

TEMA 3: ELEMENTOS DE UN SISTEMA OPERATIVO INFORMÁTICO

3.1 LA GESTIÓN DE PROCESOS

Esta tarea es similar al trabajo que se realiza en una oficina. Para ello, se dispone de una lista de tareas a realizar y se les fija unas prioridades: alta, media, baja (por ejemplo). Se comenzará haciendo las tareas de prioridad alta, y, cuando se terminen, se seguirá con las de prioridad media y, después, las de baja. Una vez realizada la tarea se elimina de la lista. Esto puede traer un problema ya que las tareas de baja prioridad puede que nunca lleguen a ejecutarse y permanezcan en la lista para siempre. Para solucionar esto, se puede asignar alta prioridad a las tareas más antiguas.

Estas tareas se denominan procesos y son programas en ejecución que necesitan recursos para realizarlos: tiempo de CPU, memoria, archivos y dispositivos de E/S.

La gestión de procesos realiza las siguientes operaciones:

- Creación y eliminación de procesos
- Control del avance del proceso.
- Tratamiento de excepciones.
- Asignación de recursos: procesador, memoria, dispositivos hardware.
- Sistema de comunicación, por mensajes o señales, entre los procesos.

3.1.1 La planificación de procesos

Para compartir el procesador entre los diversos procesos activos residentes en memoria principal, el planificador del sistema operativo (scheduler) implementa una determinada política de planificación para realizar la asignación de la CPU a un solo proceso en cada momento.

El paso de un proceso inacabado al que determina el planificador se denomina **cambio de contexto** y consiste en guardar en memoria principal el estado completo (registros y todo tipo de datos intermedios) del proceso saliente, que pasará a ser suspendido, para poder reanudar posteriormente de ejecución y, a continuación, cargar en la CPU el estado del proceso entrante antes de comenzar su ejecución.

Se puede realizar una **planificación a largo plazo** cuando se trabaja con la cola de lotes y se selecciona el siguiente trabajo a ejecutar y una **planificación a corto plazo**, en la que se asigna finalmente el procesador a uno de los procesos preparados de acuerdo con la política de planificación elegida.

El programa que realmente se encarga de mandar el control de la CPU a un proceso de la cola de procesos preparados es el **despachador** (dispatcher). Esta asignación se realiza en varios pasos:

TEMA 3: ELEMENTOS DE UN SISTEMA OPERATIVO INFORMÁTICO

- Decide si se debe cambiar el proceso que está activo de acuerdo con la política de planificación y, si es así, continúa.
- Salva el entorno volátil del proceso actual.
- Toma el primer proceso de la cola de preparados o ejecutables.
- Carga el entorno volátil del proceso elegido (el contenido del PCB).
- Cede el control al proceso y lo pone en modo usuario.

La actuación del despachador se invoca por alguna de las siguientes razones:

- Un proceso finaliza.
- El sistema operativo decide que el proceso activo no debe seguir ejecutándose.
- El proceso agota su tiempo límite asignado (quantum).
- Un proceso cambia de estado de bloqueado a ejecutable.

3.1.2 Planificación de CPU-Multiprocesamiento

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

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

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

Desgraciadamente, todos estos objetivos no se pueden conseguir simultáneamente.

TEMA 3: ELEMENTOS DE UN SISTEMA OPERATIVO INFORMÁTICO

Existen dos formas básicas de realizar esta conmutación entre tareas:

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

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

3.1.3 Algoritmos de planificación

Los criterios para medir la eficiencia de un **algoritmo de planificación** son:

- **Uso del tiempo de la CPU** o fracción de tiempo que está ocupada.
- **Productividad** o cantidad de trabajo completado en la unidad de tiempo.
- **Tiempo de retorno de los procesos**, desde que se lanza el proceso hasta su finalización.
- **Tiempo de espera**, que es el tiempo que un proceso pierde por ser multiprogramado. Puede ser calculado restando el tiempo de retorno el tiempo de ejecución efectivo del proceso.
- **Tiempo de respuesta**, que va ligado a sistemas interactivos y representa el tiempo que pasa desde que se introduce un comando, o se lanza un programa, hasta que obtenemos la respuesta del sistema.

Además, la única forma de garantizar tiempos límite de ejecución para los sistemas de tiempo real consiste en utilizar un sistema de prioridades para la planificación.

Por otra parte, el planificador debe seguir principios de imparcialidad y, además, garantizar la predictibilidad y repetibilidad de la ejecución de los procesos de forma que los resultados sean los mismos que si se procesaran secuencialmente y sin interrupciones.

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

TEMA 3: ELEMENTOS DE UN SISTEMA OPERATIVO INFORMÁTICO

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

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

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

Uno de los algoritmos más sencillo, equitativo y de uso más amplio es el **Round Robin**, que hace una planificación por rondas. No se establecen prioridades. Cada proceso tiene asignado un tiempo de ejecución denominado **quantum**, o “rodaja” de tiempo. Cuando se cumple dicho tiempo y la tarea no ha concluido se guarda el estado del proceso en memoria y se da paso al siguiente proceso. El proceso saliente pasa al final de la lista de procesos en espera, siendo reanudada su ejecución cuando llegue su turno.

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

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

3.1.4 Sincronización de procesos

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

- Los **procesos independientes**: pueden ejecutarse a la vez, detenerse y reorganizarse sin ningún efecto negativo, son procesos deterministas y reproducibles. Por ejemplo, un programa que calcule números primos.
- Los **procesos cooperantes**: comparten su estado, su funcionamiento no es determinista (pueden darse situaciones inesperadas) y pueden ser irreproducibles. Por ejemplo, dos procesos que se ejecutan a la vez y comparten la pantalla, uno escribiendo “abc” y el otro “cba”. Según sea la sincronización de los procesos por el sistema operativo, en la pantalla podrá verse “abccba” o “acbbca” y otras combinaciones.

TEMA 3: ELEMENTOS DE UN SISTEMA OPERATIVO INFORMÁTICO

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

Por ejemplo, dos personas que viven juntas:

Tabla 3.1. Ejemplo práctico de procesos cooperantes

Persona A	Persona B
9:50 Llega a casa	10:10 Llega a casa
10:00 Mira la nevera. No hay leche.	10:20 Mira la nevera. No hay leche.
10:10 Va a la tienda	10:30 Va a la tienda
10:20 Llega a la tienda	10:40 Llega a la tienda
10:30 Deja la tienda	10:50 Deja la tienda
10:40 Llega a casa y guarda la leche	11:00 Llega a casa y, ¡¡ya hay leche!!

Este problema práctico puede extrapolarse a múltiples casos informáticos.

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

Lo que hay que conseguir es:

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

3.1.5 Bloqueos

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

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

TEMA 3: ELEMENTOS DE UN SISTEMA OPERATIVO INFORMÁTICO

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

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

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

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

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

3.2 LA GESTIÓN DE LA MEMORIA

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

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

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

TEMA 3: ELEMENTOS DE UN SISTEMA OPERATIVO INFORMÁTICO

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

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

Los criterios empleados para evaluar un gestor de memoria son:

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

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

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

3.2.1 Fragmentación

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

- **Fragmentación interna:** se debe a la diferencia de tamaño entre la partición de memoria y el objeto residente dentro de ella. Una posible solución a la fragmentación interna sería la asignación dinámica de memoria. En ella, los bloques no tienen un tamaño predefinido y el gestor de memoria asigna un espacio de memoria según las características del proceso a ejecutar.
- **Fragmentación externa:** se debe al desaprovechamiento de memoria entre particiones (particiones no contiguas).

A pesar de no presentar el problema de la fragmentación externa, las particiones fijas no son muy atractivas, sobre todo si se dispone de poca memoria, puesto que la mayor parte de ésta se desperdicia con programas menores que sus particiones.

TEMA 3: ELEMENTOS DE UN SISTEMA OPERATIVO INFORMÁTICO

En vez de esto, se utiliza un algoritmo de gestión de memoria basado en **particiones variables**.

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

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

3.2.2 Segmentación

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

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

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

3.2.3 Memoria virtual

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

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

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

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

TEMA 3: ELEMENTOS DE UN SISTEMA OPERATIVO INFORMÁTICO

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

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

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

3.2.4 Paginación

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

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

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

3.2.5 Algoritmos de reemplazo de páginas

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

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

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

- **FIFO** (First Input First Output). Es el más simple de ellos, tanto para entender como para programar. Consiste en elegir la página que lleve más tiempo en memoria, sin considerar cuándo se utilizó por última vez.

TEMA 3: ELEMENTOS DE UN SISTEMA OPERATIVO INFORMÁTICO

- **LIFU** (Usado Menos Recientemente). Este algoritmo consiste en considerar que un acceso a página reciente puede implicar un acceso futuro próximo. El algoritmo elige como víctima aquella página de entre las que se encuentran en memoria que hace más tiempo que no ha sido referenciada.
- **ÓPTIMO**. Es el que produce menor tasa de page fault. Es un algoritmo teórico, pues exige conocer las futuras referencias a página, eligiendo como víctima aquella que no volverá a ser referenciada nunca más o lo hará muy posteriormente. Sirve para estudios comparativos cuando se analizan nuevos algoritmos.

3.3 EL SISTEMA DE ENTRADA Y SALIDA

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

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

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

El **controlador del dispositivo** se encarga de mostrar al sistema operativo una interfaz estándar (dispositivo virtual) que hace transparentes al sistema operativo las peculiaridades del hardware que está por debajo.

3.3.1 Acceso directo a memoria

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

3.3.2 Manejadores de dispositivos

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

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

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

3.4 EL SISTEMA DE ARCHIVOS

Es el elemento del sistema operativo responsable de la administración de los archivos del almacenamiento secundario.

Un sistema de ficheros tiene las siguientes características:

- Debe permitir el acceso directo a la información contenida en los archivos, gestionando los múltiples bloques de memoria (sectores) que contienen los archivos.
- Debe permitir gestionar archivos de gran tamaño.
- La información contenida en los ficheros será permanente y se corresponderá con la última escritura. Además, el sistema de ficheros controlará el acceso concurrente a ellos y protegerá a los ficheros de accesos no autorizados (por ejemplo, Acceso de Solo Lectura, Acceso de Escritura, Acceso de Ejecución y/o varias combinaciones de estos).
- Los usuarios deben poder crear, modificar y borrar archivos.

La pérdida accidental de información es denominada **corrupción de datos** y puede ocurrir por fallos en el hardware de almacenamiento o por una mala gestión del sistema de ficheros. Las huellas dactilares pueden ocasionar errores de lectura en dispositivos magnéticos y, por ejemplo, las radiaciones alfa alteran bits en soportes magnéticos. Para evitar la corrupción de datos frente a estas contingencias, el sistema de archivos debe incluir en sus estructuras de ficheros redundancias para poder detectar y, si es posible, corregir errores. Otro problema es la destrucción maliciosa de información (virus, hackers, etc.).

TEMA 3: ELEMENTOS DE UN SISTEMA OPERATIVO INFORMÁTICO

El sistema de ficheros hace posible que el usuario no tenga que preocuparse por:

- Los dispositivos particulares.
- Dónde serán almacenados los datos.
- El formato de los datos en los dispositivos.
- Los medios físicos de la transferencia de datos hacia y desde los dispositivos.

Existen diferentes sistemas de archivos, es decir, se dispone de diferentes formas de organizar la información que se almacena en las memorias (normalmente, discos) de los ordenadores. Por ejemplo, los sistemas de archivos FAT, FAT32, NTFS, EXT4...

3.5 SISTEMA DE PROTECCIÓN

Se encarga de controlar el acceso de los procesos o usuarios a los recursos del sistema.

Mediante este control de acceso a los distintos recursos del sistema es posible implementar las medidas definidas por la organización, teniendo en cuenta las restricciones de acceso a las aplicaciones, a los datos guardados en el sistema informático, a los servicios ofrecidos (tanto internos como externos) y a otros recursos de tipo lógico del sistema.

La implantación del control de acceso en un sistema informático depende fundamentalmente de la gestión de cuentas de usuarios y de la gestión de permisos y privilegios.

Para ello, el modelo de seguridad que se aplica en los sistemas operativos se basa en la definición y gestión de determinados **objetos lógicos** (dispositivos lógicos, ficheros, servicios) y **sujetos** (usuarios y grupos, procesos, roles) a los que se conceden derechos y privilegios para realizar determinadas operaciones sobre los objetos.

Se pueden distinguir dos tipos de control de acceso:

- **Control de Acceso Obligatorio (MAC, Mandatory Access Control)** que son los permisos de acceso definidos por el sistema.
- **Control de Acceso Discrecional (DAC, Discretionary Access Control)** que son los permisos de acceso que configura el propietario de cada objeto.

TEMA 3: ELEMENTOS DE UN SISTEMA OPERATIVO INFORMÁTICO

Para definir la lista de sujetos que pueden acceder a cada objeto del sistema se utilizan las **Listas de Control de Acceso (ACL, Access Control List)**. De este modo, es posible contemplar restricciones de acceso no solo ya en función de la identidad del sujeto (usuario o proceso), sino también en función del horario y/o de la ubicación física del sujeto. Así mismo, en los sistemas gráficos se pueden establecer determinadas limitaciones en la interfaz de usuario de las aplicaciones, indicando qué menús, campos de información, botones u otros elementos gráficos puede visualizar cada usuario.

El principio de seguridad básico que se debería tener en cuenta es que “todo lo que no está expresamente permitido en el sistema debería estar prohibido”, asignando por defecto los mínimos privilegios y permisos necesarios a cada usuario del sistema, revisando de forma periódica los permisos de acceso a los recursos y registrando los cambios realizados en estos permisos de acceso.

También es recomendable controlar los intentos de acceso fraudulento a los datos, ficheros y aplicaciones del sistema informático y, cuando sea técnicamente posible, se debería guardar en un registro la fecha, hora, código y clave errónea que se han introducido, así como otros datos relevantes que ayuden a descubrir la autoría de esos intentos de acceso fraudulentos.

3.6 SISTEMA DE COMUNICACIONES

Para que los ordenadores puedan intercambiar información, es necesario que exista un sistema de comunicaciones que se encargue de transportar dicha información desde el equipo origen al destino.

Si se desea permitir la comunicación entre varios ordenadores, se necesita conectarlos a una red de transmisión de datos. Esta red está formada por tres partes fundamentales:

- **Terminales:** son los equipos que se comunican, como ordenadores, teléfonos, etc. Estos determinan la naturaleza de la información que va a tener que manejar la red (voz, datos, etc.).

TEMA 3: ELEMENTOS DE UN SISTEMA OPERATIVO INFORMÁTICO

- **Dispositivos de red:** se corresponden con el conjunto de elementos físicos que hacen posible la comunicación entre el equipo emisor y el receptor. Estos dispositivos son:
 - **Canal de comunicación:** es el medio por el que circula la información.
 - **Elementos de interconexión:** son los encargados de interconectar todos los terminales de la red y también trabajan para seleccionar el mejor camino por el que circulará la información (en caso de que exista más de un camino).
 - **Adaptadores de red:** son los encargados de convertir el formato de información de los terminales (normalmente en forma de señales eléctricas) en el formato utilizado por la red de comunicación (señales eléctricas, ondas de radio, etc.).
- **Programas de red:** son todos los programas que permiten controlar el funcionamiento de la red para hacerla más fiable.

3.7 SISTEMA DE INTERPRETACIÓN DE ÓRDENES

3.7.1 La línea de comandos

La línea de comandos es un método tradicional de interacción con el sistema, aunque hoy en día está sustituida por el entorno gráfico, ya que es un método mucho más intuitivo, requiere menos tiempo de aprendizaje y no necesita que el usuario memorice una serie de órdenes. Sin embargo, se sigue utilizando sobre todo por usuarios profesionales, ya que permite realizar cualquier operación sobre el sistema y además funciona en cualquier ordenador.

La línea de comandos tiene varias zonas diferenciadas que aparecen en la pantalla:

- **Prompt** (aviso de orden): es un carácter o conjunto de caracteres que se muestran en la pantalla del terminal para indicar al usuario que el equipo está a la espera de recibir una orden para ejecutarla. Puede ser configurado a gusto del usuario para incluir información útil y no aparece hasta que el sistema no ha terminado de ejecutar la orden introducida anteriormente. En los sistemas Linux el prompt suele incluir por defecto el símbolo \$ o #. En Windows, el símbolo >.
- **Intérprete de comandos:** es un programa que se encarga de traducir e interpretar la orden introducida para ser ejecutada. Entra en funcionamiento cuando el usuario termina de teclear una orden (normalmente lo indica utilizando la tecla [intro]).

TEMA 3: ELEMENTOS DE UN SISTEMA OPERATIVO INFORMÁTICO

- **Órdenes:** Corresponden a las órdenes o comandos que introduce el usuario para que sean ejecutadas por el sistema. En una línea de prompt se puede introducir una sola orden o más de una. Si se introduce más de una, el intérprete de órdenes reconoce una serie de caracteres o palabras especiales que indican en qué secuencia se van a ejecutar esas órdenes (una a continuación de la otra, todas a la vez, etc.) y si los resultados de unas se van a utilizar como datos de entrada para las otras.
- **Cursor:** es un carácter especial con forma de barra vertical, rectángulo o barra horizontal que destaca por su brillo o su parpadeo y se utiliza para indicar al usuario en qué parte de la orden introducida se encuentra. El cursor se desplaza a la derecha conforme el usuario escribe su orden, aunque también puede variar su posición si se utilizan las flechas del teclado (izquierda o derecha). Resulta muy útil cuando el usuario quiere corregir una orden introducida sin necesidad de tener que volver a teclearla por completo.
- **Resultados:** es la información que genera la ejecución de la orden en el equipo, que puede tener diferente formato (un simple mensaje, números, tablas, gráficos, etc.). Si la orden introducida no es reconocida correctamente por el intérprete o si se da algún problema durante la ejecución, se puede mostrar como resultado un mensaje de error indicando el problema y las posibles causas que lo han producido.

3.7.2 La interfaz gráfica

Se entiende por **interfaz gráfica** a todo aquella interfaz que utilice ventanas, iconos y ratón. Al comienzo de la informática, los ordenadores utilizaban solo el modo texto.

El desarrollo de los entornos gráficos y la facilidad de uso que ellos involucran han contribuido al desarrollo de la informática. En la actualidad, la mayoría de los sistemas operativos incorporan la visualización de entornos gráficos.

3.8 PROGRAMAS DEL SISTEMA

Son aplicaciones de utilidad que se suministran con el sistema operativo, pero no forman parte de él. Ofrecen un entorno útil para el desarrollo y ejecución de programas, siendo algunas de las tareas que realizan:

- Manipulación y modificación de archivos.
- Información del estado del sistema.
- Soporte a lenguajes de programación.
- Comunicaciones.