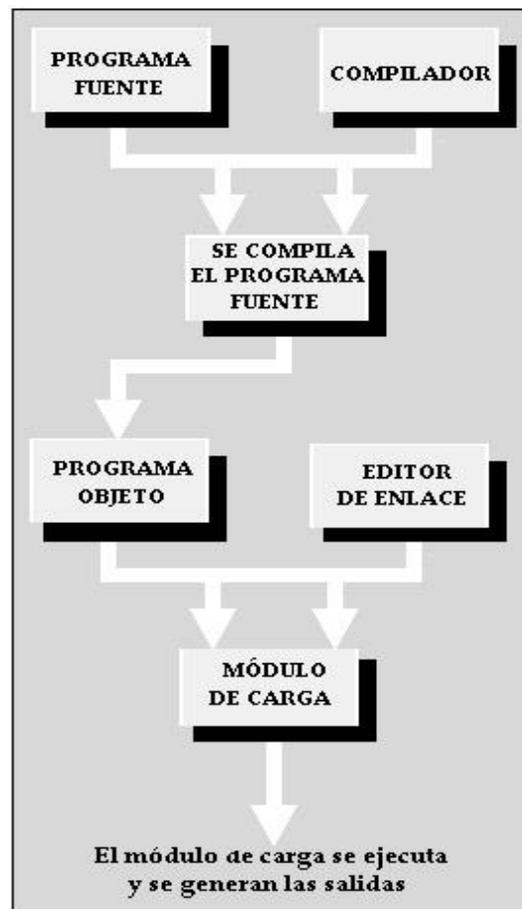


Figura 3.5.1 Proceso de compilación de un programa.

módulo de carga en donde se agregan instrucciones de control que se requieren (editor de enlace). A continuación el programa se encuentra listo para que la computadora lo ejecute.

Durante la compilación de un programa de alto nivel, el compilador se da a la tarea de revisar cada línea de programa, con el fin de detectar errores de sintaxis, por eso es tardado el proceso. Cualquier punto que no coincida con el lenguaje lo mostrará al usuario. Por su lado, un cargador es un programa del sistema que realiza la función de carga consiste en un juego de instrucciones que permiten al dispositivo de entrada (teclado ó unidad de cinta) asignar la dirección de inicio de la memoria y asegurar que el computador leerá el programa y lo cargara byte a byte. Muchos cargadores también incluyen relocalización y ligado. Un ligador es un programa de sistema que combina dos o mas programas objeto separados y permite que se hagan referencias unos a otros, o sea, que cada uno de estos programas pueden hacer referencia a código ó variables de los otros programas con los que está enlazado.



Las funciones más importantes de un cargador son: colocar un programa objeto en la memoria e iniciar su ejecución. Si tenemos un cargador que no necesita realizar las funciones de ligado, su operación es simple pues todas las funciones se realizan en un solo paso.

El avance y desarrollo de los lenguajes de programación, trajo consigo a los llamados interpretes, que también se encargan de realizar un proceso de traducción pero son más lentos que los compiladores ya que no producen un código objeto.



diferencia de un compilador, verifica, traduce y ejecuta los programas línea por línea, en lugar de crear un programa objeto, es decir no crea un programa en lenguaje de máquina, sino que prueba, y compila el programa (lo vuelve a lenguaje de máquina) en el momento en que se escribe.

Cuando la línea se lee por segunda vez, como en el caso de los programas en que se reutilizan partes del código, debe compilarse de nuevo. Aunque este proceso es más lento, es menos susceptible de provocar fallos en la computadora.

El proceso de traducción consiste en convertir una cadena del lenguaje S llamado lenguaje fuente a una cadena equivalente o conjunto de cadenas en otro lenguaje conocido como objeto. Por ejemplo, si se escribe un programa en C, diseñado para aceptar determinada entrada y producir una salida, se espera que un compilador produzca un código objeto que acepte y produzca dichas la entrada y la salida respectivas. Esta noción es conocida como equivalencia entrada-salida.

En los lenguajes de programación, donde es posible definir una semántica formal, la corrección del proceso de traducción puede ser asegurada. El programa genscript de UNIX es un traductor cuya entrada son archivos de texto ASCII para producir postscript, lenguaje de descripción de páginas, o bien un "navegador" (browser) de WWW como Netscape que interpreta archivos HTML. Puede concluirse que la traducción, en el sentido más amplio, constituye el propósito de muchos de los programas que se usan o diseñan.

### **3.6. Proceso de arranque de una computadora personal**

Antes de iniciar el proceso de arranque o levantamiento del sistema, es conveniente que se revisen las conexiones del equipo antes de encenderlo, verificando que todos los cables estén en su posición y en buen estado. Es necesario que el monitor y el CPU estén conectados para la transmisión de datos de salida, también el teclado debe estar conectado para introducir la información.

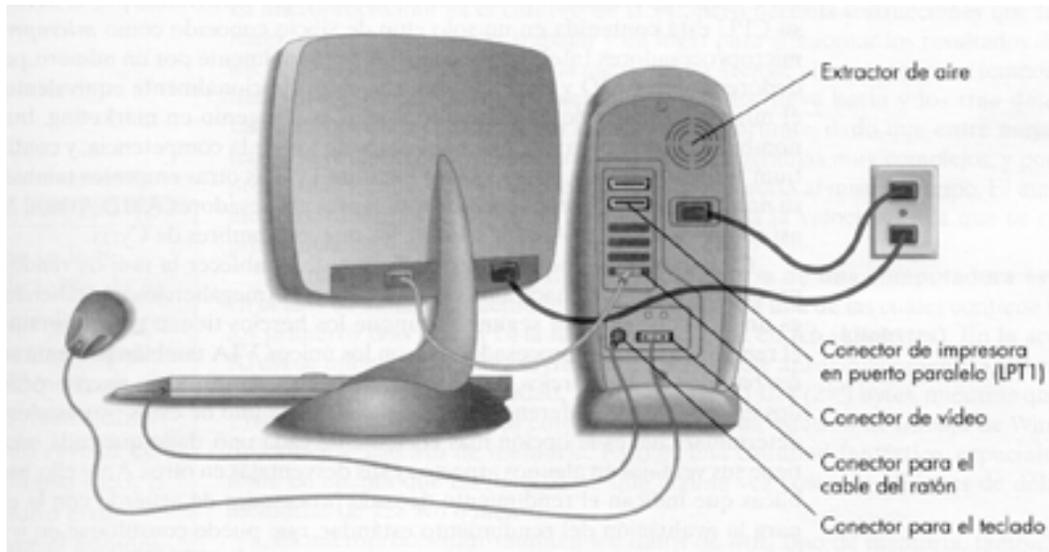


Figura 3.6.1 Conexiones de una computadora.

Todos los dispositivos periféricos se conectan al CPU, pues en él se encuentran los elementos que manipulan la información.

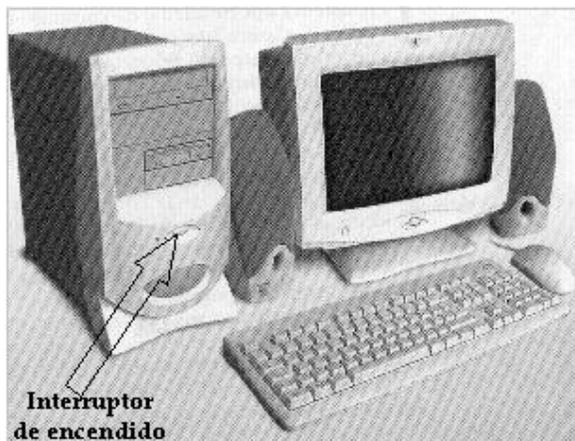


Figura 3.6.2 Proceso de arranque de una computadora.

El procedimiento de arranque de una computadora comienza con el encendido del equipo, que implica prender el monitor y el CPU (en éste orden)

Hay dos maneras como arranca el sistema de una computadora personal.

1. En frío: está apagado y lo arrancamos con el *ON*. Realiza un diagnóstico interno automático del equipo, *cachea* los dispositivos de entrada y salida, teclado, monitor, discos duros, disquetes, memoria (RAM), etcétera. Después, busca el S.O. y lo carga; primero, mira en la disquetera, y si hay disquete con sistema operativo lo arranca; después, mira el disco duro, tomándolo del lugar donde lo encuentre primero. Si no lo

encuentra, manda un mensaje a la pantalla indicando que el disco no tiene sistema operativo.

2. En caliente: se ejecuta apagando el ordenador con el botón *reset* o con *Ctrl+Alt+Supr.* No se llevan a cabo todos los pasos anteriores y se carga directamente el S.O., suele emplearse tras caídas del sistema operativo o bloqueo del ordenador.

En el proceso de arranque intervienen distintos archivos, cada cual con una misión específica.

Módulo BIOS (*Basic Input Output System*). Específico de cada fabricante, se encarga de controlar las diversas unidades *hardware* de entrada y salida (teclado, impresoras, reloj...). Durante el inicio del sistema, el *BIOS* se lee y guarda en la *RAM* cargando el fichero *IO.SYS*. Este fichero no se ve al hacer un *dir*, ya que tiene el atributo de oculto *hidden* y el de sistema *system*, que indica al sistema que no es un archivo normal y no puede modificarse, editarse, etcétera.

El *IO.SYS* es el *Kernel* o núcleo del sistema y realiza funciones como la gestión de ficheros, registros de memoria, generación de otros programas, etcétera. Es independiente del *hardware* y contiene una serie de servicios del sistema.

El *COMMAND.COM* (procesador de órdenes). Se responsabiliza del análisis gramatical y la gestión de órdenes del usuario. Se suministra por defecto con el MS-Dos, pero es posible crear uno propio poniendo una orden en el *config.sys*.

*CONFIG.SYS*. Es un fichero de sistema. El Dos lo ejecuta antes del *autoexec.bat* y contiene comandos de configuración del equipo. Según las tareas, es posible que necesitemos configuración distinta. Sin este archivo o sin algún comando suyo el sistema no funciona. Carga el país (*country*) y los dispositivos (*device*), establece el máximo de archivos a abrir a la vez (*files*), zonas temporales de memoria (*buffers*)...

*AUTOEXEC.BAT*. De procesamiento de lotes. Se ejecuta antes de llegar a nosotros. Inicia opciones. Cambio de fecha y día, principia actividad del ratón. Cambia memoria extendida a expandida. Definición de teclado (*keyb*).