

Oposiciones de Secundaria)

TEMA 15

SISTEMAS OPERATIVOS. COMPONENTES. ESTRUCTURA. FUNCIONES. TIPOS

ÍNDICE

1. INTRODUCCIÓN
2. EVOLUCIÓN DE LOS SISTEMAS OPERATIVOS
3. COMPONENTES DE UN SISTEMA OPERATIVO
 - 3.1. Programas de utilidad
 - 3.2. Programas de aplicación
 - 3.3. Programas de servicio
 - 3.4. Programas de control
4. ESTRUCTURA DE LOS SISTEMAS OPERATIVOS
5. FUNCIONES DE LOS SISTEMAS OPERATIVOS
6. TIPOS DE SISTEMAS OPERATIVOS
 - 6.1. Multiprogramación
 - 6.2. Tiempo compartido
 - 6.3. Sistemas más usuales
7. BIBLIOGRAFÍA

1. INTRODUCCIÓN

Sin software, un ordenador es inútil, no sirve para nada. Es el software el que permite que un ordenador pueda almacenar, procesar y recuperar información. Por ello, surge la necesidad de aislar al usuario de la complejidad del hardware. Para conseguirlo, se añade una capa de software sobre el hardware puro que se encarga de gestionar todos los elementos del sistema y de presentar al usuario una interfaz que haga más fácil su entendimiento. Esta capa de software tan ligada al hardware recibe el nombre genérico de *sistema operativo*.

El **sistema operativo** es el software básico y fundamental de un sistema informático. Sin su existencia, el elemento hardware sólo sería una entidad física sin utilidad, ya que no podríamos asignarle ningún tipo de tarea. Por consiguiente, se trata de una de las partes más fundamentales de un sistema informático y es importante conocerlo. De hecho, algunas empresas de software desarrollan sus productos sólo para ciertos tipos de sistemas operativos.

Una primera aproximación nos presentaría al sistema operativo como un tipo de software que controla el funcionamiento del sistema físico (hardware), ocultando sus detalles al usuario, permitiéndole así trabajar con el ordenador de una forma fácil y segura. El sistema operativo enmascara los recursos físicos, permitiendo un manejo sencillo y flexible de los mismos, y proporciona determinados servicios de alto nivel al usuario que no están directamente presentes en el hardware subyacente. En este sentido, hace posible la ejecución y uso de aplicaciones por parte del usuario. De esta forma, el sistema operativo junto con el hardware (máquina física) definen una máquina virtual que puede ser utilizada sin conocer las características concretas de los dispositivos hardware.

Se puede definir sistema operativo como un *programa* (o conjunto de programas) *de control* que tiene por objeto *facilitar* el uso del ordenador y conseguir que éste se utilice *eficientemente*.

Es un *programa de control*, ya que se encarga de gestionar y asignar los recursos hardware a los usuarios. Controla los programas de los usuarios y los dispositivos de E/S. Los recursos hardware son:

- El **procesador** (CPU): Dado que es el lugar donde se ejecutan las instrucciones de los programas de aplicación, se debe controlar la forma en que se ejecutan dichos programas.
- La **memoria principal**: Dado que es el dispositivo donde residirán los datos a procesar y los programas a ejecutar, es necesario regular su uso y ocupación.
- Los **dispositivos periféricos**: Estas unidades permiten la comunicación de los programas con el exterior, y debe asegurarse su adecuado direccionamiento y control.

El sistema operativo *facilita el uso del ordenador*. Contiene *módulos de gestión de entradas/salidas* que evitan a los usuarios tener que incluir esas instrucciones cada vez que realizan una operación de ese tipo. Se puede decir que esos módulos del sistema operativo hacen “transparente” al usuario las características hardware concretas de los dispositivos.

El sistema operativo también *hace que el ordenador se utilice eficientemente*. Por ejemplo, en los sistemas de multiprogramación hace que, cuando un programa esté dedicado a una operación de entrada o salida, la CPU pase a atender a otro programa.

2. EVOLUCIÓN DE LOS SISTEMAS OPERATIVOS

Es habitual considerar cuatro etapas o generaciones en la evolución de los sistemas operativos.

Primera etapa

Desde la construcción de los primeros ordenadores (1943) hasta aproximadamente el año 1955 no existen sistemas operativos. El usuario que deseaba utilizar un ordenador debía apuntarse en un cuadro de reserva de horas. Cuando llegaba su turno, el programador, que hacía también las funciones de operador, introducía su programa y esperaba a que poco a poco fuesen escribiéndose en una impresora los resultados. El programa se daba a través de tarjetas perforadas y se redactaba a mano en lenguaje máquina. Los datos también se suministraban al ordenador con tarjetas perforadas.

Posteriormente, se idearon los ensambladores. Después de ensamblar el programa, el programa objeto obtenido (en lenguaje máquina) se montaba manualmente; es decir, se unía con otros paquetes de tarjetas que contenían las funciones utilizadas en el programa y no definidas en el mismo. El paquete total de tarjetas, por último, se ejecutaba con los datos del programa.

Con esta forma de trabajar, no se obtiene un aprovechamiento adecuado del ordenador. Cuando la CPU está trabajando, los dispositivos de E/S están inactivos, y cuando un dispositivo de E/S está actuando, las demás unidades, incluyendo la CPU, no trabajan. Las grandes posibilidades operativas de la CPU, en cuanto a velocidad de ejecución, no se aprovechan adecuadamente.

Como resumen de la primera etapa, podemos decir que la introducción y control de la ejecución de programas se hacía manualmente, uno a uno: no existían sistemas operativos.

Segunda etapa

En esta etapa (aproximadamente de 1956 a 1963) en vez de utilizar sólo lectoras de tarjetas e impresoras, se utilizan soportes de información intermedios, como cintas magnéticas.

Inicialmente, los traductores y funciones de biblioteca se encontraban en cinta magnética, y el usuario introducía mediante tarjetas el programa y los datos.

Posteriormente, se dio un paso importante al utilizar un ordenador satélite para gestionar las entradas/salidas y un ordenador principal para leer y escribir en cinta magnética. Se disponía de programas, suministrados por el fabricante, que volcaban sobre cinta magnética los programas en tarjetas de los usuarios.

Para hacer más fácil el trabajo del usuario, apareció el denominado **procesamiento de trabajos por lotes** o *por lotes-serie*, consistente en intercalar entre las tarjetas de un programa y otro, una serie de tarjetas de control con instrucciones u órdenes del *lenguaje de control de trabajos* (JCL – Job Control Language). Así, por ejemplo, había tarjetas que indicaban las características del trabajo a procesar (\$JOB), la finalización de un trabajo (\$END), el compilador que había que utilizar, cuándo debía ejecutarse el programa (\$RUN), cuándo se debía cargar el programa en memoria (\$LOAD), etc.

Este tipo de funcionamiento era posible gracias a la existencia de un programa de control denominado **monitor** (residente en memoria) que controlaba el secuenciamiento de los trabajos. El monitor se puede considerar como el primer sistema operativo. Se encargaba fundamentalmente de dos tareas:

- Identificar e interpretar las tarjetas de control. Una vez interpretada se daba paso a una rutina que realizaba las funciones propias de la tarjeta.
- Asegurar que se inicie el siguiente trabajo del lote cuando termina la ejecución del actual, es decir, una secuenciación automática de los trabajos.

Poco a poco los monitores de lotes se perfeccionaron, incluyendo rutinas de control de E/S, módulos controladores del tiempo de CPU, etc.

Esta segunda etapa la podríamos denominar de “procesamiento de trabajos en serie o por lotes-serie”, o también de utilización de dispositivos de E/S fuera de línea, o de dispositivos de E/S virtuales.

Tercera etapa

La tercera etapa (1963 a 1979) se caracteriza fundamentalmente por el perfeccionamiento del sistema de trabajo por lotes, el desarrollo de sistemas operativos en multiprogramación, que permite trabajar a los usuarios en forma interactiva/conversacional, y la introducción del concepto de memoria virtual.

Cuarta etapa

La cuarta etapa (entre 1980 y la actualidad) se caracteriza por el desarrollo de sistemas operativos para ordenadores personales (CP/M, MS-DOS, OS/2, etc.) y para estaciones de trabajo (UNIX, Windows-NT, etc.).

También se desarrollan sistemas operativos de red y sistemas operativos distribuidos. Los **sistemas operativos en red** no se diferencian sustancialmente de los sistemas operativos convencionales, ya que pueden considerarse como una ampliación de éstos con programas de control y de interfaz con la red y que permiten realizar una sesión de trabajo con un ordenador remoto y copiar ficheros de un ordenador a otro. Los

sistemas operativos distribuidos son mucho más complejos, ya que permiten que un usuario ejecute, de forma transparente, uno o varios programas en varios procesadores trabajando simultáneamente y, posiblemente, con datos distribuidos en diferentes servidores de ficheros, lo cual requiere que el sistema operativo disponga de sofisticados algoritmos de planificación que paralelicen adecuadamente el trabajo a realizar.

3. COMPONENTES DE UN SISTEMA OPERATIVO

Un sistema operativo está formado por un conjunto de programas que utiliza el ordenador para producir un trabajo útil.

Estos programas se suelen clasificar en cuatro tipos: programas de aplicación, programas de utilidad, programas de servicio y programas de control.

3.1. Programas de utilidad

Son los destinados a realizar una función específica. Muchas veces se les considera aparte del sistema operativo propiamente dicho.

Son programas normalmente diseñados para un hardware específico y elaborados por la empresa constructora del ordenador.

Realizan funciones de tipo repetitivo como pueden ser la clasificación de ficheros, la localización de errores, protección de ficheros, etc.

Dentro de los programas de utilidad se distinguen dos clases:

- Los organizadores del sistema.
- Los organizadores de ficheros.

Organizadores del sistema

Se ejecutan bajo la supervisión del sistema operativo mismo y permiten manejar las librerías de programas así como la protección de la información que maneja el ordenador.

Organizadores de ficheros

Se encargan de manipular los datos clasificando o reorganizando los registros de los diversos ficheros.

3.2. Programas de aplicación

Bajo este nombre se designan programas cuya finalidad es resolver los problemas particulares de los usuarios. A veces forman parte de paquetes que suministran ciertas empresas de software de forma general y que pueden ser particularizados en algunos casos muy concretos.

Existen dos tipos de programas de aplicación:

- Programas de gestión.
- Programas de cálculo.

Programas de gestión

Se destinan a los diversos aspectos de gestión de las empresas. Manejan gran cantidad de datos y utilizan cálculos muy sencillos por lo que necesitan gran espacio para su almacenamiento y poca cantidad de memoria central.

Programas de cálculo

Se destinan a funciones científicas y técnicas y requieren gran rapidez de CPU, no mucha memoria y poca gestión de E/S.

Como programas de gestión se pueden citar la impresión de facturas, la gestión de un almacén, la confección de nóminas, etc.

Entre los de cálculo se encuentran los programas de cálculo de estructuras para la construcción, resistencia de materiales, y otros.

3.3. Programas de servicio

Son programas destinados a gestionar y poner a punto los programas de aplicación.

En general, están escritos en un lenguaje comprensible por el usuario y por ello poco comprensible por el ordenador.

Así es necesario, pues, que existan programas traductores que los hagan comprensibles al ordenador, que faciliten su corrección cuando existan errores y su readaptación a casos particulares.

Entre los programas de servicio están:

- * Programas traductores.
- * Programas de gestión de librerías.
- * Programas editores.

Programas traductores

Realizan el paso del lenguaje del usuario (alto nivel) en el cual están escritos al lenguaje que comprende el ordenador (máquina).

Al programa original de alto nivel se le suele llamar programa fuente y al que queda tras la traducción con el nombre de programa objeto.

Los programas traductores, se pueden clasificar según la forma de realizar el proceso de traducción en:

- * Compiladores.
- * Intérpretes.
- * Procesadores.
- * Ensambladores.

Los compiladores traducen todo el programa fuente a código máquina dejándolo dispuesto para su ejecución.

Un intérprete realiza la traducción del programa fuente pero lo hace paso a paso, ejecutando cada instrucción traducida.

Los procesadores modifican ligeramente el programa fuente, traduciéndolo a otro lenguaje intermedio que aún comprende el usuario permitiéndole mayor potencia antes de la traducción completa.

El ensamblador es un compilador sencillo que permite la traducción de una instrucción del lenguaje fuente a otra del lenguaje máquina. No obstante existen ensambladores que poseen macroinstrucciones que generan varias instrucciones objeto por una fuente.

Programas de gestión de librerías

Son programas destinados a conservar y organizar el conjunto de los programas de aplicación.

Entre sus funciones cabe destacar:

- * Incorporación de programas.
- * Anulación o borrado de programas.
- * Organización de programas.

Como los programas pueden estar en lenguaje de alto nivel, en módulos objeto o preparados para la ejecución, se pueden clasificar las librerías en tres modalidades:

- * Librería fuente, contiene programas tal y como los crea el programador.
- * Librería objeto, contiene los módulos traducidos de los diversos programas.
- * Librería ejecutable, contiene programas en código máquina una vez ensamblados los módulos que lo componen.

Programas editores

Son los programas que permiten la escritura, grabación, corrección y puesta a punto por el programador de los programas de aplicación.

Son programas de propósito general, es decir, no van destinados a un lenguaje concreto, sino que con un editor cualquiera, el programador puede tratar el lenguaje que desee.

3.4. Programas de control

Son el conjunto de programas que facilitan y mejoran el rendimiento en el uso de los ordenadores.

Son los encargados de llevar a cabo la gestión de los recursos del sistema, facilitando el acceso ordenado y eficaz de los procesos a los recursos disponibles en el sistema en cada momento, impidiendo el interbloqueo de actividades.

Los programas de control suelen estar formados por rutinas o módulos cada uno de los cuales gestiona uno de los tres grupos fundamentales: hardware, software y trabajos.

El gestor del hardware se encarga de controlar y gestionar la CPU, la memoria y las rutinas de E/S.

La gestión de ficheros maneja los datos externos moviéndolos entre la memoria y los dispositivos periféricos.

La gestión de trabajos se encarga de asignar recursos a las diversas tareas y permite la comunicación con el exterior.

4. ESTRUCTURA DE LOS SISTEMAS OPERATIVOS

Un sistema operativo gestiona las operaciones de un ordenador y proporciona un gran número de servicios a los usuarios. En la medida en que se añaden más características a los sistemas operativos y en que el hardware se hace más complejo y versátil, el tamaño y la complejidad de los sistemas operativos ha ido creciendo.

Para gestionar la complejidad de los sistemas operativos y solucionar sus problemas, se ha prestado mucha atención a su estructura. El software debe ser modular. Esto ayuda a organizar su proceso de desarrollo y reduce las tareas de diagnóstico y detección de errores. Los módulos deben tener interfaces bien definidas entre sí y estas interfaces deben ser tan simples como sea posible.

Para grandes sistemas operativos, la programación modular por sí sola no es suficiente. En su lugar, ha ido creciendo el uso de conceptos como los de *niveles jerárquicos* y *abstracción de la información*. La **estructura jerárquica** de un sistema operativo moderno separa sus funciones de acuerdo a su complejidad, su escala característica de tiempo y su nivel de abstracción. Se puede contemplar al sistema como una serie de niveles. Cada nivel lleva a cabo un determinado subconjunto de funciones requeridas por el sistema operativo. Éste se basa en el nivel inferior para llevar a cabo funciones más primitivas y ocultar los detalles de dichas funciones. A su vez, cada nivel ofrece servicios al nivel superior.

En general, las capas más bajas trabajan con escalas de tiempo más cortas. Algunas partes del sistema operativo deben interactuar directamente con el hardware del ordenador, donde los sucesos pueden ocurrir en billonésimas de segundo. En el otro extremo, las partes del sistema operativo que se comunican con el usuario pueden trabajar en una escala de tiempo de unos pocos segundos. El uso de un conjunto de niveles se adapta bien a este entorno.

La forma en que se aplican estos principios varía enormemente entre los distintos sistemas operativos actuales. Sin embargo, es útil presentar un modelo de sistema operativo jerárquico propuesto por Denning y Brown, aunque no corresponda a ningún sistema operativo particular. Consta de los siguientes niveles:

Nivel 1: Consta de circuitos electrónicos. Los objetos que se tratan son registros, celdas de memoria y puertas lógicas. Se definen operaciones tales como borrar un registro o leer una posición de memoria.

Nivel 2: Es el conjunto de instrucciones del procesador. Las operaciones son las permitidas por el lenguaje de la máquina, tales como sumar, restar, cargar y depositar.

Nivel 3: Añade el concepto de procedimiento o subrutina, así como las operaciones de llamada y retorno.

Nivel 4: Introduce las interrupciones, las cuales hacen que el procesador salve el contexto actual e invoque a una rutina de tratamiento de la interrupción.

Estos primeros cuatro niveles no forman parte del sistema operativo, sino que constituyen el hardware del procesador. Sin embargo, comienzan a aparecer algunos de los elementos de los sistemas operativos, tales como las rutinas de tratamiento de interrupción. Es en el nivel 5 donde comienza a alcanzarse el sistema operativo propiamente dicho y en el que comienzan a aparecer los conceptos asociados con la multiprogramación.

Nivel 5: Se introduce la noción de **proceso** como un programa en ejecución. Entre los requisitos fundamentales de un S.O. que ofrezca soporte para múltiples procesos se incluye la capacidad de suspender y reanudar los procesos. Esto exige salvaguardar los registros del hardware, de modo que la ejecución pueda cambiar de un proceso a otro. Además, si los procesos necesitan cooperar, hace falta algún método de sincronización. Una de las técnicas más simples es el *semáforo*.

Nivel 6: Tiene que ver con los dispositivos de almacenamiento secundario del ordenador. Aquí tenemos las funciones de ubicación de las cabezas de lectura y escritura y se producen las transferencias reales de bloques.

Nivel 7: Crea un espacio de direcciones lógicas para los procesos. Organiza el espacio de direcciones virtuales en bloques que se pueden mover entre la memoria principal y la memoria secundaria. Tres son los esquemas más habituales: los que utilizan páginas de longitud fija, los que utilizan segmentos de longitud variable y los que utilizan ambos.

Hasta aquí, el sistema operativo se ocupa de los recursos de un solo procesador. A partir del nivel 8, el sistema operativo trata con objetos lógicos, objetos con nombre que pueden ser compartidos por varios procesos en un mismo ordenador o en varios ordenadores.

Nivel 8: Se dedica a la comunicación de información y mensajes entre los procesos. Una de las herramientas más potentes en este nivel es el “*tubo*” (pipe), que es un canal lógico para el flujo de datos entre los procesos. Un tubo se define con su salida en un proceso y su entrada en otro proceso. También se puede usar para enlazar dispositivos externos o ficheros con los procesos.

Nivel 9: Da soporte al almacenamiento a lo largo de los ficheros con nombre. Los datos del almacenamiento secundario se contemplan en términos de entidades abstractas de longitud variable.

Nivel 10: Es el que proporciona acceso a los dispositivos externos mediante interfaces estandarizadas.

Nivel 11: Es responsable de mantener la asociación entre los identificadores externos e internos de los recursos y objetos del sistema. El identificador externo es un nombre que puede ser empleado por una aplicación o un usuario. El identificador interno es una dirección que se utiliza en los niveles inferiores del sistema operativo para ubicar y controlar un objeto. Estas asociaciones se mantienen en un directorio.

Nivel 12: Proporciona servicios completos de soporte a los procesos. Se da soporte a toda la información necesaria para la gestión ordenada de los procesos. Esto incluye el espacio de direcciones virtuales del proceso, una lista de objetos y procesos con los que puede interactuar y las limitaciones de dicha interacción, los parámetros pasados al proceso en su creación y otras características del proceso que puedan ser utilizadas por el sistema operativo para su control.

Nivel 13: Ofrece al usuario una interfaz con el sistema operativo. Se denomina “*caparazón*” o *shell* porque separa al usuario de los detalles y le presenta el sistema como un simple conjunto de servicios. El *shell* acepta las órdenes del usuario o las sentencias de control de trabajos, las interpreta, crea y controla los procesos según sea necesario.

5. FUNCIONES DE LOS SISTEMAS OPERATIVOS

Un sistema operativo es un programa que controla la ejecución de los programas de aplicación y que actúa como interfaz entre el usuario de un ordenador y el hardware del mismo. Puede considerarse que un sistema operativo tiene tres objetivos o lleva a cabo tres funciones:

Comodidad: Hace que el ordenador sea más cómodo de utilizar.

Eficiencia: Permite que los recursos de un sistema informático se aprovechen de manera más eficiente.

Capacidad de evolución: Debe permitir el desarrollo efectivo, la verificación y la introducción de nuevas funciones en el sistema y, a la vez, no interferir en los servicios que brinda.

A continuación se tratan estos tres aspectos de los sistemas operativos.

El Sistema Operativo como Interfaz Usuario/Ordenador

El usuario final de las aplicaciones no tiene que ocuparse de la arquitectura del ordenador. Ve al sistema en términos de aplicaciones. Las aplicaciones pueden construirse con un lenguaje de programación y son desarrolladas por los programadores. Si se tuviera que desarrollar una aplicación como un conjunto de instrucciones máquina que sean del todo responsables del control del hardware, se tendría una tarea abrumadora y compleja. Para facilitarla, se ofrecen una serie de programas de sistemas. Algunos de estos programas se denominan utilidades e implementan funciones muy utilizadas que ayuda a la creación de los

programas, la gestión de los ficheros y el control de los dispositivos de E/S. El programa de sistemas más importante es el sistema operativo. Éste oculta al programador los detalles del hardware y le proporciona una interfaz cómoda para utilizar el sistema.

De forma resumida, un sistema operativo ofrece servicios en las áreas siguientes:

Creación de programas: Ofrece servicios como los editores y depuradores para ayudar al programador en la creación de programas. Normalmente, estos servicios están en forma de programas de utilidad que no forma realmente parte del sistema operativo, pero que son accesibles a través del mismo.

Ejecución de programas: El sistema operativo se encarga de cargar las instrucciones y los datos en memoria principal, de inicializar los ficheros y los dispositivos de E/S y de preparar otros recursos.

Acceso a los dispositivos de E/S: El sistema operativo se encarga de las instrucciones y señales de control.

Acceso controlado a los ficheros: Se encarga del formato de los ficheros y del medio de almacenamiento. También brinda los mecanismos de control de acceso para controlar ficheros compartidos por varios usuarios.

Detección y respuesta a errores: El sistema operativo debe dar una respuesta que elimine la condición de error con el menor impacto posible sobre las aplicaciones que están en ejecución. La respuesta puede ser desde terminar el programa que produjo el error, hasta reintentar la operación o, simplemente, informar del error a la aplicación.

Contabilidad: Debe recoger estadísticas de utilización de los diversos recursos y supervisar los parámetros de rendimiento tales como el tiempo de respuesta.

El sistema operativo como administrador de recursos

Un ordenador es un conjunto de recursos para el traslado, almacenamiento y proceso de datos y para el control de estas funciones. El sistema operativo es el responsable de la gestión de estos recursos.

El control del sistema operativo se ejerce de una manera especial debido a que:

- El sistema operativo funciona de la misma manera que el software normal de un ordenador, es decir, es un programa ejecutado por el procesador.
- El sistema operativo abandona con frecuencia el control y debe depender del procesador para recuperarlo.

El sistema operativo es, de hecho, nada más que un programa del ordenador. Como otros programas, da instrucciones al procesador. Dirige al procesador en el empleo de otros recursos del sistema y en el control del tiempo de ejecución de otros programas. Pero para que el procesador pueda realizar estas cosas, debe cesar la ejecución del programa del sistema operativo y ejecutar otros programas. Así pues, el sistema operativo cede el control al procesador para hacer algún trabajo “útil” y luego lo retoma el tiempo suficiente para preparar al procesador para que lleve a cabo la siguiente parte del trabajo.

Una parte del sistema operativo está en la memoria principal. En esta parte está el núcleo (*kernel*), que incluye las funciones utilizadas con más frecuencia en el sistema operativo y, en un momento dado, puede incluir otras partes de dicho sistema operativo que estén en uso. El resto de la memoria principal contiene datos y otros programas de usuario. La asignación de la memoria principal va a ser controlada conjuntamente por el sistema operativo y por el hardware de gestión de memoria en el procesador. El sistema operativo decide cuándo puede utilizarse un dispositivo de E/S por parte de un programa en ejecución y controla el acceso y la utilización de los ficheros. El procesador es, en sí mismo, un recurso y es el sistema operativo el que debe determinar cuánto tiempo del procesador debe dedicarse a la ejecución de un programa de usuario en particular. En el caso de sistemas multiprocesador, la decisión debe distribuirse entre todos los procesadores.

Facilidad de evolución de un sistema operativo

Un sistema operativo evolucionará en el tiempo debido a una serie de razones:

Actualizaciones del hardware y nuevos tipos de hardware.

Nuevos servicios, como respuesta a las demandas del usuario o a las necesidades de los administradores del sistema.

Correcciones, debido a los fallos que se descubren con el curso del tiempo.

6. TIPOS DE SISTEMAS OPERATIVOS

Los primeros ordenadores estaban compuestos exclusivamente por el sistema físico que se reducía a un computador de grandes dimensiones al que había que dotar de todas las instrucciones para que ejecutase un trabajo.

No existía pues un sistema operativo propiamente dicho. Los usuarios ponían su nombre y el tiempo que necesitaban utilizar la máquina en una lista de espera. Durante el tiempo de utilización, cada usuario disponía de todos los recursos y empleaba la ejecución paso a paso para poner a punto sus programas. Toda la memoria era utilizada por el usuario.

Este modo de trabajo era denominado de puerta abierta y era poco económico pues la inactividad de la CPU, podía alcanzar hasta el 90% del tiempo, esperando las decisiones del programador.

Además si se agotaba el tiempo asignado, se debía abandonar la unidad sin concluir el trabajo y, por el contrario, si se terminaba antes de lo previsto, el ordenador quedaba inactivo hasta la llegada del siguiente usuario.

Hoy en día existen dos posibilidades que potencian esa infrautilización antes comentada por parte de la CPU: multiprogramación y el tiempo compartido.

6.1. Multiprogramación

La posibilidad de la CPU de poder planificar los trabajos que están en el disco, proporciona una ventaja adicional.

Cuando un trabajo en ejecución ha de detenerse para efectuar una E/S, la CPU ya no permanece inactiva sino que toma otro trabajo y lo ejecuta, y si este también se detiene tomará otro o volverá sobre el primero.

La multiprogramación exige sistemas operativos mucho más avanzados ya que al existir varios programas en memoria se requiere la planificación de la CPU.

6.2. Tiempo compartido

Los sistemas en tiempo compartido consiguen un funcionamiento interactivo a bajo coste de la inactividad de la CPU, por tanto permite a varios usuarios compartir el uso simultáneo del ordenador mediante la planificación de la CPU y la multiprogramación.

Aunque la idea de tiempo compartido es de 1960 el avance tecnológico que requería hizo que dichos sistemas no se difundieran hasta los años setenta.

6.3. Sistemas más usuales

Como sistemas operativos que utilicen los conceptos actualmente descritos y por su gran generalización en el mercado merecen destacar el Unix y Windows NT.

6.3.1. Unix

Es un sistema operativo que trabaja en entorno multitarea de tiempo compartido con grandes requerimientos de memoria y espacio en disco duro para su ejecución.

Está escrito en C en lugar de ensamblador, permitiéndole una completa adaptación a diferentes máquinas.

Como otros sistemas, consta de dos partes: núcleo y programas de sistemas. El primero se encarga de la planificación de la CPU, gestión de memoria y sistema de ficheros y los segundos de la ejecución y manipulación de los ficheros.

La planificación de los procesos se hace mediante un algoritmo de prioridades de forma que a números mayores corresponde una menor prioridad y los procesos del sistema tienen una prioridad negativa.

En cuanto a la planificación de la memoria existen dos tendencias. En unas se utiliza el fichero de intercambio entre los diferentes procesos y en otras se utiliza la memoria virtual con paginación previa solicitud.

Los ficheros se organizan en directorios estructurados en árbol y se manejan mediante bloques de datos almacenados en disco.

El sistema de E/S maneja dispositivos en bloque (cintas), de caracteres (impresoras) y puntos finales de comunicación.

6.3.2. Windows NT

Windows NT es un sistema operativo de red de 32 bits reales que incluye funciones multitarea, multiproceso, portabilidad y soporte para multiproceso simétrico desarrollado por Microsoft.

Usa un sistema de ficheros propio conocido con el nombre de NTFS. Este sistema cambia la filosofía hasta entonces mantenida por Microsoft en cuanto a nombres de fichero y permite longitudes de hasta 256 caracteres.

Dispone de un seguimiento de transacciones, es decir, que si el sistema falla por alguna razón, al arrancar de nuevo, éste vuelve a poner los datos en el estado en que se encontraban antes de la caída.

Ha sido diseñado con una arquitectura modular y con la portabilidad en mente. Básicamente consta de un núcleo o Kernel y varios subsistemas diferentes. Hay subsistemas que pueden ejecutar programas bajo OS/2 y un DOS virtual que permite ejecutar MS-DOS y aplicaciones Windows de 16 bits. El Kernel es el encargado de realizar las operaciones básicas de NT y de asignar y sincronizar los distintos procesos, así como la gestión de interrupciones y errores.

El código de NT ha sido escrito casi en su totalidad en lenguaje C, de ahí su alto nivel de portabilidad.

Su capacidad de procesamiento simétrico, le permite en caso de existir varios procesadores, repartir la carga equitativamente entre ellos.

Windows NT incluye software de red puerto a puerto, de modo que los usuarios pueden compartir ficheros y aplicaciones con otros usuarios que dispongan de Windows, DOS u OS/2.

En cuanto al diseño de la memoria, utiliza un modelo de memoria plana en vez de paginada, lo que le permite a las aplicaciones con gráficos funcionar mucho más rápido.

7. BIBLIOGRAFÍA

Stallings, W. *Sistemas operativos* Prentice Hall, 2ed., 1997

Tanenbaum, Andrew S.
Sistemas operativos: Diseño e implementación
Prentice Hall, 1990