

Clase 8

Sistemas Operativos

8.1 Introducción

En clases anteriores hemos mencionado a los sistemas operativos dentro de la clasificación del software.

En esta clase profundizamos nuestro estudio de los sistemas operativos con una breve historia. La historia es interesante y también permite ofrecer una visión general de los principios de los sistemas operativos.

Examinaremos los objetivos y funciones de los sistemas operativos. Luego mostraremos cómo han evolucionado los sistemas operativos, desde los primitivos sistemas en lotes hasta los sofisticados sistemas multitarea y multiusuario con interfaz gráfica.

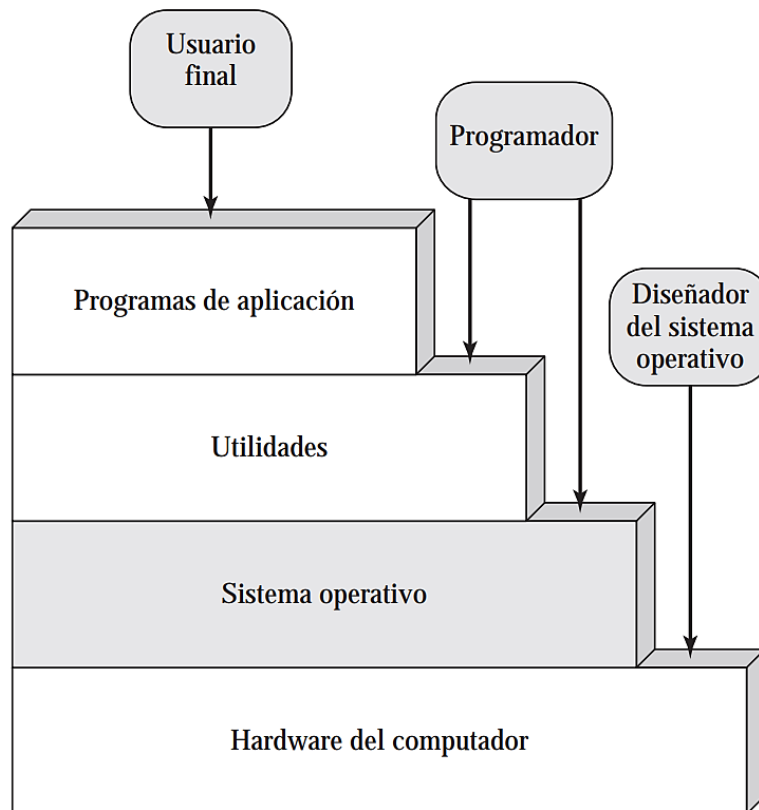
8.2 Objetivos y funciones del sistema operativo

Un sistema operativo es un programa que controla la ejecución de aplicaciones y programas y que actúa como interfaz entre las aplicaciones y el hardware del computador. Se puede considerar que un sistema operativo tiene los siguientes tres objetivos:

- **Facilidad de uso:** Un sistema operativo debe facilitar el uso del dispositivo informático.
- **Eficiencia:** Un sistema operativo permite que los recursos de un sistema de computación se puedan utilizar de una manera eficiente.
- **Capacidad para evolucionar:** Un sistema operativo se debe construir de tal forma que se puedan desarrollar, probar e introducir nuevas funciones en el sistema sin interferir con su servicio.

8.2.1 El sistema operativo como interfaz usuario / máquina

El hardware y software utilizados para proporcionar aplicaciones a los usuarios se pueden ver de forma jerárquica o en capas, tal y como se muestra en la siguiente figura:



El usuario de las aplicaciones, es decir, el usuario final, normalmente no se preocupa por los detalles del hardware de la computadora. Por lo tanto, el usuario final ve un sistema de computación en términos de un conjunto de aplicaciones.

Una aplicación se puede expresar en un lenguaje de programación y normalmente es desarrollada por un programador. Si un programador tuviera que desarrollar una aplicación como un conjunto de instrucciones en código máquina que se encargaran de controlar completamente el hardware del computador, se enfrentaría a una labor extremadamente compleja.

Para facilitar esta tarea, se proporcionan un conjunto de programas de sistema. Algunos de estos programas se conocen como utilidades.

Estos programas utilizan frecuentemente funciones que asisten al programador en las fases de creación de programas, gestión de ficheros y control de los dispositivos de Entrada/Salida. Un programador hará uso de estas utilidades cuando desarrolle una aplicación, y las aplicaciones, invocarán a las utilidades durante su ejecución para llevar a cabo ciertas funciones.

El programa de sistema más importante es el sistema operativo.

El sistema operativo oculta los detalles del hardware al programador y le proporciona una interfaz apropiada para utilizar el sistema. Actúa como mediador, haciendo más fácil al programador y a la aplicación el acceso y uso de dichas utilidades y servicios.

De forma resumida, el sistema operativo proporciona normalmente los siguientes servicios:

- **Desarrollo de programas.** El sistema operativo proporciona una variedad de utilidades y servicios, tales como editores y depuradores, para asistir al programador en la creación de los programas. Normalmente, estos servicios se ofrecen en la forma de utilidades que, aunque no forman parte del núcleo del sistema operativo, se ofrecen paralelamente a dicho sistema y se conocen como herramientas de desarrollo de programas de aplicación o IDE (entorno de desarrollo integrado).
- **Ejecución de programas.** Se necesita realizar una serie de pasos para ejecutar un programa. Las instrucciones y los datos se deben cargar en memoria principal. Los dispositivos de E/S y los ficheros se deben inicializar, y otros recursos deben prepararse. Los sistemas operativos realizan estas labores de planificación en nombre del usuario.
- **Acceso a dispositivos de E/S.** Cada dispositivo de E/S requiere su propio conjunto de instrucciones o señales de control para cada operación, organizadas en un conjunto de drivers. El sistema operativo proporciona una interfaz uniforme que esconde esos detalles de forma que los programadores puedan acceder a dichos dispositivos utilizando lecturas y escrituras sencillas.
- **Acceso controlado a los ficheros.** Para el acceso a los ficheros, el sistema operativo debe reflejar una comprensión detallada no sólo de la naturaleza del dispositivo de E/S (disco rígido, ssd, pendrive, etc.), sino también de la estructura de los datos contenidos en los ficheros del sistema de almacenamiento (NTFS, FAT32, exFAT, ext3, ext4, etc.). Adicionalmente, en el caso de un sistema con múltiples usuarios, el sistema operativo puede proporcionar mecanismos de protección para controlar el acceso a los ficheros.
- **Acceso al sistema.** Para sistemas compartidos o públicos, el sistema operativo controla el acceso al sistema completo y a recursos del sistema específicos. La función de acceso debe proporcionar protección a los recursos y a los datos, evitando el

uso no autorizado de los usuarios y resolviendo conflictos en el caso de conflicto de recursos.

- **Detección y respuesta a errores.** Se pueden dar gran variedad de errores durante la ejecución de un sistema informático. Éstos incluyen errores de hardware internos y externos, tales como un error de memoria, o un fallo en un dispositivo; y diferentes errores software, tales como la división por cero, el overflow, el underflow, el intento de acceder a una posición de memoria prohibida o la incapacidad del sistema operativo para conceder la solicitud de una aplicación. En cada caso, el sistema operativo debe proporcionar una respuesta que elimine la condición de error, suponiendo el menor impacto en las aplicaciones que están en ejecución. La respuesta puede oscilar entre finalizar el programa que causó el error hasta reintentar la operación o simplemente informar del error al usuario.
- **Contabilidad.** Un sistema operativo puede recoger estadísticas de uso de los diferentes recursos y monitorizar parámetros de rendimiento, tales como el tiempo de respuesta. En cualquier sistema, esta información es útil para anticipar las necesidades de mejoras futuras y para optimizar el sistema a fin de mejorar su rendimiento.

8.3 El sistema operativo en la memoria principal y como gestor de recursos

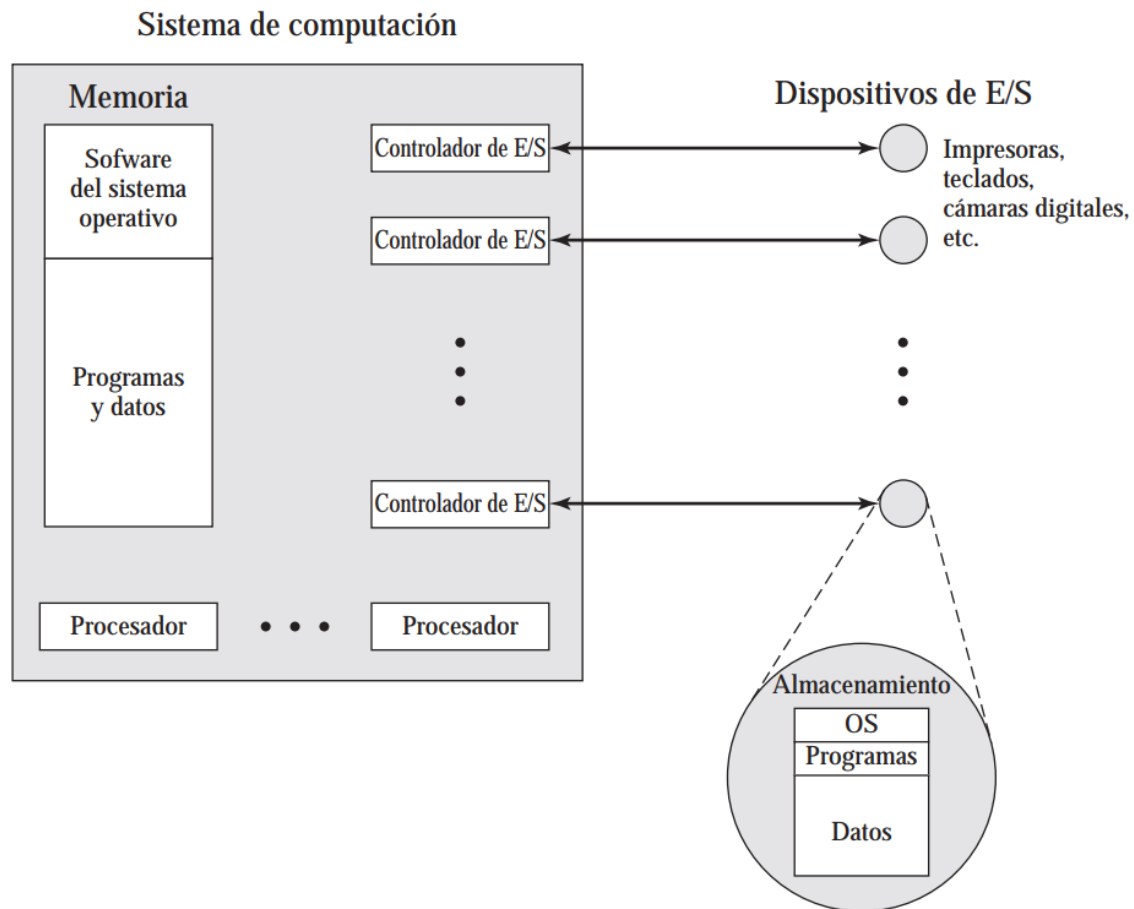
Una porción del sistema operativo se encuentra en la memoria principal. Esto incluye el kernel, o núcleo, que contiene las funciones del sistema operativo más frecuentemente utilizadas y, en cierto momento, otras porciones del sistema operativo actualmente en uso. El resto de la memoria principal contiene programas y datos de usuario.

La asignación de este recurso (memoria principal) es controlada de forma conjunta por el sistema operativo y el hardware de gestión de memoria del procesador, como se verá.

El sistema operativo decide cuándo un programa en ejecución puede utilizar un dispositivo de E/S y controla el acceso y uso de los ficheros.

El procesador es también un recurso, y el sistema operativo debe determinar cuánto tiempo de procesador debe asignarse a la ejecución de un programa de usuario particular. En el caso de un sistema multiprocesador, esta decisión debe ser tomada por todos los procesadores.

La siguiente figura muestra los principales recursos gestionados por el sistema operativo:



Como vemos parte del sistema operativo se almacena en la memoria principal, junto con otros programas y datos. Por otra parte, en el almacenamiento o memoria secundaria se almacena el Sistema Operativo (OS) completo.

El sistema operativo interactúa con los distintos dispositivos de entrada y salida mediante los controladores de E/S o drivers.

8.4 Facilidad de evolución de un sistema operativo

Un sistema operativo importante debe poder evolucionar en el tiempo, por las siguientes razones:

- **Nuevos tipos de hardware.** Por ejemplo, el mítico Windows 95 no incorporaba de base soporte para USB o puerto AGP. Posteriores actualizaciones de dicho sistema fueron incorporando estos puertos a medida que fueron popularizándose.
- **Nuevos servicios.** En respuesta a la demanda del usuario o en respuesta a las necesidades de los gestores de sistema, el sistema operativo puede o debe ofrecer nuevos servicios. Siguiendo con el ejemplo de Windows 95, en un principio no

incorporaba soporte para el sistema de archivos FAT32, tampoco soportaba, porque no existían, las librerías de gráficos Direct x. Con la instalación de Internet Explorer 4.0 se incluyó una actualización llamada *Actualización de escritorio de Windows* que confería a Windows 95 una interfaz de usuario muy similar al que sería su sucesor, Windows 98.

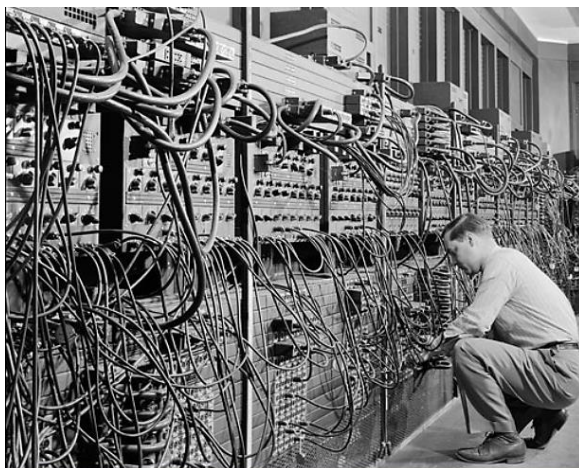
- **Resolución de fallos.** Cualquier sistema operativo tiene fallos. Estos fallos se descubren con el transcurso del tiempo y se resuelven. Por supuesto, esto puede implicar la introducción de nuevos fallos.

La necesidad de cambiar regularmente un sistema operativo introduce ciertos requisitos en su diseño. Un hecho obvio es que el sistema debe tener un diseño modular, con interfaces entre los módulos claramente definidas, y que debe estar bien documentado.

8.4 La evolución de los sistemas operativos

Procesamiento en serie: 1946 hasta 1952

Con las primeras computadoras digitales, desde finales de los años 40 hasta mediados de los años 50, el programador interaccionaba directamente con el hardware del computador; no existía ningún sistema operativo. Estas máquinas eran utilizadas desde una consola que contenía luces, interruptores, algún dispositivo de entrada y una impresora. Los programas en código máquina se cargaban a través del dispositivo de entrada (por ejemplo, un lector de tarjetas). Si un error provocaba la parada del programa, las luces indicaban la condición de error. El programador podía entonces examinar los registros del procesador y la memoria principal para determinar la causa de error. Si el programa terminaba de forma normal, la salida aparecía en la impresora.



La ENIAC

La ENIAC (Electronic Numerical Integrator and Computer), cuyo dispositivo de entrada era un lector de tarjetas perforadas y su salida era un perforador de tarjetas (IBM 405). No poseía un sistema operativo, ni un lenguaje de programación, ni siquiera assembler, de manera que todas las instrucciones eran escritas en las tarjetas perforadas mediante código de máquina.

Estas primeras computadoras se comunicaban con el operador mediante unas pequeñas luces, que se encendían o se apagaban al acceder a determinadas posiciones de memoria o ejecutar ciertas instrucciones.

Procesamiento en lotes: 1952 hasta 1960

Las primeras máquinas eran muy caras y, por lo tanto, era importante maximizar su utilización. El tiempo malgastado en la planificación y configuración de los trabajos era inaceptable. Durante el tiempo que un dispositivo está leyendo las tarjetas perforadas, tanto el dispositivo procesador como el dispositivo de salida no están haciendo ningún trabajo útil.

Para mejorar su utilización, se desarrolló el concepto de sistema operativo en lotes. Parece ser que el primer sistema operativo en lotes (y el primer sistema operativo de cualquier clase) fue presentado en 1952 por General Motors para el uso de un IBM 701.

IBM 701 supone un antes y un después en la historia al ser considerado como el primer sistema de procesamiento de información digital. Mientras en anteriores computadores era el propio usuario el que, junto a la información, añadía las operaciones a realizar, en el 701 los programas eran almacenados en una memoria interna, como se hace en la actualidad. También se lo considera una de las piezas clave para el abandono de las tarjetas perforadas.



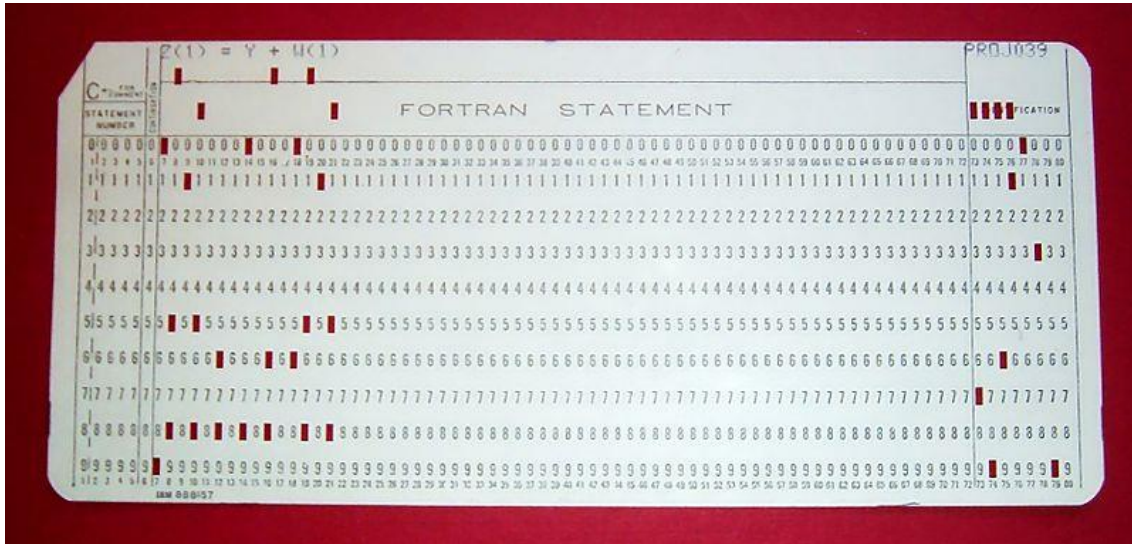
La IBM 701

La idea central bajo el esquema de procesamiento en lotes sencillo es el uso de una pieza de software denominada **monitor**. Con este tipo de sistema operativo, el usuario no tiene que acceder directamente a la máquina. En su lugar, el usuario envía un trabajo a través de una tarjeta o cinta a la computadora, que crea un sistema por lotes con todos los trabajos enviados y coloca la secuencia de trabajos en el dispositivo de entrada, para que lo utilice el **monitor**. Cuando un programa finaliza su procesamiento, devuelve el control al **monitor**, punto en el cual dicho **monitor** comienza la carga del siguiente programa.

Al programa monitor, o monitor residente (MR) se lo considera la forma más primitiva de sistema operativo.

La Universidad de California en Livermore desarrolló un lenguaje con su sistema de compilación y ejecución para su IBM 701, llamado "KOMPILE".

El sucesor del 701 fue el IBM 704, introducido cuatro años después del 701. En dicho sistema se introdujo el uso del lenguaje de programación FORTRAN.



Código FORTRAN en una tarjeta perforada

Estas computadoras funcionaban con tarjeta perforada, que permitían introducir programas. Durante los años 50, la forma más común de interactuar con una computadora era mediante un teletipo, que se conectaba directamente a este e imprimía todos los datos de una sesión informática.



Un teletipo

Multiprogramación y multiusuario: 1960 hasta 1970

El procesador se encuentra frecuentemente ocioso, incluso con el secuenciamiento de trabajos automático que proporciona un sistema operativo en lotes. El problema consiste en que los dispositivos de E/S son lentos comparados con el procesador.

La computadora malgastaba aproximadamente el 96% de su tiempo esperando a que los dispositivos de E/S terminen de transferir datos hacia y desde el fichero. Esta situación, donde existe un único programa, se denomina monoprogramación. El procesador ejecuta durante cierto tiempo hasta que alcanza una instrucción de E/S.

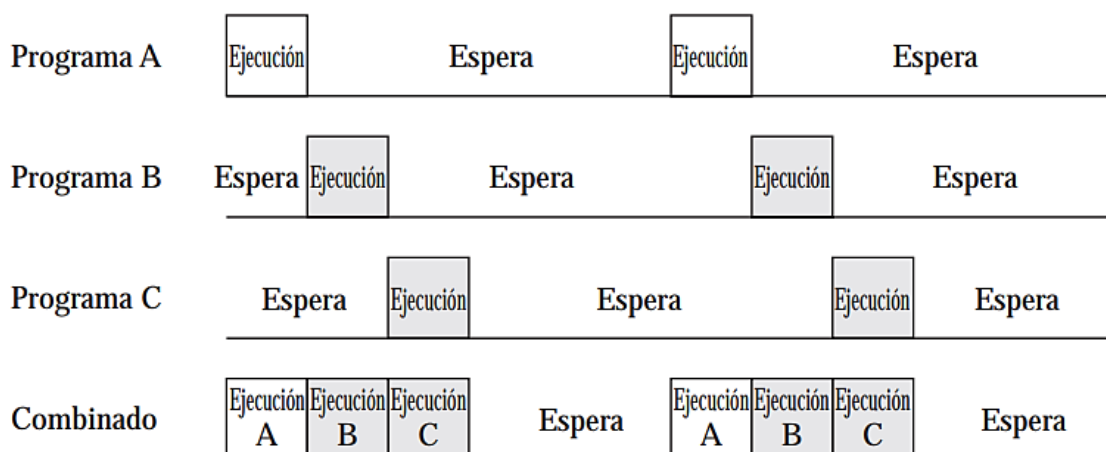
Entonces debe esperar que la instrucción de E/S concluya antes de continuar.

Esta ineficiencia puede evitarse. Para la época existía suficiente memoria para contener al sistema operativo (monitor residente) y un programa de usuario.

Incluso había espacio para el sistema operativo y dos (o más) programas de usuario. Cuando un trabajo necesita esperar por la E/S, se puede asignar el procesador al otro trabajo, que probablemente no esté esperando por una operación de E/S.

Este enfoque se conoce como multiprogramación o multitarea.

En la siguiente imagen se observa un esquema que refleja el trabajo de la multiprogramación, en el cual se aprovechan los tiempos de espera para ejecutar otros programas, con lo cual se reduce el tiempo ocioso total:



En esta época también se desarrollaron los sistemas compartidos, en los que los usuarios podían utilizar la computadora a través de terminales. Surgieron sistemas de tiempo real, que se caracterizan por proveer una respuesta inmediata.

Los Sistemas Operativos más significativos que se desarrollaron en esta etapa fueron:

- ❖ Multics (1964), sistema operativo multiusuario - multitarea desarrollado por los laboratorios Bell de AT&T
- ❖ OS/360 (1964), El OS/360 fue desarrollado por IBM. Inicialmente procesaba los trabajos secuencialmente (procesamiento por lotes); posteriormente la Multiprogramación con un número Fijo de Tareas añadió multitarea, pero solo permitía un número fijo de tareas concurrentes, cada una teniendo un lugar predefinido

en la memoria. Finalmente, se permitió un número variable de tareas cuya memoria podía cambiar dinámicamente.

Hasta esta época aún continuaba el uso del teletipo como manera de visualizar los resultados obtenidos.

Intérpretes de líneas de comandos: 1970 a 1980

En esta década aparece UNIX, la base de la gran mayoría de los sistemas operativos que existen hoy en día.

En los años 70 se produce un boom en cuestión de ordenadores, acercando las computadoras al público general.

Esto hace que se multiplique el desarrollo, creándose, por ejemplo, el lenguaje de programación C (diseñado específicamente para reescribir por completo el código UNIX).

En esta década las terminales para interactuar con la computadora pasaron del formato impreso a los VDU (video display unit – unidad de visualización de video), las cuales mostraban la información ingresada hacia y obtenida por el ordenador en una pantalla de tubos de rayos catódicos (CRT).



Teletipo ASR model 33



VDU DEC VT100

La interacción, en el caso de Unix, se hacía mediante el intérprete de línea de comandos (Shell) V6, introducido en 1971, también conocido como el "Thompson Shell", por su desarrollador Key Thompson. Este Shell es conocido también como osh. El programa estaba escrito en unas 900 líneas de código en lenguaje C.

El Shell introdujo una serie de bloques de construcción que hacen a la filosofía Unix. Sin embargo, un problema que tenía el shell de Thompson era la imposibilidad de generar scripts.

El Bourne Shell fue creado por Stephen Bourne en 1977 para la versión V7 de UNIX, el mismo incluso puede encontrarse en nuestros días en algunas distribuciones de Unix.

El Bourne shell tenía básicamente dos grandes propósitos:

- ❖ Ser un intérprete interactivo de ejecución de comandos para el sistema operativo.
- ❖ Poder correr scripts: un script es un programa que puede ser ejecutado por el Shell del sistema operativo.

Además, Bourne Shell ofrecía muchas otras características que el Shell de Thompson no tenía:

- Estructuras de Control
- Bucles
- Variables en los scripts

Sistemas operativos modernos: 1980 a la actualidad

Con la creación de los circuitos LSI (integración a gran escala), empezó el auge de los ordenadores personales.

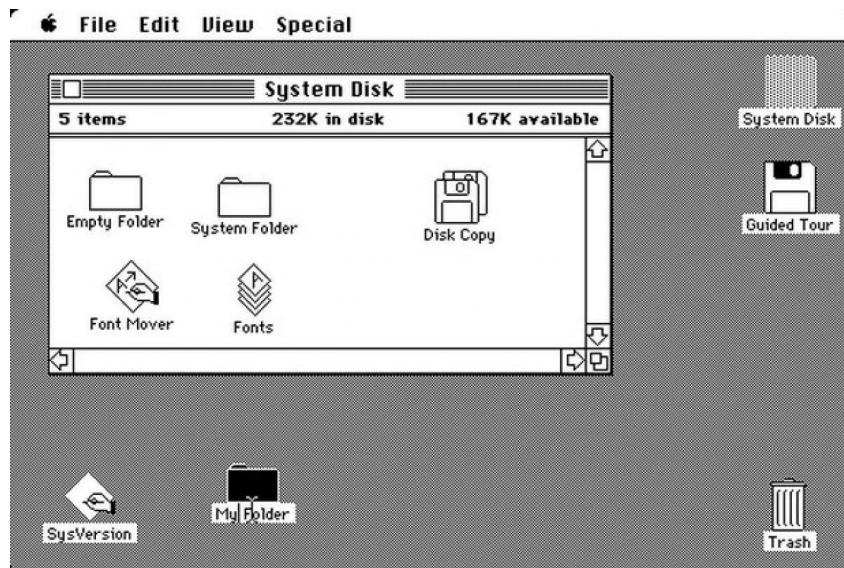
Como consecuencia de este crecimiento exponencial de usuarios, la gran mayoría de ellos sin ningún conocimiento sobre lenguajes de bajo o alto nivel, la prioridad a la hora de diseñar un sistema operativo pasó a ser la facilidad de uso por sobre la velocidad, mejorando así las primeras interfaces de usuario, buscando que sean más amigables, surgiendo menús, e interfaces gráficas.

En 1981 Microsoft compró un sistema operativo llamado QDOS que, tras realizar unas pocas modificaciones, se convirtió en la primera versión de MS-DOS (Microsoft Disk Operating System). A partir de aquí se sucedieron una serie de cambios hasta llegar a la versión 8 en Windows Milenium, a partir de la cual MS-DOS dejó de existir como un componente del Sistema Operativo.

En la década de los 90 hace su aparición Linux, publicándose la primera versión del núcleo (kernel) en septiembre de 1991, que posteriormente se uniría al proyecto GNU, un sistema operativo completamente libre, similar a UNIX, al que le faltaba para funcionar un núcleo funcional. Hoy en día la mayoría de la gente conoce por Linux al Sistema Operativo que realmente se llama GNU/Linux.

Por su parte Apple se enfocó en 1979 en su línea de ordenadores Macintosh, lanzándolo al mercado en 1984 y cuyo sistema operativo Mac OS (hoy conocido como Classic Mac OS) se caracterizaba por ser un sistema operativo con interfaz gráfica de usuario, y sin líneas de comandos. Algo revolucionario para la época.

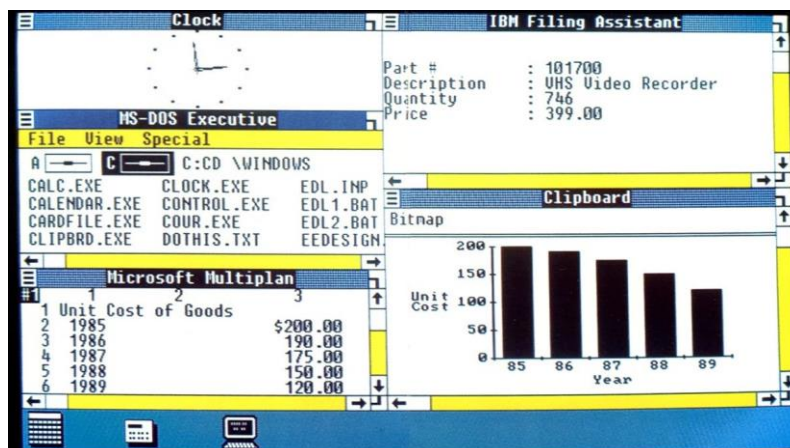
Es así como el 24 de enero de 1984, Steve Jobs presentó al mundo el primer Apple Macintosh. Fue la primera computadora personal en incorporar el uso del mouse y una interfaz gráfica, a un precio relativamente asequible para el público.



Interfaz del Mac OS de 1984

La revolucionaria interfaz del Mac OS impulsó el nacimiento de Windows, cuya primera versión 1.0, presentada en diciembre de 1985, compitió con el sistema operativo de Apple. Sin embargo carecía de las mismas funcionalidades que su competencia y logró muy poca popularidad.

Además, Windows 1.0 no era un sistema operativo completo; sino que era una extensión gráfica que corría sobre el sistema operativo MS-DOS.



Interfaz de Windows 1.0

No sería hasta 1990, año en el cual Microsoft lanzaría Windows 3.0, que Microsoft consiguiese un éxito comercial, vendiendo 2 millones de

copias en los primeros seis meses. Presentaba mejoras en la interfaz de usuario y en la multitarea. Recibió un lavado de cara en Windows 3.1, que se hizo disponible para el público en general el 1 de marzo de 1992.

La historia posterior de Windows es relativamente bien conocida, pasando por (en líneas generales): Windows 95, Windows 98, Windows XP, Windows Vista, Windows 7, Windows 8.1, hasta llegar al actual Windows 10.

8.5 Gestión de memoria y memoria virtual

Las computadoras utilizan la memoria RAM para almacenar los archivos y datos que necesitan tanto el sistema operativo como el software que estemos ejecutando; su buen rendimiento garantiza un funcionamiento óptimo, pero, tarde o temprano, siempre termina por llenarse. Es en ese momento cuando el sistema operativo necesita recurrir a la memoria virtual.

En el caso de Windows, la memoria virtual hace uso de un archivo en la unidad de almacenamiento que tengamos asignada, sea un disco duro tradicional o un SSD; dicho archivo se denomina `pagefile.sys` (se encuentra oculto en el directorio raíz) donde va almacenando los datos que no caben en la memoria RAM pero que son necesarios para el funcionamiento del PC.

Así, cuando trabajamos con aplicaciones muy exigentes o tenemos varias funcionando al mismo tiempo el sistema se puede notar más lento pues en ese momento Windows está recurriendo al archivo de paginación.

Dado que la memoria RAM se ha visto desbordada; al utilizar la memoria virtual se evitan los errores, cuelgues y la inestabilidad del sistema, pero a cambio el rendimiento desciende considerablemente.

Por lo tanto, es fácil concluir que cuanto más Memoria RAM tengamos en el equipo es mejor, y notaremos más aún la diferencia cuanto más exigente sea el software que utilizamos (en términos de consumo de memoria).

Otra conclusión es que al utilizar memoria virtual el sistema se ralentiza aún más si el archivo de paginación se almacena en un disco rígido tradicional, que si se lo hace en un disco SSD. Si bien en el pasado no era recomendable almacenar el archivo de paginación en un disco SSD porque estos tienen un limitado número de escrituras antes de que se produzca una falla, hoy en día la tecnología permite que este límite sea cada vez más alto.