

SISTEMAS OPERATIVOS

6.1. Concepto de sistema Operativo. Introducción histórica.

Un Sistema Operativo es un programa que actúa como interfase entre el usuario de un computador y el hardware del mismo, ofreciendo el entorno necesario para que el usuario pueda ejecutar programas. Su principal cometido es facilitar el uso del sistema informático, siendo su objetivo secundario permitir la utilización del hardware del computador en modo eficiente. Así pues, un Sistema Operativo es el soporte lógico que controla el funcionamiento del equipo físico, la eficiencia, fiabilidad, facilidad de mantenimiento, y un pequeño tamaño serían las características deseables de este software.

El sistema operativo es una parte importante de todo sistema informático que engloba a:

- El hardware: CPU, memoria y dispositivos de entrada y salida (E/S).
- El sistema operativo.
- Los programas de aplicación (compiladores, sistemas de bases de datos, utilidades, videojuegos, programas de gestión, etc.).
- Los usuarios (personas, máquinas u otros computadores).

Los recursos básicos de un sistema informático, son el hardware, el software y los datos. El sistema operativo facilita los medios para el uso adecuado de estos recursos.

Un sistema operativo es un conjunto de programas que supervisan y controlan los programas de usuario para evitar errores y el uso inadecuado del computador. El sistema operativo controla así mismo las operaciones de E/S, podemos pues entender por sistema operativo a una colección de programas cuya principal misión es gestionar los recursos del computador, y en consecuencia

será el soporte lógico que controla el funcionamiento del equipo físico (*Hardware*). Los objetivos buscados por un sistema operativo son dos: eficacia en la gestión de los recursos, y facilitar la gestión a los usuarios (interfase usuario/máquina de fácil manejo, por ejemplo, entornos gráficos tipo Windows). Estos objetivos no son fácilmente compatibles, si se consiguen se logra un buen sistema operativo.

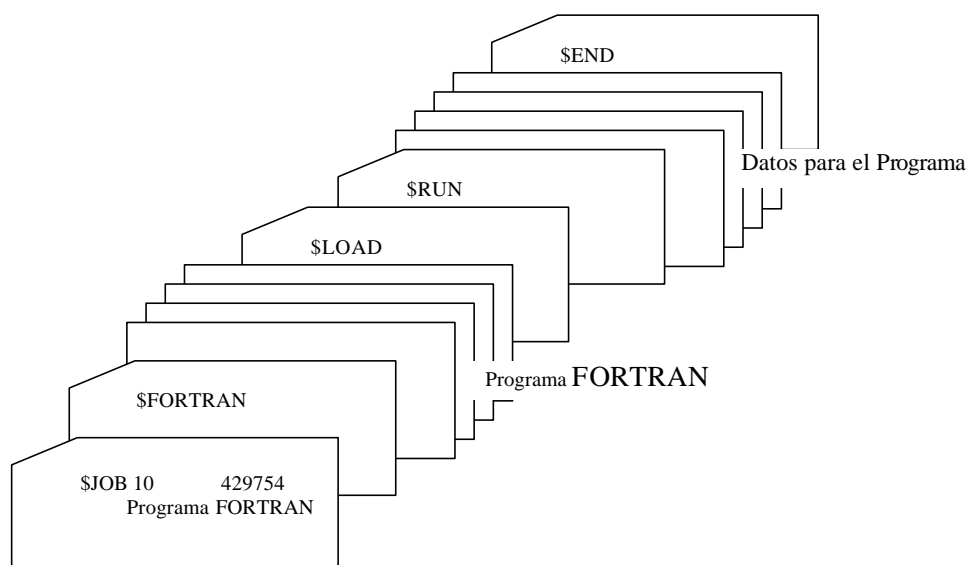
En los primeros computadores sólo era posible que la CPU atendiera a un programa cada vez (monoprogramación), que se introducía mediante tarjetas perforadas. Una de las soluciones fue el manejo de la máquina por un especialista en operación (operador). Su misión era la de controlar el sistema, cargar el programa que le pasaba el programador, y obtener resultados. Estas operaciones no utilizaban toda la capacidad de proceso de la CPU, pues dejaban tiempos de CPU sin ocupar, aún cuando se agrupaban los trabajos con una necesidad común de recursos (físicos y lógicos), y se ejecutaban como si fueran un bloque.

Analizando el trabajo que realizaba un operador, se observó que era bastante mecánico y que podía ser automatizado en gran parte, ganando tiempo, así pues, se desarrolló un pequeño programa que permanecía constantemente en memoria (residente), y que transfería automáticamente el control de un trabajo a otro. Este programa se denominó **programa Monitor** y, puede ser considerado como el precursor de los sistemas operativos actuales.

En el momento de encender la computadora se pasaba el control al monitor, que a su vez daba el control al primer trabajo; cuando terminaba su ejecución tomaba de nuevo el control el monitor, dando paso al segundo de los trabajos, y así sucesivamente. El programa monitor contenía las siguientes partes:

- El secuenciador automático de trabajos.
- El interprete de tarjetas de control.
- Controladores software de entrada/salida (*drivers*).

En la figura siguiente se puede ver un bloque de tarjetas que representan un trabajo, realizado en Fortran, y en la que puede apreciarse las tarjetas de control del compilador Fortran y las tarjetas con los datos.



6.2. Mejora de las prestaciones de las computadoras.

6.2.1. Operaciones "OFF-LINE".

Una vez resuelto el problema de la automatización en la ejecución de los trabajos, aun quedaba mucho tiempo en el que el procesador estaba desocupado. Esta situación se debía fundamentalmente a la diferencia de velocidad entre el procesador y los dispositivos de E/S, cuyo funcionamiento era principalmente mecánico. Para realizar la comunicación con lo periféricos se desarrollaron los *procesadores satélites*, especializados en estos procesos y que descargaban de trabajo al procesador central, optimizando el tiempo de CPU ya que la comunicación con los periféricos se dejaba al procesador satélite. (Operación *off-line*).

6.2.2. Memorias intermedias (*buffers*).

Otra solución a la lentitud de los dispositivos de E/S es el *buffering*, que es la utilización de memoria intermedia (o buffer). El buffering trata de mantener permanentemente ocupados tanto la CPU como los dispositivos de E/S. La idea es muy sencilla, los datos que se transmiten de memoria al periférico, y viceversa, son volcados en el buffer donde están disponibles para ser utilizados. Por ejemplo, para el envío de datos a un dispositivo de salida, la CPU vuelca los datos al buffer (memoria intermedia) hasta que el dispositivo de salida pueda aceptarlos. Cuando el dispositivo de salida vacíe el buffer, reclamará tiempo de CPU mediante una línea de interrupción habilitada para tal efecto. La CPU dejará la tarea que esté haciendo (aquí entra en juego la prioridad del dispositivo implicado en la tarea en ejecución y del dispositivo que genera la interrupción) guardando el estado del programa que se estaba ejecutando en la *pila*, para continuar donde estaba, una vez atendida la interrupción, que en este caso consistirá en volver a llenar el buffer e instruir al dispositivo para que inicie la lectura de los datos. Por tanto, debe existir una comunicación entre CPU y periférico, y una gestión del buffer.

Existen dos tipos de dispositivos, los que transfieren información carácter a carácter y los que lo hacen por bloques de caracteres. Esto depende de lo rápido que sea el dispositivo periférico, así por ejemplo la pantalla y el teclado serían del primer grupo (de carácter en carácter), mientras que los discos, tanto fijos (o duros) como los extraíbles (flexibles) serían del segundo grupo. También hay dispositivos de E/S que realizan acceso directo a memoria, sin necesidad de ocupar tiempo de CPU. Son los dispositivos con *DMA* (Acceso Directo a Memoria), su velocidad de transferencia de datos es próxima a la de la memoria, y no necesitan la CPU para realizar la transferencia de información a la memoria.

Cuando un dispositivo con DMA interrumpe a la CPU, la rutina de servicio de interrupciones comprueba en primer lugar los errores en la transferencia previa, luego selecciona, para la transferencia, un buffer de una cola de buffers (vacío para entradas, y lleno para salidas).

6.2.3. *Spooling*.

La preparación de los trabajos off-line fue reemplazada rápidamente en la mayoría de los sistemas por la técnica denominada *spooling*. Esto fue debido a que los sistemas informáticos con disco se difundieron rápidamente y se resolvió el problema de los sistemas de cintas, donde el acceso a la información era exclusivamente secuencial y, para poder leer la cinta, había que escribirla entera antes de rebobinarla.

En un sistema de disco, las tarjetas perforadas se leen directamente desde la lectora de tarjetas sobre el disco. La posición de las imágenes de las tarjetas se registra en una tabla mantenida por el sistema operativo. En la tabla se anota cada trabajo una vez leído. Cuando se ejecuta un trabajo sus peticiones de entrada desde la lectora de tarjetas se satisfacen leyendo en el disco. Análogamente cuando el trabajo solicita la salida de una línea por la impresora, esta línea se copia en el buffer del sistema y se escribe en el disco. Cuando la tarea se ha completado, la salida se imprime realmente buscando los datos en el disco.

Esta forma de procesamiento se denomina *spooling* y consiste en utilizar el disco fijo como un buffer muy grande para leer por delante, tanto como sea posible, de los dispositivos de entrada y, almacenar los ficheros de salida hasta que los dispositivos de salida sean capaces de aceptarlos. Las diferencias entre buffering y spooling son que el buffering solapa la E/S de un trabajo con su propio tratamiento (procesamiento), y el spooling solapa la E/S de un trabajo con el procesamiento de otro.

6.3. Multiprogramación.

La multiprogramación es un modo de trabajo distinto en el que se pueden ejecutar varios programas simultáneamente con el fin de aprovechar al máximo los recursos de la computadora.

Los trabajos realizados en una computadora, desde el punto de vista de ocupación de tiempo de procesador y de los dispositivos periféricos, se pueden agrupar en dos bloques:

- Trabajos limitados por procesos: aquellos que consumen la mayor parte de su tiempo en el tratamiento de la información y muy poco en operaciones de entrada/salida.
- Trabajos limitados por operaciones de E/S: aquellos que dedican la mayor parte del tiempo a operaciones de E/S, haciendo poco uso del procesador, que se mantiene inactivo durante grandes períodos de tiempo.

Obviamente, un proceso particular puede estar decantado hacia un lado u otro, o bien situado en una situación intermedia.

El segundo tipo de procesos dio lugar a una nueva técnica denominada multiprogramación. Esta técnica consiste en aprovechar la inactividad del procesador durante la ejecución de una operación de E/S de un proceso, para atender a otro proceso. Desde el punto de vista del usuario, los procesos que se están ejecutando en la computadora están corriendo en paralelo, en realidad lo que sucede es que se están utilizando pequeños tiempo de CPU (ráfagas) para cada uno de los procesos que se ejecutan. En estas condiciones nos encontraremos con varios problemas:

- El acceso al procesador deberá seguir algún tipo de reglas, o políticas de trabajo, que permitan la realización de las distintas tareas que implica la ejecución de los programas.
- Se hace necesario algún tipo de administración de la memoria, ya que tiene que ser compartida por los distintos trabajos que se están ejecutando.
- Varios trabajos pueden necesitar la utilización de un mismo recurso al mismo tiempo, dando lugar a un problema de concurrencia conocido como “*abrazo mortal*”.

6.3.1. Tiempo compartido.

En este modo de trabajo la organización no se realiza por trabajos, sino por sesiones. Una sesión es todo el conjunto de trabajos que se realizan desde que un usuario se conecta a la computadora hasta que se despide de la misma. Durante estas sesiones se pueden realizar multitud de operaciones controladas por un proceso, denominado *interprete de órdenes*, que mantiene el diálogo entre el usuario y el sistema operativo.

Durante su sesión el usuario cree tener todos los recursos de la computadora, aunque existan otros usuarios con sus sesiones activadas simultáneamente. Los sistemas de tiempo compartido se caracterizan por:

- Ser interactivos.
- Atender a varios usuarios simultáneamente.
- Ofrecer unos tiempos de respuesta relativamente cortos.
- Poseer una fuerte gestión de archivos.
- Utilizar técnicas de buffering y spooling.
- Gestionar memoria virtual.

6.3.2. Tiempo real.

El tiempo real es otra posibilidad de los sistemas operativos multiprogramados en los que se necesita un tiempo de respuesta pequeño ante cualquier petición.

Suele emplearse en aplicaciones dedicadas a sistemas de control, que utilizan sensores como elementos de entrada y donde es necesaria una respuesta rápida sobre el sistema a controlar. Podemos decir que un sistema trabaja en tiempo real si el tiempo de respuesta permite controlar y regular al medio sobre el que opera.

6.4. Multiproceso.

Uno de los problemas actuales ha sido la aparición de aplicaciones que manejan gran cantidad de datos, que un solo procesador no es capaz de procesar en el tiempo requerido. Podemos citar el caso de sistemas expertos, donde el volumen de datos es extremadamente grande y se necesitan unos tiempos de respuesta muy pequeños.

El multiproceso se basa en la posibilidad de descomponer el algoritmo que se está ejecutando en varios subalgoritmos, de tal forma que estos puedan ser tratados por varios procesadores en paralelo. Al finalizar se deben unir los resultados parciales y obtener un resultado final en la computadora. Evidentemente el tiempo de ejecución es mucho menor que si solo dispusiéramos de un procesador.

6.5. Módulos de un sistema operativo estándar.

Las funciones de un sistema operativo pueden clasificarse de forma general en:

- a. Asignación de tiempos de CPU
- b. Control de recursos
- c. Control de entrada/salida
- d. Control de errores y protección
- e. Interfase con el operador y facilidades contables.

La asignación de tiempos de CPU, consiste en planificar las distintas actividades que tienen o van a tener lugar en el computador, y a las cuales habrá que dedicar tiempo de CPU.

El control de recursos consiste en asignar los recursos del computador de una forma racional. Entre las tareas más importantes a la hora de asignar recursos se encuentra el reparto de la memoria del computador entre los programas, la distribución del espacio en las unidades de almacenamiento masivo, tanto para los programas como para los datos y, el control de las colas de los dispositivos de salida.

En el control de entrada/salida se gestionan los canales de datos hacia y desde los periféricos.

El control de errores, y la protección frente a estos, supone informar de cualquier condición anómala, a la vez que se intenta minimizar su efecto sobre el resto del sistema informático.

La interfase con el operador proporciona a la persona que utiliza el computador, facilidad para el uso del mismo.

Las facilidades contables permiten facturar a los usuarios el importe que supone su uso de los recursos del computador, de acuerdo a una tabla de costes previamente establecida.

6.5.1. El núcleo.

La estructura del sistema operativo estándar, se presenta en módulos. El primer módulo es el núcleo que da servicio al resto. Por tanto, el núcleo es el módulo de nivel más bajo del sistema operativo y descansa directamente sobre el hardware del computador, proporcionando una serie de servicios a las capas superiores del sistema operativo. Entre las tareas que desempeña el núcleo está el manejo de interrupciones, la asignación de trabajo al procesador, y proporciona una vía de comunicación entre los distintos programas.

Cuando el hardware del computador detecta una interrupción, el control se transfiere al módulo de control de interrupciones del núcleo, que analiza el carácter de la interrupción y toma las acciones apropiadas. Estas acciones suelen consistir en transferir el control a otro módulo del sistema operativo, iniciar otro programa o continuar la ejecución del programa interrumpido. Dado que muchas interrupciones tienen lugar por motivos de entrada/salida, el módulo de control de E/S es uno a los que se transfiere el control con más frecuencia.

Para asignar trabajo al procesador, el núcleo transfiere el control al programa que el planificador ha determinado que sea el próximo en ejecutarse. La comunicación entre programas se logra manteniendo una cola de mensajes en espera para cada uno de los programas activos. El núcleo

recibe los mensajes y los va almacenado en la cola apropiada, para distribuirlos cuando el programa destino se active.

La mayoría de los computadores disponen de instrucciones en lenguaje máquina cuyo uso esta restringido al núcleo del sistema operativo. Entre ellas se encuentran instrucciones que transfieren el control de un programa a otro, así como instrucciones que acceden a determinados registros. Restringir estas instrucciones es un modo apropiado de controlar el funcionamiento global del computador y limitar los efectos de los errores.

6.5.2. Gestión de memoria.

La memoria principal de muchos de los computadores es mucho más pequeña de lo que sería necesario para manejar todos los programas y datos que maneja un computador en un momento dado. El modulo de gestión de memoria de un sistema operativo, es el encargado de asignar ciertas porciones de la memoria principal a los diferentes programas o partes de los programas que la pueden necesitar, mientras el resto de los datos y los programas se mantienen en los dispositivos de almacenamiento masivo. De este modo cuando se asigna una parte de la memoria principal se hace de una forma estructurada, siguiendo un determinado orden.

6.5.3. Control de la E/S.

Los problemas asociados con la entrada y la salida de datos tienen su origen en el hecho de que los distintos dispositivos de entrada/salida tiene distintas características y diferentes velocidades. Por ejemplo una impresora de líneas acepta una línea cada vez y un teclado trabaja carácter a carácter, lo que significa que una impresora de líneas tiene una velocidad mas de 100 veces superior al teclado.

El modulo de control de entrada/salida de un sistema operativo trata estos problemas presentando al programador la E/S como una cuestión independiente del dispositivo. El sistema operativo se encarga de los aspectos físicos de la transferencia (por ejemplo determinar que sector, pista y cara del disco se van a utilizar), dejando al programador libre para que solo se preocupe de los aspectos lógicos de la transferencia (los registros y archivos involucrados), siendo pues independiente del dispositivo en concreto que se utiliza. Una técnica muy común especialmente en la salida es la ya explicada del *spooling*.

6.5.4. Gestión de dispositivos de almacenamiento masivo.

Los dispositivos de almacenamiento masivo de un computador constituyen el sistema físico donde se depositan de una forma permanente los datos y los programas de los usuarios. El modulo encargado de la gestión de estos dispositivos tiene la misión de mantener la estructura de esta información y de asegurar el uso eficiente de estos medios.

Los datos y programas de un dispositivo de almacenamiento masivo se mantienen en ficheros. El modulo de gestión supervisa la creación, actualización y eliminación de los archivos. Este modulo mantiene un directorio con todos los archivos que existen en cada momento en el

dispositivo de almacenamiento masivo y coopera con el módulo de gestión de memoria durante las transferencias de datos desde y hacia la memoria principal. Los archivos almacenados en los sistemas de almacenamiento masivo tienen distintos propósitos. Algunos contienen información que puede ser compartida, otros son de carácter privado y incluso secreto. Por tanto, cada archivo está dotado de un conjunto de privilegios de acceso, que indican la extensión con la que se puede compartir la información contenida en el archivo. El sistema operativo comprueba que estos privilegios no son violados.

6.5.5. Asignación de recursos y planificación.

La mayor parte del tiempo en que un computador está funcionando, la demanda de recursos es mayor que los que realmente existen. Para resolver este problema los sistemas operativos disponen de una política de gestión de recursos, el módulo de asignación de recursos es el responsable de llevar a la práctica esta política.

La solución sería sencilla si fuera posible utilizar una política directa del tipo "se atenderá primero al que antes lo solicite", esta política puede llevar a situaciones denominadas de *abrazo mortal*, en la que dos programas quedan bloqueados al esperar insistentemente a otro recurso que es justamente el que bloquea el otro programa, de tal forma que ninguno suelta su recurso, si no que piden utilizar el recurso que les falta para completar la tarea siendo este el recurso bloqueado por el otro programa.

El planificador se ocupa fundamentalmente de asignar tiempo del procesador a los programas de acuerdo a una cierta política de planificación, que varía notablemente de un sistema operativo a otro. Normalmente se hace uso de una cierta jerarquía de prioridades, que determina la colocación en la cola de espera de tiempos de CPU, y en consecuencia, como se asignará el tiempo de CPU a cada programa. Una política de planificación muy común es la de asignar una fracción de tiempo de CPU a cada programa de tal forma que si en esa fracción de tiempo el programa no ha finalizado vuelve a la cola de programas en espera de tiempo de CPU. Las políticas de planificación no deben de ser muy complicadas, ya que si no el procesador empleará mucho tiempo en decidir que es lo que se debe de hacer a continuación.

6.5.6. Protección.

Cuando en un computador existen diferentes programas corriendo en el mismo simultáneamente, es necesario protegerlos entre sí. En consecuencia el primer programa a proteger será el propio sistema operativo.

La protección tiene lugar mayoritariamente contra dos procesos, errores y abusos deliberados de los recursos del sistema. Aunque es imposible para el sistema operativo prever los errores de los programas de aplicación, es esencial detectarlos y diagnosticarlos lo antes posible para limitar sus efectos. Aunque los mecanismos de protección están diseñados para evitar que tengan lugar acciones no autorizadas, pocos sistemas operativos pueden considerarse seguros.

Especial interés tiene la seguridad de la memoria principal y de los medios de almacenamiento masivo. Los archivos almacenados en los medios de almacenamiento masivo se protegen mediante los medios de privilegios de acceso mencionados anteriormente.

La protección de la memoria principal del computador se considera el aspecto más importante de la seguridad de un computador, ya que todo lo que hace un computador lo hace con intervención de su memoria principal. El modulo de gestión de memoria asigna a cada tarea una parte de la memoria principal del computador y a continuación asigna a cada una de estas porciones un grado de protección según la naturaleza de las tareas que tiene encomendadas. Se realizan comprobaciones reiteradas para asegurar que no se producen violaciones de la memoria.

Algunos sistemas operativos asignan diversos niveles de privilegio a los programas que se están ejecutando en el sistema. El núcleo del sistema operativo tiene el grado mas alto, mientras que los programas de aplicación se encuentran en el nivel mas bajo.

6.5.7. Interfase Usuario – Sistema operativo.

Es necesario facilitar el uso de la computadora e independizar los recursos del sistema, en cuanto a sus características de funcionamiento físico de su utilización. Se busca pues una interfase sistema-usuario de fácil uso y practica en cuanto prestaciones.

En un sistema multiacceso la comunicación entre el usuario y el sistema operativo es interactiva. El usuario puede ir dando ordenes al sistema operativo desde un terminal y recibir en el mismo terminal los mensajes con los que responde el sistema.

La interfase entre el sistema y la persona responsable del sistema esta constituida por ordenes y mensajes. El operador puede dirigir gran parte de la funciones del sistema operativo. Esto se aplica particularmente a la planificación y asignación de recursos. En todo momento el operador tiene el control global del computador.

6.6. Memoria virtual.

El espacio de direccionamiento lógico de un proceso tiene que encontrarse en memoria física antes de que el proceso pueda ser ejecutado. Esto es, la parte del programa que se este ejecutando debe de estar cargada en memoria física, sin embargo es posible cargar por partes el programa, trayéndolo por partes desde un sistema de almacenamiento físico.

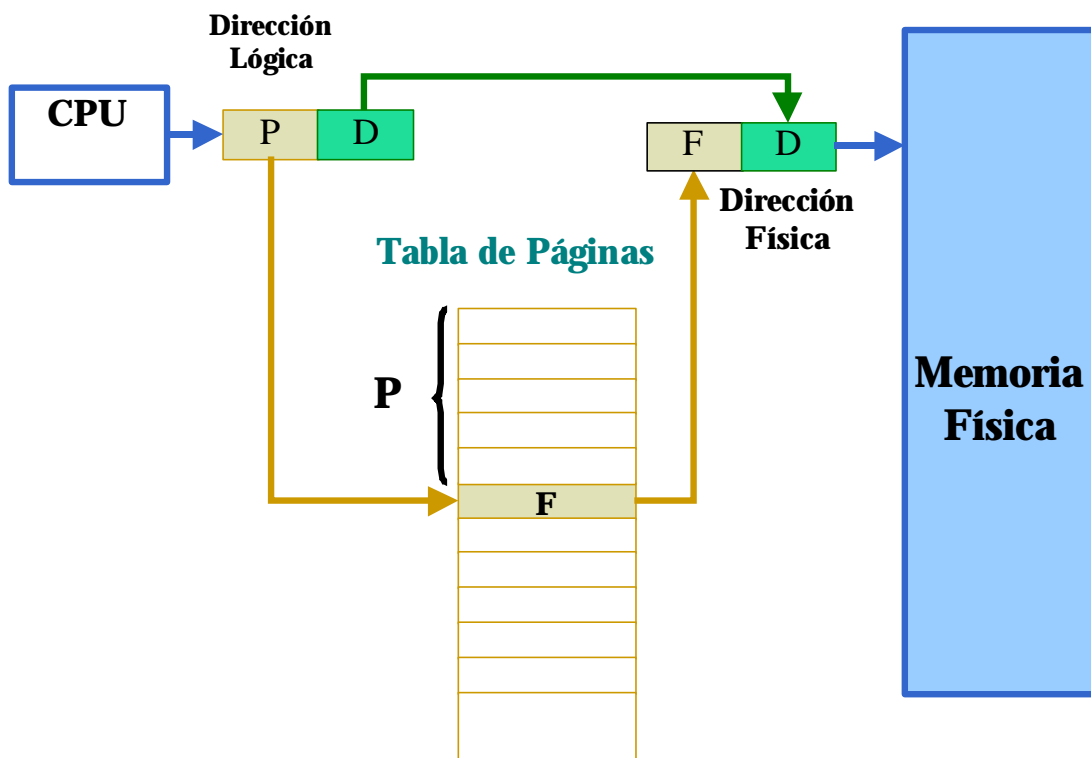
La memoria virtual es una técnica de gestión de memoria, que permite la ejecución de programas que no pueden hallarse en su totalidad en la memoria principal ya que ocupan mas memoria de la disponible, su ventaja más evidente es que los programas de usuario pueden ser mayores que la memoria física. En la memoria virtual se combinan hardware y software, permitiendo la ejecución de programas cargados parcialmente en memoria real. Con ello se tienen las siguientes ventajas:

- Si los programas se pueden ejecutar por partes, la memoria lógica puede ser mayor que la real disponible.
- Puesto que cada programa ocupa menos memoria real, se puede elevar el índice de multiprogramación, y por tanto la eficiencia del sistema.

6.6.1. Paginación.

La memoria asignada a un trabajo determinado debe de ser contigua o lograr algún método que admita que la memoria de un programa sea no contigua. La paginación es un método de gestión de memoria que admite que la memoria que utiliza un programa sea no contigua. Toda dirección generada por la CPU se divide en dos partes: un número de página (p) y un desplazamiento (d). El número de página se utiliza a modo de índice en una tabla de páginas. La tabla de páginas contiene la dirección base de cada página en la memoria física. Esta dirección base se combina con el desplazamiento para definir la dirección de la memoria física que se envía a la unidad de memoria.

El modelo de la paginación de memoria se muestra en la figura siguiente. La memoria física se fragmenta en bloques de tamaño fijo llamados celdas de página. La memoria lógica también se fragmenta en bloques del mismo tamaño llamados páginas. Cuando hay que ejecutar un programa se cargan sus páginas en cualesquiera de las celdas que haya disponibles, y se define la tabla de páginas para traducir las páginas de usuario en celdas de memoria. Una forma de llevar a cabo la memoria virtual es mediante el modelo de la paginación, otra forma sería mediante el modelo de la segmentación, que veremos a continuación.



6.6.2. Segmentación.

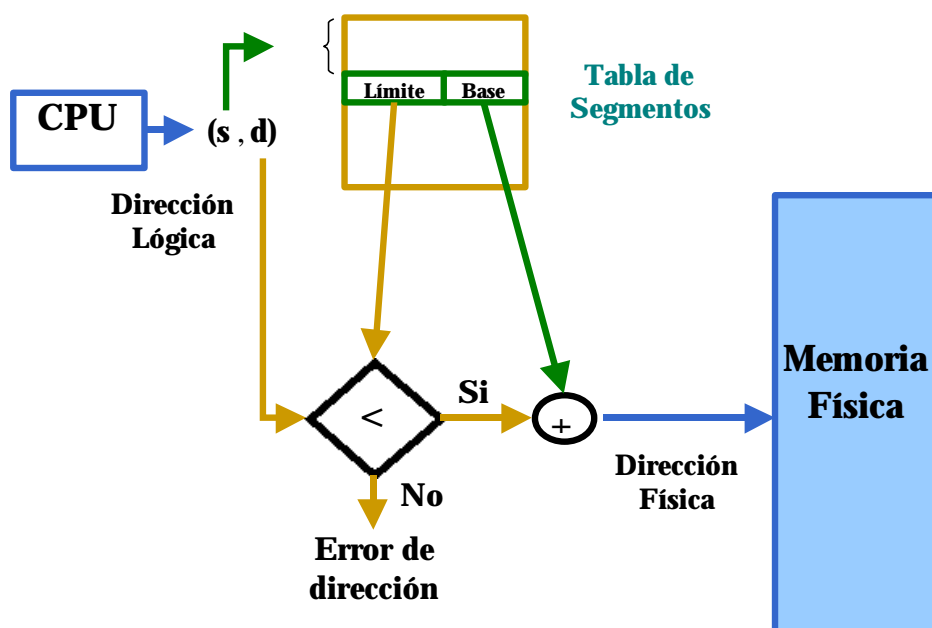
Un aspecto importante de la gestión de la memoria que la paginación convierte en inevitable es la separación de la visión que el usuario tiene de la memoria y la memoria física real. La visión del usuario no coincide con la memoria física real. La visión del usuario se transforma en la memoria física. La traducción de direcciones permite esta diferencia entre la memoria lógica y la memoria física. El usuario y el programador ven la memoria como una colección de segmentos de longitud variable no necesariamente ordenados, y no como una tabla lineal de palabras.

La segmentación es un esquema de gestión de memoria que soporta la visión que el usuario tiene de la misma. Un espacio de direcciones lógicas es una colección de segmentos. Cada segmento tiene un nombre y una longitud. Las direcciones especifican tanto el nombre del segmento como el desplazamiento dentro del segmento. Por la tanto el usuario especifica cada dirección mediante dos cantidades: un nombre de segmento y un desplazamiento. En comparación en la paginación se especificaba una única dirección, que el hardware particionaba en número de páginas y desplazamiento, siendo todo ello invisible para el programador.

Un programa consta de varias partes: una parte principal y unas subrutinas, procedimientos o módulos. También puede haber diversas estructuras de datos: tablas, matrices, pilas variables, etc. Cada uno de estos módulos o elementos de datos se referencian por un nombre. Estos elementos (*segmentos*) son de *longitud variable*. Los elementos dentro de un segmento están identificados por su desplazamiento desde el principio del segmento.

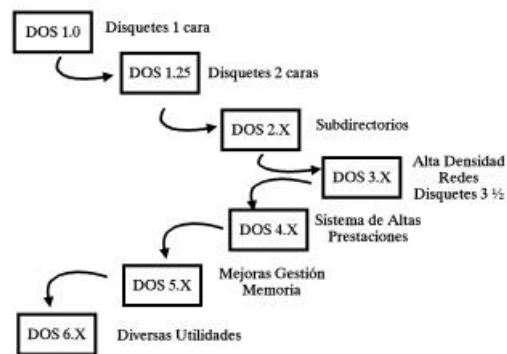
Aunque el usuario ahora puede referenciar los objetos del programa por una dirección de dos dimensiones, la memoria física real es por supuesto una secuencia unidimensional de palabras. La transformación se efectúa mediante la tabla de segmentos.

Una dirección lógica consta de dos partes: un número de segmento (*s*) y un desplazamiento (*d*). El número de segmento se utiliza como un índice en la tabla de segmentos. Cada entrada en la tabla de segmentos tiene una base del segmento y un límite. El desplazamiento (*d*) de la dirección lógica tiene que estar comprendido entre el cero y el límite del segmento.



6.7. DOS (Disk Operating System).

El MS-DOS fue desarrollado en 1979 por Tim Paterson, que trabajaba en Seattle Computer Products. El producto fue un encargo de Microsoft Corporation, que a su vez trabajaba para ofrecérselo a IBM como sistema operativo para un nuevo computador personal (PC) que irrumpiría en el mercado a mediados de 1981 (concretamente apareció en agosto). Este desarrollo ocurrió en condiciones un tanto especiales, ya que por aquel entonces Digital Research, propietaria del Sistema Operativo más ampliamente distribuido en los ordenadores de la época (CPM), no alcanzó un acuerdo satisfactorio con IBM. El resultado fue la aparición de un sistema operativo en dos versiones diferentes: PC-DOS, propiedad de IBM y para computadores de su marca; y MS-DOS, propiedad de Microsoft, para computadores *clónicos*. Se trata de un sistema operativo monousuario y monotarea. En nuestros días ha cedido el testigo a sistemas con una presentación menos austera y más intuitiva.



Evolución del Sistema Operativo DOS.

6.7.1. La estructura de un disco bajo DOS.

La estructura lógica de un disco que DOS pueda gestionar se consigue mediante un proceso que se realiza diferenciando disquetes y discos duros. Formatear un disco duro o un disquete implica dos pasos distintos: (1) formateo de bajo nivel (físico), (2) formateo de alto nivel (formateo lógico del DOS). La orden `FORMAT` del DOS ejecuta ambos procesos en los disquetes. En los discos duros `FORMAT` realiza solo el formateo lógico, el físico lo realiza el fabricante o utilidades especialmente diseñadas para ello.

Durante el proceso de formateo de bajo nivel todos los discos DOS se dividen en unidades direccionables de 512 Bytes denominadas sectores (hay otros programas que permiten utilizar otros tamaños). Cuando los controladores de discos acceden a ellos lo hacen manejando sectores. Los sectores están dispuestos uno tras otro en círculos de forma concéntrica denominadas pistas. Las pistas que comparten el mismo radio y eje (hay una por cara de disco) se denominan cilindros.

Los sectores de un disco son las unidades fundamentales de la actividad del disco, y a la mínima parte a la que accede la controladora del disco es el sector, la controladora del disco duro realiza las operaciones de entrada/salida por bloques o paquetes de datos de longitud fija, lo cual permite una gestión más rápida en el intercambio de la información, estos varios sectores a los cuales se accede de una vez, se denominan cluster. Es conveniente conocer como el DOS contempla el disco, ya que lo aplanar, prescindiendo de sus dimensiones, que son pista, cara, y sector que serán necesarias para acceder físicamente al disco. DOS trata los sectores de un disco como una lista secuencial de sectores del primero al ultimo, de ahí el limite de los 32 Mbytes cuando se guardaba esta información en un entero positivo de 16 bits (2^{16} . 12 bytes). Sólo en el último momento, cuando la información se lee o se escribe realmente en el disco, el DOS traduce su notación interna (números secuenciales) a la notación de tres dimensiones propia del disco

(propiamente esta operación de transformación lógica a física, o mapeo, la realiza el BIOS). El precio a pagar es que, el DOS al no conocer que sectores están en el mismo cilindro, hace el acceso al disco más lento al desplazar cabezales innecesariamente.

El formateo a alto nivel del disquete o partición del disco duro establece una estructura lógica distribuyendo los sectores entre el área de sistema (en la parte externa del disco) y el área de datos (el resto del disco). El área de sistema contiene el sector de arranque, la tabla de asignación de archivos (FAT), y el directorio raíz. La idea de que un disco duro pueda contener varias particiones, se concibió originalmente para que pudieran almacenarse varios sistemas operativos en la mismo computador, la elección de la partición activa permite elegir cual de ellas se elige para arrancar el computador, normalmente el "primer arranque" se establece de la unidad de disquete flexible A, que es un disco totalmente DOS y no admite particiones.

El sector de arranque esta localizado en el primer sector de la cara 0 pista 0 del disco, este sector contiene un pequeño programa de arranque conocido como el programa de arranque maestro y a continuación la tabla de particiones, existe también en este sector un área de datos donde se encuentra información tal como: bytes por sector, sectores por pista, numero de cabezas, sectores por cluster, numero total de sectores. En la tabla de particiones hay espacio para 4 registros de 16 bits (4 particiones), la información que guardan incluye: sector inicial y número de sectores, tipo de partición (DOS, XENIX, DOS extendida, etc.). Con la partición extendida de DOS se encuentra un puntero a una tabla situada en el interior del disco, que da soporte a varias unidades lógicas. El BIOS carga y ejecuta el programa de arranque maestro, consultando en la tabla de particiones para conocer la situación de la partición activa. Cuando llega a la partición activa, carga el sector de arranque de esa partición en memoria y le entrega el control.

La tabla de asignación de archivos, esta situada inmediatamente después del sector de arranque. A diferencia del sector de arranque la FAT utiliza varios sectores. En realidad se mantienen dos copias de la FAT en el disco aunque DOS no utiliza para nada la segunda (algunas utilidades de recuperación de archivos si la utilizan). La FAT es esencialmente una tabla de datos con registros de longitud fija (12 o 16 bit) asociados con cada cluster del disco. El numero registrado en cada entrada de un cluster indica si éste está marcado como defectuoso (durante el formateo), utilizado por un archivo o libre para almacenar nuevos datos. Un cero significa que el cluster está libre. Cualquier otro número indica que está ocupado, este número se utiliza para enlazar entre sí los diferentes cluster de tal forma que el contenido de un cluster apunta al cluster siguiente donde continua el archivo, marcándose el final del archivo con un número especial, el cluster de inicio esta situado en el directorio raíz que enseguida veremos.

Cuando un programa hace una petición para aumentar el tamaño de un archivo de forma que se hace necesario un cluster más, todo lo que DOS hace es buscar un cluster libre, copiar los datos nuevos allí (utiliza para ello los buffers que se reservan en CONFIG.SYS, empaquetando la información en ellos como paso intermedio) y cambia la entrada de la FAT para el que fue el último cluster que pasa ahora a apuntar al nuevo cluster utilizado, y cambia la entrada del nuevo cluster para indicar que es ahora el final del archivo. Sí se amplía un archivo y los cluster disponibles están a lo largo del disco, los cluster se asignan con esta disposición, dejando un archivo fragmentado. DOS puede gestionarlo sin problemas, pero ahora el cabezal del disco debe de moverse de un lado para otro para leer el archivo enlenteciendo el acceso al disco.

El directorio raíz, situado a continuación de la FAT y anterior al área de datos, contiene un numero fijo de entradas en función del disco o disquete que se trate. Existe una entrada por

archivo de 32 bits donde se registra el nombre del archivo, la extensión, el atributo, hora y fecha de creación o de última modificación, cluster de inicio y tamaño en bytes. El primer carácter de la parte reservada para el nombre puede ser un 0 que indica que la entrada no ha sido utilizada nunca, un punto que indica que la entrada está reservada por el DOS, una letra sigma minúscula lo que indica que dicha entrada contiene un archivo que ha sido borrado y una letra o carácter de los permitidos para nombres de archivo (que es el primer carácter del nombre del archivo). Los subdirectorios son archivos con atributo de subdirectorio, así no tienen límite en cantidad salvo el espacio de la zona de datos y el número de entradas que soporta el directorio raíz. Sin embargo tienen dos entradas al inicio, para el propio directorio y para el directorio anterior. Por último la siguiente zona de la partición del disco o del disquete es la zona de datos, donde se instalan los programas y las aplicaciones del usuario.

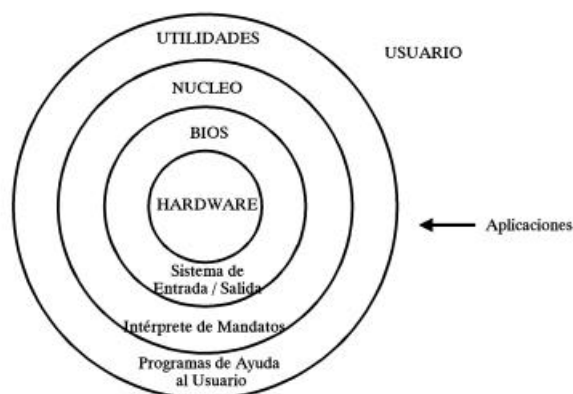
6.7.2. Núcleo, BIOS e interprete de ordenes.

Se entiende por núcleo o *kernel* de DOS a un conjunto de rutinas que se encuentran en los archivos IO.SYS (IBMIO.COM), MSDOS.SYS (IBMDOS.COM) y al interprete de comandos. Los dos archivos del sistema se encuentran en las primeras posiciones de la zona de disco denominada área de datos, junto con CONFIG.SYS y AUTOEXEC.BAT y tienen que ver con la inicialización del sistema.

El conjunto de rutinas básicas que comunican el computador y los dispositivos se denomina BIOS (Basic Input/Output System). Estas rutinas efectúan operaciones fundamentales, tales como leer y escribir en el disco o visualizar los datos en la pantalla. La BIOS reside fundamentalmente en memoria ROM (Read Only Memory) bajo la forma de dos chips que se insertan en la tarjeta madre del computador, también

forman parte las rutinas de inicialización del sistema que están en el archivo IO.SYS, y si se trata de una BIOS de IBM hay ROM-BASIC. La BIOS principal es la que se carga en los últimos 64 Kb del espacio de direcciones de memoria y es conocida como la BIOS planar (en referencia a los manuales técnicos de IBM). Las tarjetas gráficas EGA, MCGA y VGA tienen su propia BIOS si como el controlador del disco duro o las tarjetas de red local. A grandes rasgos es preciso distinguir entre 3 BIOS. La BIOS del PC, BIOS de AT y BIOS de PS/2 o superiores. La BIOS del AT amplía la BIOS del PC en varias rutinas para gestionar el reloj en tiempo real, la memoria extendida, los teclados ampliados. La BIOS de los PS/2 supone una fuerte mejora sobre su predecesora con la agrupación de dos conjuntos de servicios CBIOS y ABIOS, la CBIOS compatible con la del AT mejora algunas cosas como servicios para controladoras de discos ESDI y otros, la ABIOS (advanced BIOS) tiene funciones para gestionar el modo protegido. La BIOS reside en memoria ROM, sin embargo la BIOS también utiliza información almacenada en RAM, estas tres zonas son: el área de la CMOS-RAM, la tabla de vectores de interrupción y el área de comunicaciones de la BIOS, estas dos últimas son direcciones de memoria asignadas a RAM convencional (memoria base).

La memoria CMOS-RAM consta de una memoria no volátil (tecnología CMOS y pilas de alimentación, para cuando se desconecta el computador) que se almacenan en un chip. Aquí se



Estructura del Sistema Operativo DOS.

almacenan los tipos de unidades de disco instalados, tamaño de memoria RAM, y forma de gestión, valor del reloj en tiempo real, etc.

A la tabla de vectores de interrupción se le asigna el primer Kbyte del espacio de direcciones de memoria, cuando un dispositivo genera una interrupción (cada interrupción es identificada por un número), el microprocesador debe de guardar en la pila el trabajo que estaba realizando, de tal manera que después pueda continuar donde estaba, realmente guarda el proceso: programa en ejecución junto con el entorno asociado (registros, variables). El procesador pasa el control al gestor de interrupciones que mediante la tabla de vectores de interrupción localiza la dirección de la rutina de interrupciones adecuada para tratar la interrupción generada por ese dispositivo en concreto. Por ultimo hay comentar que las interrupciones están jerarquizadas según prioridades, por ejemplo al controlador del disco duro le corresponde la prioridad 9, mientras que si se llena el buffer del teclado activa una interrupción con prioridad 1, así mismo hay fundamentalmente tres tipos de interrupciones, la interrupción hardware, interrupción de excepción (el procesador se encuentra ante alguna instrucción sin sentido), y interrupción software. Podemos pues considerar a las interrupciones como la fuerza motriz de los computadores que tengan instalado el DOS.

El área de comunicaciones de la BIOS viene a continuación en el mapa de direcciones de memoria, se instala en RAM durante la inicialización del sistema (POST: Power On Self Test, del que hablaremos posteriormente) se rellenan varios bytes a continuación de la tabla de vectores, que serán utilizadas como área de datos en los servicios de la BIOS. El área de comunicaciones extendida (utilizada por la CBIOS, por ejemplo para los nuevos servicios del ratón que incorpora el PS/2) es el Kbyte más alto de la memoria convencional (de 639 a 640 Kb). Por ultimo en esta zona especial nos encontramos con el área de trabajo del DOS y del BASIC, que ocupa 256 bytes.

Un controlador de dispositivos es un programa que permite la comunicación entre el DOS y un dispositivo hardware. Los controladores de dispositivos residentes son extensiones a la BIOS localizados en uno de los dos archivos (ocultos) del sistema, el IO.SYS (IBMIOS.COM en el PC-DOS de IBM), que se cargan en memoria cuando se inicializa el sistema. Estos controladores de dispositivos residentes en memoria forman lo que se conoce como la BIOS del DOS, si pues el DOS proporciona automáticamente controladores de dispositivos (residentes) para todos los dispositivos del DOS (CON, COM1-COM4, LPT1-LPT3, CLOCK\$ y NUL).

Mediante un controlador de dispositivos instalable es posible instalar en el computador periféricos no soportados por la BIOS planar, tal como un ratón, un escáner o un gestor de memoria extendida, los controladores instalables son archivos que suelen presentarse con la extensión SYS, y se invocan desde el archivo CONFIG.SYS mediante el orden DEVICE, algunos presentan la extensión EXE y se pueden ejecutar desde el intérprete de mandatos (COMMAND.COM).

Se distingue entre controladores de dispositivos de caracteres que efectúan la transferencia de datos carácter a carácter y controladores de dispositivos de bloques. Los dispositivos de caracteres efectúan las operaciones de entrada/salida de datos en modo serie, un byte cada vez. Estos dispositivos se identifican con un nombre de 8 caracteres, que sirve para que los programas de aplicación puedan abrir el dispositivo para entrada y salida como si se tratase de un archivo. Los dispositivos de bloques realizan las operaciones de entrada/salida por bloques o paquetes de datos con una longitud fija. Todas las unidades de disco son dispositivos de bloques, estos dispositivos no tienen nombre propio sino que utilizan las letras del alfabeto (A,B,C...) para ser

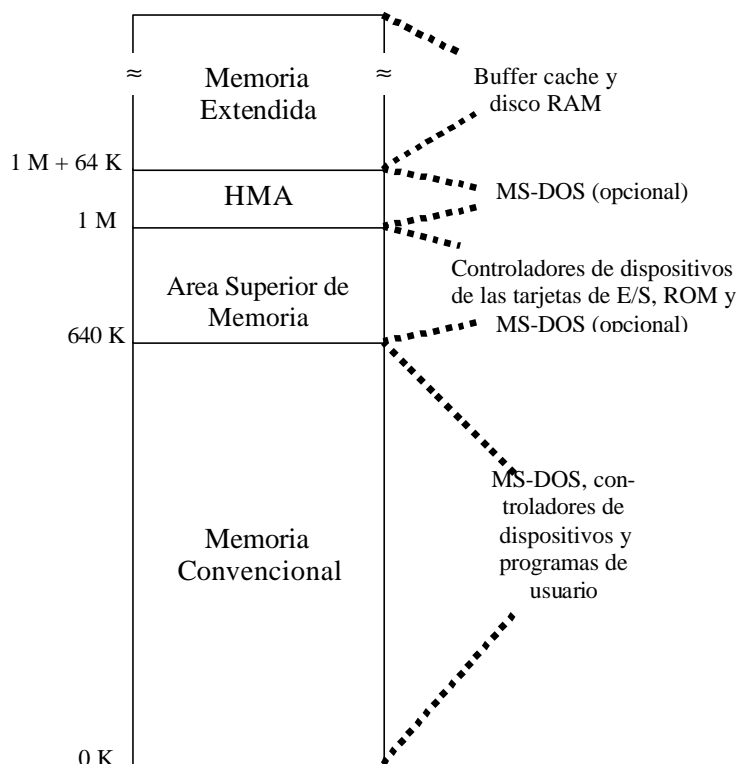
reconocidas. Un controlador de dispositivo orientado al bloque, puede soportar varias unidades físicas, no sucede lo mismo con los orientados al carácter.

El intérprete de ordenes o mandatos contiene todas las ordenes internas del DOS y es el responsable de que aparezca en pantalla el prompt del DOS, también llamado inductor del DOS. Al ejecutar el intérprete de mandatos las ordenes internas quedan residentes en memoria. Es posible iniciar un nuevo procesador de órdenes que no sea el citado, con lo cual el DOS crea un nuevo ambiente para órdenes. La orden EXIT permite salir del nuevo ambiente de órdenes y volver al anterior.

6.7.3. Gestión de memoria.

Cuando IBM diseñó su primer microcomputador, el PC, utilizó un bus de direcciones de 20 bits, se trataba del microprocesador de INTEL 8086. Lo que permitía direccionar $2^{20} = 1.048.576$ posiciones de memoria. En la distribución de la memoria era conveniente asignar una zona para el sistema (normalmente en ROM) y otra para el usuario (RAM), de tal forma que se asignaron los primeros 640 Kb a la RAM (aquí se encuentra en las posiciones mas bajas parte del sistema: la tabla de interrupciones y la zona de comunicaciones de la BIOS y el área de trabajo del DOS) y el resto del espacio de direcciones estaba destinado al sistema y protegido del acceso de los programas de usuario. De los 640 Kb hasta el Mbyte, nos encontramos, con la RAM de vídeo, ROM de vídeo, ROM del controlador de disco duro, zona sin asignar y ROM-BIOS.

El método de acceso a memoria utiliza lo que se denomina direcciones segmentadas, el motivo está en que la palabra del microcomputador es de 16 bits, es decir la forma en que se comunican las distintas partes que forman el computador y los registros implicados son de 16 bits ($2^{16} = 65.536$), este método implica a dos valores de 16 bits, el segmento y el desplazamiento, en lo que se denomina suma desplazada, logrando un numero de 20 bits con el que se puede acceder a todo el rango de direcciones de memoria. La suma desplazada se logra añadiendo al segmento, 4 bits, que serán los menos significativos y con valor cero, sumando a este resultado, el desplazamiento.



Existe por conveniencia, límites artificiales de 64 Kb que son los bloques del espacio de direcciones de memoria, estos son 16 bloques de 64 Kb, de tal forma que se dice por ejemplo que la ROM-BIOS tiene reservados los dos últimos bloques de memoria. Es decir los 128 Kb situados justo por debajo del Mbyte, que es todo el espacio de memoria cuando el

microprocesador opera en modo real (que es el modo del 8086). Si el microprocesador opera en modo protegido puede gestionar otros límites de memoria, dependiendo del número de bits de su BUS de direcciones, según la fórmula $2^{(n^{\circ}_{\text{de_bits}})}$. El problema es que al mantener la compatibilidad hacia abajo del DOS (los programas realizados sobre versiones anteriores, deben seguir "corriendo" en las actuales) se restringe el espacio de memoria al Mbyte (el DOS trabaja con el microprocesador en modo real), siendo así que a partir del microprocesador 80286 es posible gestionar memoria virtual en modo protegido, aunque el abaratamiento de la memoria física disminuyó el interés por la memoria virtual.

La primera solución a este problema la propusieron Lotus, Intel y Microsoft (LIM), y se trata de una solución que utiliza hardware (no necesario de 386 en adelante) y software, especificando unas condiciones que se deben cumplir para tal efecto, denominada especificaciones de memoria expandida (EMS). Se necesita como requerimiento hardware una tarjeta de ampliación de memoria que cumpla las especificaciones EMS junto con el gestor de memoria expandida (EMM), y el programa de aplicación que usa la memoria expandida que es la parte software. Su modo de funcionamiento se apoya en la utilización de la memoria reservada o memoria superior que se corresponde con los últimos 384K del espacio de direcciones (los últimos 6 bloques), ya que parte de este espacio no está asignado a ninguna memoria física, y el microprocesador puede acceder a ella en modo real. Para ello se reserva un bloque de memoria (64 Kb) de la memoria superior que esté libre, denominado Marco de Página y que está dividido en 4 páginas de 16 Kb. Esta memoria utiliza conmutación de páginas, una o más ventanas (páginas) del marco son asignadas a un programa de aplicación que sepa trabajar con datos fragmentados en páginas de 16 Kb, cuando esta información no es necesaria para ejecutar el programa (por ejemplo una porción no utilizada de una hoja de cálculo que sin embargo debe estar en memoria principal), se la puede mapear a la memoria expandida. La conmutación de páginas consiste en traer a memoria superior la página que se vaya a utilizar para la ejecución del programa y llevar a la memoria expandida las partes no necesarias (páginas) de la memoria superior. EMS está pues diseñada para almacenamiento, no para la ejecución de programas, llegando con la versión 4.0 a proporcionar acceso hasta 32Mbytes de memoria expandida.

Una segunda alternativa, más actual y más rápida es la memoria extendida. Básicamente la memoria extendida es RAM por encima del Mbyte del espacio de direcciones de memoria, y que en un computador basado en 286 (24 bits de bus de direcciones) o superior (32 bits de bus de direcciones), se podría acceder a esta memoria por encima del Mb, en concreto hasta 16 Mb en el 286 y 4Gb en el 386. Para esto es necesario conmutar el microprocesador a modo protegido y luego antes de pasar el control al DOS, regresar al modo real, ya que el DOS es un sistema operativo en modo real. Para gestionar la memoria extendida con el DOS, Lotus, Intel, AST Research y Microsoft establecieron un estándar llamado especificación de memoria extendida o XMS (eXtended Memory Specification). XMS no permite (al igual que EMS) a los programas que corren bajo DOS, ejecutarse en memoria extendida, sino que se utiliza como memoria principal de almacenamiento, salvo las aplicaciones extended-DOS, que secretamente conmutan del modo real al protegido, como son Windows, Autocad vers 11, Lotus 1-2-3 versión 3.0. Desde la versión del DOS 3.2, la memoria extendida estaba disponible para usarse como dispositivo de almacenamiento, y se podían instalar Discos RAM, caches de disco, o buffers de impresora.

La memoria alta (HMA), es la zona de memoria comprendida entre 1024 Kb y 1088 Kb, es decir los primeros 64 Kb de memoria extendida. El DOS 5.0 puede cargar parte del DOS en memoria alta dejando libre de esta forma unos 44 Kb de memoria convencional (en computadores con microprocesador 286 en adelante). Para ello es necesario instalar el controlador HIMEN.SYS que carga en memoria la especificación XMS 2.0 y activar en CONFIG.SYS la orden DOS=HIGH, que carga las rutinas del archivo del sistema (MSDOS-SYS), buffers del disco y parte del

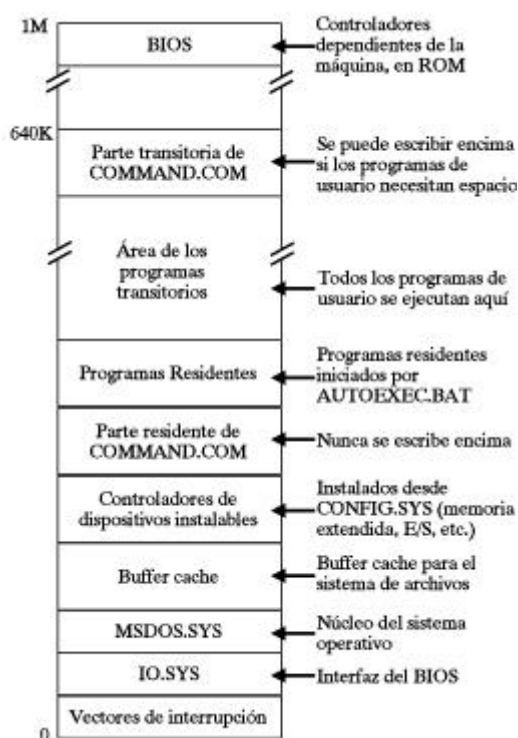
COMMAND.COM en memoria alta. El DOS puede acceder a estos 64 Kb de memoria alta, sin conmutar el microprocesador al modo protegido (sigue siendo un sistema operativo en "modo real"). Se puede simular memoria expandida en la memoria extendida, solo en microprocesadores 386 en adelante y con el software adecuado. El archivo EMM386.EXE realiza esta simulación, habilitando o deshabilitando la compatibilidad con memoria expandida.

Ya vimos que la zona comprendida entre los 640 Kb y 1024 Kb (1 Mb) se denominaba memoria superior y que estaba reservada al sistema, encontrándonos allí con RAM de vídeo, ROM de vídeo, ROM del controlador del disco duro y ROM del BIOS. Vimos también que existían bloques de memoria (direcciones) sin usar en los que se apoyaba la gestión de memoria expandida (marco de página). Con el DOS es posible también cargar controladores de dispositivos y programas residentes en estas direcciones de memoria, liberando así memoria convencional que es donde "corren" las aplicaciones de usuario desarrolladas bajo DOS. Es necesario cumplir unos requisitos de hardware y software para poder utilizar los bloques de memoria superior (UMB): un microprocesador 386 o superior y 350 Kb de memoria extendida libre que cumpla las especificaciones XMS, ya que se necesitan para mapear los huecos libres de la memoria superior. En software el requisito es instalar HIMEN.SYS (aporta las especificaciones XMS 2.0) e instalar el gestor de memoria EMM386.EXE, además hay que activar la orden DOS=UMB. Con ello es posible cargar los controladores de dispositivos en memoria superior con los mandatos *DEVICEHIGH* y los programas residentes con *LOADHIGH*.

6.7.4. Arranque del DOS.

Cuando se pone en marcha la computadora, el BIOS pasa las rutinas de comprobación y chequeo y carga el sector de arranque del disquete situado en A, si lo hubiera y estuviese formateado con el sistema (aceptando que se ha situado el primer arranque desde el disquete A en la CMOS-RAM, situación que se da por defecto). En el caso de que no exista disquete en A, buscaría en la tabla de particiones del disco duro para ver cual es la partición puesta como activa, cargando y ejecutando el programa maestro de arranque del sector de arranque de la partición activa.

El siguiente paso es cargar en memoria el IO.SYS (IBMIO.COM) que esta en las primeras posiciones del área de datos, este archivo contiene el software encargado de gestionar los servicios de entrada/salida junto con un pequeño programa SYSINIT, encargado de la inicialización del sistema. SYSINIT se encarga de cargar en memoria el archivo MSDOS.SYS (IBMDOS.COM). Buscando posteriormente en el directorio raíz del área de datos el archivo CONFIG.SYS, utilizando la



Distribución del espacio de memoria al terminar el procedimiento de arranque

información de dicho archivo (número de buffers, archivos abiertos al mismo tiempo, controladores de dispositivos instalables, etc.) para configurarse a así mismo en la memoria. Si no lo encuentra coge los valores por defecto con el mismo propósito. Por ultimo SISINIT localiza y ejecuta (pasa el control) al interprete de mandatos (COMMAND.COM), instalándose en memoria los mandatos residentes del DOS y presentado en pantalla el prompt del DOS. Posteriormente se ejecuta el archivo AUTOEXEC.BAT (si esta creado y en el directorio raíz), consiguiéndose con esto la personalización del sistema (AUTOEXEC.BAT y CONFIG.SYS). Si no se encuentra el archivo AUTOEXEC.BAT se ejecutan las ordenes del DOS: DATE y TIME.

6.7.5. Organización de archivos en disco.

Una de las tareas mas importantes que desarrolla DOS es la de controlar el sistema de gestión de archivos en disco. Cada disco dispone de un directorio raíz que contiene los detalles de cada uno de los archivos del área de datos, entre los cuales están los subdirectorios. De esta forma los directorios constituyen una estructura jerárquica, en forma de árbol. El usuario está conectado a un determinado directorio (el directorio de trabajo) y a menos que se especifique otra cosa, todos los archivos se buscan o se crean en él. Para localizar un archivo fuera del directorio de trabajo, hay que utilizar el *path* que contiene los nombres de todos los subdirectorios que forman su ruta de acceso, empezando por el directorio raíz.