

TABLA DE CONTENIDOS

INTRODUCCIÓN	3
ADMINISTRACIÓN DE ENTRADA/SALIDA.....	4
PRINCIPIOS DE HARDWARE DE E/S.....	4
<i>Tipos de dispositivos</i>	<i>4</i>
PRINCIPIOS DE SOFTWARE DE ENTRADA/SALIDA	4
<i>Manejadores de interrupciones</i>	<i>5</i>
<i>Manejadores de dispositivos.....</i>	<i>5</i>
<i>Software Independiente del dispositivo.....</i>	<i>6</i>
<i>Software de nivel usuario</i>	<i>6</i>
DISPOSITIVOS DE ENTRADA/SALIDA DE USO MASIVO:.....	6
DISCOS.....	6
<i>Algoritmos de planificación de discos.....</i>	<i>8</i>
RELOJES.....	10
<i>Software para relojes</i>	<i>11</i>
TERMINALES.....	11
<i>Terminales seriales.....</i>	<i>11</i>
<i>Terminales mapeados a memoria.....</i>	<i>12</i>
SISTEMA DE ARCHIVOS	13
ARCHIVOS	14
<i>Algunos aspectos importantes respecto a los archivos.....</i>	<i>14</i>
DIRECTORIOS.....	17
<i>Algunos aspectos importantes de los directorios.....</i>	<i>17</i>
IMPLEMENTACIÓN DE ARCHIVOS	19
<i>Asignación adyacente</i>	<i>20</i>
<i>Asignación mediante listas enlazadas.....</i>	<i>20</i>
<i>Nodos-i.....</i>	<i>21</i>
IMPLEMENTACIÓN DE DIRECTORIOS.....	22
CONFIABILIDAD DEL SISTEMA DE ARCHIVOS	23
CONSISTENCIA DEL SISTEMA DE ARCHIVOS.....	24
SEGURIDAD Y PROTECCIÓN DEL SISTEMA DE ARCHIVOS.....	25
CONTROL DE ACCESO	25
<i>Contraseñas.....</i>	<i>26</i>
<i>Identificación física</i>	<i>26</i>
ATENTADOS CONTRA LA SEGURIDAD	26
<i>Virus.....</i>	<i>26</i>
<i>Gusanos</i>	<i>27</i>
MECANISMOS DE PROTECCIÓN.....	27
<i>Dominios de protección</i>	<i>27</i>
<i>Matriz de acceso</i>	<i>28</i>
<i>Listas de control de acceso (ACL, Access Control List).....</i>	<i>28</i>
<i>Listas de Capacidades</i>	<i>29</i>
BIBLIOGRAFÍA.....	30
RESUMEN.....	31
EVALUACIÓN DEL MÓDULO 4	32

INTRODUCCIÓN

En este módulo analizaremos dos aspectos importantes de todo Sistema operativo: el manejo de las Entradas y Salidas y la administración de Archivos.

El control de todos los dispositivos de Entrada/Salida, tales como por ejemplo impresoras, disco rígido, dispositivos de sonido, etc., es responsabilidad del Sistema Operativo. Para ello debe emitir comandos a los dispositivos, capturar interrupciones y manejar errores que puedan surgir. También debe proporcionar una interfaz entre los dispositivos y el resto del sistema que sea fácil de utilizar, pero a la vez ocultar la complejidad de la utilización de estos dispositivos de hardware.

Por otra parte la cara más visible de un Sistema Operativo es el sistema de archivos. La mayoría de los programas leen o escriben cuando menos un archivo y los usuarios siempre tienen conocimiento de la existencia de archivos y sus propiedades. Es más, para muchas personas la comodidad y utilidad del sistema operativo depende principalmente de la interfaz, estructura y confiabilidad del sistema de archivo.

A lo largo de la primera parte de este módulo se desarrollarán los conceptos esenciales para poder comprender la función de E/S, así como una explicación detallada de los distintos dispositivos intervinientes en cualquier sistema computacional.

Paso seguido, en la segunda parte estudiaremos en forma exhaustiva el sistema de administración de archivos, definiendo conceptos nuevos relacionados con este tópico y desarrollando las distintas maneras que posee el sistema para manejar e implementar los archivos y su protección.

ADMINISTRACIÓN DE ENTRADA/SALIDA

El sistema de E/S se puede dividir básicamente en dos componentes. Uno Hardware y otro Software. Ambos componentes tienen sus funciones bien definidas y dominios de acción.

Principios de Hardware de E/S

El hardware de E/S está determinado por el *dispositivo físico* propiamente dicho, el cual, por lo general, consta de componentes mecánicos y componentes electrónicos, denominado *controlador*, que es el encargado de actuar sobre el dispositivo físico de acuerdo a los requerimientos que sean solicitados de las capas superiores del sistema de E/S.

Existen diversos tipos de controladores de acuerdo al tipo de dispositivo.

Tipos de dispositivos

Por lo general los dispositivos se clasifican en dispositivos tipo *Bloque* y dispositivos tipo *Carácter*. Los de tipo bloque son los que definen un interfaz de comunicación mediante bloques de bits que se mueven en conjunto, de una sola vez, como los discos; y los dispositivos tipo carácter son los que definen una interfaz de comunicación serial mandando la información, por el cable por ejemplo, de a bits en forma consecutiva o serial, como los terminales e impresoras. Además existen otros dispositivos que no se ajustan a ninguno de estos tipos como los relojes.

Principios de software de Entrada/Salida

El software de E/S es el encargado de acceder y usar los distintos dispositivos existentes, para lo cual, en general, su estructura tiende a cumplir mínimamente las siguientes funciones:

- Manejo de errores.
- Software independiente de los dispositivos.
- Tratamiento y reconocimiento de dispositivos de uso exclusivo (por ejemplo, impresoras) y no exclusivo (por ejemplo, disco rígido).

Desde el punto de vista del software podemos decir que el sistema de E/S puede conformarse por diferentes capas donde cada una va atacando y solucionando un determinado problema, y entre todas logran el manejo eficiente de los dispositivos.

Así, desde mayor a menor nivel, tenemos las siguientes capas:

- Software de nivel usuario
- Software independiente del dispositivo
- Manejadores de dispositivos
- Manejadores de Interrupciones

Manejadores de interrupciones

Como se ha visto anteriormente, toda la gestión de distintos dispositivos se realiza a través de interrupciones y éstas naturalmente, a los ojos del usuario no son para nada entendibles. Esta capa es la que está en contacto con el dispositivo físico coordinando el uso de las interrupciones que lo manejan y en consecuencia realizando la interfaz necesaria para los distintos manejadores de dispositivos.

Manejadores de dispositivos

Existen diferentes tipos de manejadores de acuerdo a los tipos de dispositivos presentes en el sistema. Los manejadores de dispositivos son los encargados de comunicarse con el controlador de dispositivos para realizar las operaciones requeridas por las capas de nivel superior. Esta comunicación es realizada principalmente en base a interrupciones. El manejador de dispositivo entrega los comandos al controlador de dispositivos para que éste efectúe las acciones necesarias. Por ello un manejador de dispositivo tendrá todo el código necesario para poder entenderse con un tipo específico de dispositivo, o cuando mucho, con un conjunto de dispositivos íntimamente relacionados. Estos manejadores son también conocidos como Drivers y siempre son provistos por el fabricante del dispositivo como tal.

Por lo general las acciones que realiza el manejador de dispositivo son las siguientes.

- Atiende requerimientos de las capas superiores.
- Si el manejador está ocupado, arma una cola de requerimientos a ser atendidos.
- Si el manejador está libre, se comunica con el controlador para que efectúe el requerimiento.
- Entrega los resultados de los requerimientos solicitados a las capas superiores.

Software Independiente del dispositivo

El objetivo de esta capa es proveer una abstracción de la implementación del sistema de E/S de manera de hacer más fácil su uso para el usuario y ocultarle la complejidad del uso de interrupciones y comandos de bajo nivel. Esta capa es la encargada, por ejemplo, de asignarle las letras a las unidades de disco.

Entre las acciones que se contemplan en el software independiente se encuentran:

- Nombramiento de archivos.
- Tamaño del bloque.
- Mecanismos de Protección.
- Asignación y liberación de dispositivos de uso exclusivo y no exclusivo.

Software de nivel usuario

Esta capa contempla un conjunto de bibliotecas que implementan las llamadas a sistema y algunas otras operaciones que requieren manejar los programadores, entre las que se cuentan, por ejemplo, darle formato a un disco, manejar la cola de impresión, etc.

DISPOSITIVOS DE ENTRADA/SALIDA DE USO MASIVO:

Discos

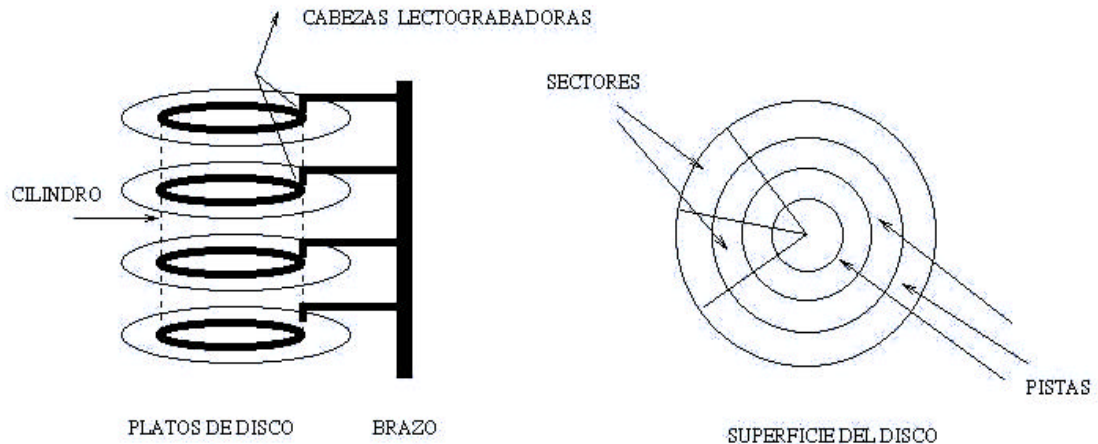
Los discos son dispositivos de almacenamiento secundario, que se utilizan para almacenar información. El almacenamiento en estos dispositivos tiene ciertas ventajas sobre el uso de memoria principal. En primer lugar el costo por bit es menor; la capacidad de almacenamiento es mucho mayor, y la información de tipo persistente, lo cual significa que no se pierde cuando se apaga la máquina (como ocurre con la memoria RAM).

Físicamente los discos son un conjunto de platos apilados. Cada plato está recubierto por una superficie magnetizable, la cual se divide en pistas concéntricas y éstas a su vez se vuelven a dividir en sectores. Estas pistas y sectores se numeran y permiten de esta forma hacer referencia a alguna parte determinada del plato en forma unívoca.

Para poder trabajar, todos los platos están montados sobre un mismo eje que los hace girar a alta velocidad y cada plato posee su cabezal de lectura – escritura que permite acceder a la información almacenada tanto en la cara de arriba como en la de abajo.

Además de las pistas y los sectores se maneja un concepto más que es el de cilindro. Un cilindro está compuesto por todas las pistas de los distintos platos que se encuentran a la misma distancia del eje.

Mayor información de la estructura física de los discos podrá obtener en los materiales de estudio de la materia **Arquitectura de Computadores II**.



Cuando se solicita al controlador de disco que extraiga o grave alguna información en el disco, éste tiene que desarrollar una serie de actividades hasta que localiza la posición correcta dentro del disco y lleva a cabo, entonces, la acción de lectura o escritura.

Estas acciones toman un determinado tiempo y éste toma diversas denominaciones:

- **Tiempo de búsqueda:** es el tiempo que se demora en colocar el cabezal sobre el cilindro indicado. Los cabezales se mueven todos juntos, y recorren la superficie del disco desde el eje hacia el extremo y viceversa. Dependiendo de dónde esté parado el brazo, éste tendrá que desplazarse hacia adentro o hacia fuera; se denomina a esta acción *tiempo de búsqueda*
- **Tiempo de latencia rotacional:** Es el tiempo que demora en pasar bajo el cabezal, el sector solicitado. Una vez que el cabezal está posicionado sobre el cilindro indicado, se debe esperar que pase por debajo del brazo el sector indicado; por ello, los discos rígidos giran constantemente y a una velocidad estable. El tiempo que se espera hasta que el sector llegue hasta el cabezal se denomina *Tiempo de latencia rotacional*.
- **Tiempo de transmisión:** es el tiempo que se tarda en realizar la escritura o lectura del sector solicitado. Cuando el cabezal está en posición y llegó el sector que se estaba esperando se puede realizar la lectura o escritura.

La sumatoria de todos estos tiempos influyen en la transferencia de información desde y hacia el disco.

Algoritmos de planificación de discos

A medida que van llegando las solicitudes de lectura o escritura en disco, éstas pueden ser que sean atendidas inmediatamente debido a que el disco en ese instante no estaba trabajando, o bien, puede ser que tengan que esperar hasta que el disco se desocupe y formar así una cola entre todas las solicitudes pendientes de ser atendidas.

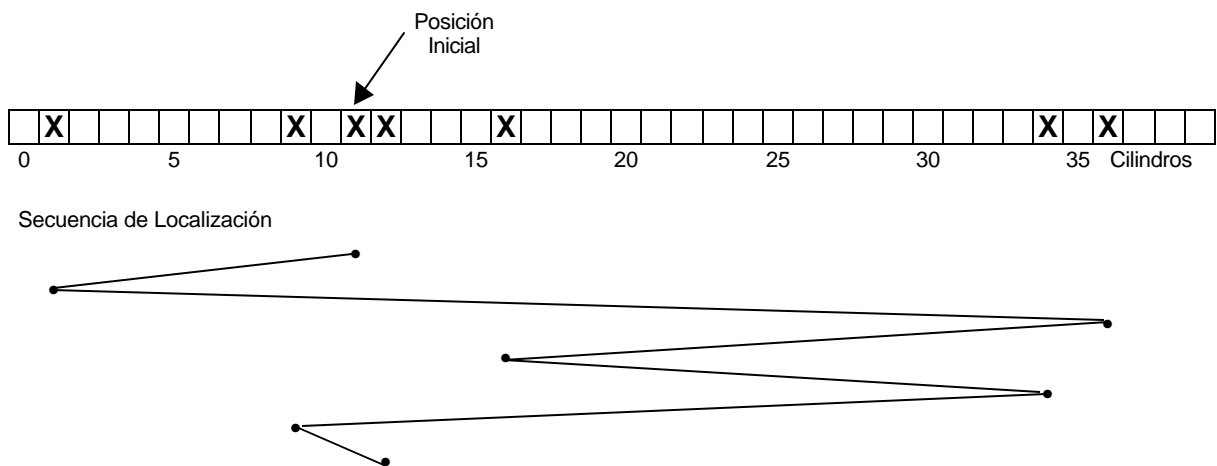
La forma en que se atiende esta cola de solicitudes a disco se conoce como planificación de disco, y el principal objetivo que busca es atender a todas las peticiones en el menor tiempo posible. Existen diversos algoritmos destinados a planificar los requerimientos de acceso al disco, así como se vio anteriormente para procesos y planificación de la memoria. Estos se explican, brevemente a continuación:

- FCFS. Los requerimientos son atendidos por orden de llegada. El primero que llega es el primero que es atendido y así sucesivamente. En este caso no se tiene en cuenta el sector a ser leído sino simplemente el orden de llegada.
- SSTF. (Primero el de menor tiempo de búsqueda). Se atienden requerimientos de acuerdo al que requiera menor tiempo de búsqueda. Este algoritmo optimiza el movimiento del brazo ya que para atender a las solicitudes, lo que realiza es una búsqueda dentro de la cola para encontrar, de todas las solicitudes, cuál es la que pretende acceder al sector más cercano a su posición actual. De esta manera se desplaza lo menos posible el brazo y se gana tiempo de búsqueda. Aquí no se tiene en cuenta el orden de llegada sino el sector que se quiere leer en relación con la posición actual del cabezal.
- SCAN. El brazo del disco se mueve hacia dentro y hacia afuera atendiendo todos los requerimientos que encuentre a su paso. Constantemente el cabezal se desplaza de adentro hacia afuera atendiendo a las solicitudes que se encuentra en su camino. En este caso no se tiene en cuenta ni el orden de llegada ni su posición relativa con respecto al cabezal, simplemente hay que esperar que el cabezal pase por el sector que deseamos utilizar y es entonces cuando la solicitud será atendida.
- C-SCAN. Igual al SCAN, sólo que el brazo no cambia de dirección, sino que siempre se mueve hacia adentro. Una vez que no tiene más requerimientos internos, el brazo avanza hasta el cilindro más interno y se mueve al cilindro más externo y repite su movimiento hacia adentro. Las peticiones que llegan después de haber comenzado el barrido, se atienden en el siguiente barrido.
- LOOK. Posee la misma filosofía que el SCAN, es decir, va recorriendo las distintas pistas y sectores de adentro hacia afuera y viceversa, y en su paso atiende las solicitudes. Pero realiza una mejora en el sentido de que este algoritmo no necesariamente recorre desde el primer cilindro hasta el último, sino que va observando si es que en la cola hay más solicitudes en el sentido en que avanza. De ser así continúa; caso contrario inmediatamente cambia el sentido de recorrida aplicando el mismo criterio. Se gana tiempo debido a que el SCAN, para cambiar el sentido de recorrida, indefectiblemente debe llegar hasta el final del disco, por más que no haya ninguna solicitud en esa zona para ser atendida.

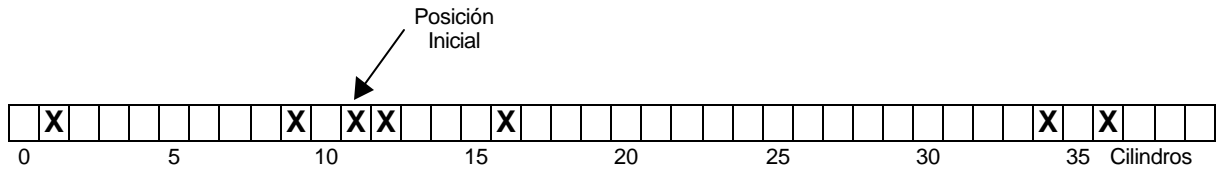
- C-LOOK. Igual al LOOK, sólo que el brazo no cambia de dirección, sino que siempre se mueve hacia adentro y una vez que no tiene más requerimientos internos, sin necesidad de llegar hasta el cilindro más interno, el brazo se mueve al cilindro más externo y repite su movimiento hacia adentro. Las peticiones que llegan después de haber comenzado el barrido se atienden en el siguiente barrido.

Veámoslo con un ejemplo: Considérese un disco con 40 cilindros. Una solicitud llega para leer un bloque en el cilindro 11, mientras procesa la localización del cilindro 11, llegan nuevas solicitudes para los cilindros 1; 36; 16; 34; 9 y 12, en ese orden. Estas se anexan en la cola de solicitudes pendientes. Cuando termine de atender la solicitud corriente (cilindro 11) el manejador del disco tiene una elección de cuál solicitud manejar después.

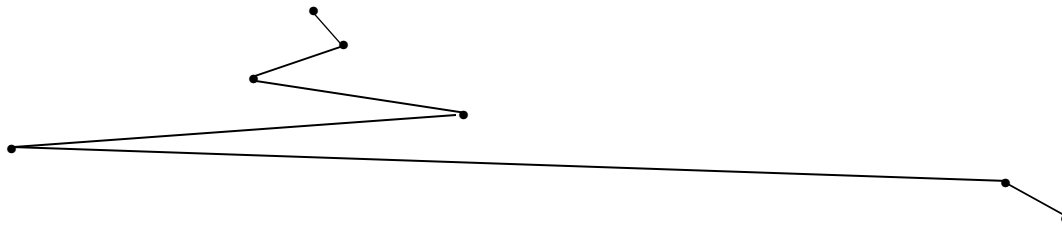
Si utilizara FCFS se dirigiría después al cilindro 1, luego al 36 y así sucesivamente. Este algoritmo requeriría movimientos del brazo de 10 cilindros (del 11 al 1) 35, (del 1 al 36) 20; 18; 25 y 3 respectivamente según el orden de las solicitudes haciendo un total de 111 cilindros desplazados hasta atender a todos, como se observa en la figura siguiente



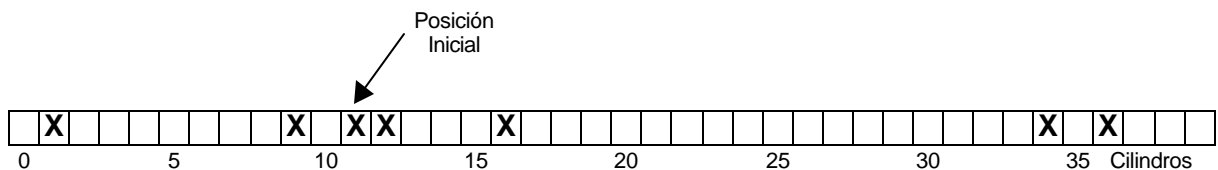
Manejando la mismas solicitudes, pero con un algoritmo SSTF, la secuencia de atención cambia, reduciendo la cantidad de cilindros recorridos a 61 tal como lo muestra la siguiente figura.



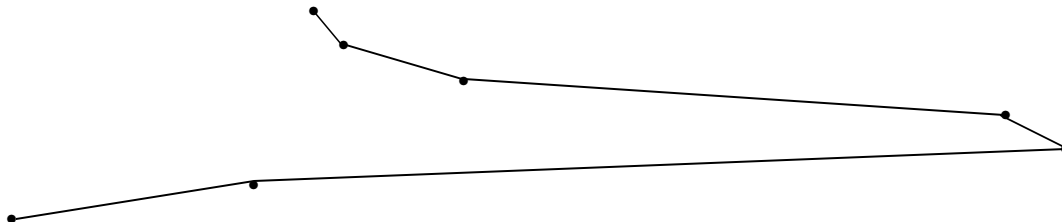
Secuencia de Localización



Por otra parte, tratando la serie de solicitudes con un algoritmo SCAN la cantidad de cilindros recorridos (60) y la secuencia en que se atienden, variaría. Para representar este algoritmo hay que tener en cuenta cuál es el movimiento inicial del cabezal, ya que como mencionamos anteriormente, el brazo está constantemente recorriendo desde una punta a la otra. En la siguiente figura se muestra la atención mediante SCAN y se consideró el movimiento inicial del cabezal hacia fuera (cilindro 40)



Secuencia de Localización



Relojes

Como ya se citó, los relojes son dispositivos que no pertenecen a la categoría de dispositivos de bloques, ni a la de dispositivos de tipo carácter debido a la naturaleza de la información que proporciona. Son esenciales para la operación de cualquier sistema de

tiempo compartido por una diversidad de razones. Estos conservan la hora del día y evitan que un proceso monopolice la CPU, entre otras cosas.

En general, el hardware de reloj puede estar basado en dos tipos distintos, y ambos a su vez son totalmente diferentes de los que usa la gente:

- relojes que se alimentan con la red eléctrica y generan interrupciones periódicas.
- relojes basados en cristal de cuarzo. Éstos tienen la característica de que son programables.

Software para relojes

La función del hardware del reloj es generar interrupciones en intervalos conocidos y regulares. Todo lo demás en que intervenga el tiempo debe ser realizado por el software, conocido como Manejador de Reloj.

El manejador de reloj tiene diferentes acciones y las realiza de acuerdo a las interrupciones que genera el reloj. En base a estas interrupciones trabaja el manejador de reloj. Entre las funciones que controla este manejador se encuentran:

- Control de tiempo de ejecución por proceso.
- Mantención de la hora del día.
- Realización del monitoreo del sistema.
- Manejo de alarmas.

Terminales

La idea de una terminal es la de una estación de trabajo: que es un puesto donde el operador puede generar entradas y salidas, pero el procesamiento es realizado por la máquina a la cual la terminal está conectada.

Desde el punto de vista del sistema operativo, las terminales se pueden dividir en dos amplias categorías con base en la forma en que el sistema operativo se comunica con ellas:

- Seriales bajo la norma RS-232.
- Mapeados a memoria

Terminales seriales

Las terminales seriales o RS232 son dispositivos que contienen un teclado y una pantalla que se comunican mediante el uso de un cable o manguera. Ésta posee en las puntas un conector o terminal DB-25 (25 pines o patitas), de las cuales una se utiliza para

transmitir los datos, otra para recibirlos y una tercera para tierra. Las otras 22 patitas son para funciones de control que normalmente no se utilizan.

Se llaman seriales debido a que la transmisión de datos desde y hacia la terminal se realiza de bit a bit, uno por vez y en secuencia, en serie, a una velocidad de entre 300 y 9600 bps. (bits por segundo)

La norma RS232 o V.24 es la que establece todos los parámetros de cómo es que se debe realizar esta transmisión para que funcione correctamente.

En cada lado, tanto de la computadora principal como de la terminal, se instala un controlador serial que posee un chip denominado UART (Transmisor–Receptor Asíncrono Universal), que es el encargado de realizar la transmisión y recepción de bits. Existen varios tipos de terminales seriales entre los que se encuentran:

- **Terminales tontos** que únicamente transmiten y reciben información sin la capacidad de generar ningún cómputo con ello, dejando todo el trabajo de cálculo librado a la computadora principal.
- **Terminales inteligentes** que además de realizar las actividades de transmisión y recepción de información, propias de una terminal, poseen capacidad de cálculo y pueden trabajar con los datos recibidos.

Terminales mapeados a memoria

Este tipo de terminales no utilizan como medio de comunicación con el computador una interfaz serial, sino una memoria RAM que comúnmente se denomina memoria de video. En este caso el controlador de terminal posee un chip denominado controlador de video que permite manejar el despliegue de caracteres y símbolos en el monitor. Un ejemplo de este tipo de terminales es lo que ocurre en el PC-IBM, donde a partir de una determinada dirección de memoria, se gestiona las imágenes que son presentadas en el monitor quedando así integrado todo en el mismo equipo central.

En los terminales mapeados a memoria, lo que en realidad está mapeado en memoria, es el monitor. El teclado se trata en forma independiente y por lo general, se comunica mediante una interfaz paralela.

SISTEMA DE ARCHIVOS

La utilización de memoria para almacenar información tiene ciertos inconvenientes. En primer lugar, **la información que se almacena en la memoria no persiste en el tiempo**, pues cuando el computador se apaga, la información se pierde. Por otro lado, si se desea almacenar una gran cantidad de información, no es posible hacerlo en memoria principal, pues por lo general los computadores no mantienen grandes cantidades, debido a que es un recurso de alto costo.

El sistema de archivos es una de las partes más visibles del sistema operativo. Los usuarios utilizan los archivos para almacenar sus datos o programas en forma persistente. Esto quiere decir que la información no se pierde una vez que el computador es apagado. Para almacenar dicha información se utilizan dispositivos de memoria secundaria, como discos, cintas, etc.

El sistema de archivos como parte del sistema operativo proporciona al usuario una interfaz sencilla, amigable y organizada, que le permite almacenar y organizar su información. Dicha interfaz el sistema la proporciona principalmente mediante archivos y directorios con operaciones definidas sobre ellos. Así también una parte importante de las operaciones que el sistema otorga a los usuarios, tiene relación con los mecanismos de protección que les permite proteger su información. Los mecanismos de protección son especialmente importantes para sistemas multiusuarios, por razones obvias.

Por otro lado, el sistema de archivos debe representar la información a almacenar de alguna manera en el sistema para que pueda interactuar fácilmente con la Unidad de Entrada / Salida, pues ésta debe almacenar realmente la información de archivos y directorios en los dispositivos de almacenamiento secundario. Es decir, más allá de la representación que se haga con los archivos y directorios, para que el usuario pueda entenderlos y manejarlos, internamente éstos deben ser almacenados de alguna manera en forma real en el medio de almacenamiento y esta tarea es también responsabilidad del sistema.

En esta parte del material analizaremos los dos puntos de vista : La visión Arriba–Abajo, es decir mediante la interfaz que proporciona al usuario y una Visión Abajo–Arriba, lo cual equivale a analizar cómo el SO administra dicha información para que la unidad de entrada / salida actúe sobre los dispositivos de almacenamiento secundario. Finalmente veremos los aspectos importantes que tienen que ver con la seguridad y la protección de los sistemas.

Archivos

El sistema de archivos proporciona una interfaz de operación hacia los usuarios principalmente mediante dos componentes: Archivos y Directorios. Estos dos componentes le permiten al usuario almacenar y organizar su información.

Algunos aspectos importantes respecto a los archivos

Concepto

Un archivo conceptualmente corresponde a un conjunto de información que se agrupa con el fin de ser utilizada posteriormente. En general, los archivos pueden ser programas, datos, imágenes, sonido, etc. En general, un archivo es una secuencia de bits, bytes o registros, cuya información es entendida por la entidad que crea el archivo (puede ser una aplicación) y por el usuario que lo define.

Es decir, este mismo documento que está leyendo, está hecho con un procesador de texto. Este documento es un archivo, pero si lo miramos desde una visión más abstracta, es una secuencia de caracteres, de bytes que sin el programa que lo generó (el procesador de texto) no tienen ningún sentido.

Por lo general los archivos son almacenados en disco. Cada archivo puede estar representado por uno o más bloques en memoria secundaria. Un bloque es la unidad de información que utilizan los archivos, de manera similar a la utilización de páginas de un proceso. Un bloque en general puede estar representado por uno o dos sectores del disco. (ver gráfico descriptivo de Disco).

Nombre de archivos

Como los archivos corresponden a un objeto que tiene información almacenada en disco, su identificación se realiza mediante un nombre específico para cada uno de ellos.

Las normas que rigen el nombramiento de los archivos son dependientes de cada sistema. Así, por ejemplo, en MS-DOS no se hace diferencia alguna en el uso de minúsculas y mayúsculas, en cambio Unix sí hace diferencia; MS-DOS tiene una estructura para el nombre de cada archivo: todos tienen un nombre que puede ser de un máximo de 8 caracteres y una extensión de 3 caracteres; en cambio en Unix el largo puede ser de 255 caracteres, y los nombres de los archivos no requieren extensión. Windows 95/98, por su parte, mantiene la estructura establecida por MS-DOS pero no limita el largo del nombre a 8 caracteres.

La extensión del archivo, hace referencia a la utilidad que se le va a dar al mismo. Así por ejemplo ".exe" hace alusión a un archivo ejecutable; ".doc" a un documento; ".dat" que se trata de un archivo de datos, etc.

Estructura de archivos

Por lo general la estructura de un archivo puede ser de tres tipos; utilizando como componente lógico un byte, o bien registros y sino claves.

En el primer caso, un archivo es visto por el sistema como un conjunto de bytes. Esto otorga flexibilidad y se puede escribir en ellos cualquier cosa.

La segunda estructura contempla un registro como componente lógico. Un registro debemos entenderlo como un conjunto de datos con estructura. Por ejemplo en un archivo de personal, un registro estaría compuesto por los datos de una persona (Apellido, nombre, edad y domicilio en ese orden). En este caso, cada entidad del archivo es un registro, de manera que la inserción y consulta de datos de él, debe regirse por la estructura definida para el registro. Pueden leerse o escribirse registros arbitrarios, pero no puede insertarse o eliminarse registros a la mitad de un archivo. Este tipo de archivos no es tan flexible como el anterior pero hace más fácil su tratamiento.

En el tercer caso, cuando se utilizan claves, también se tiene como componente lógico el registro, sólo que se organiza mediante bloques de registros, donde cada bloque lleva un índice o clave que hace más rápida la búsqueda.

Tipos de Archivos

Existen varios tipos de archivos, los cuales se clasifican respecto a la información que ellos almacenan. Entre los distintos tipos de archivos se encuentran los siguientes:

- *Archivos regulares*: Corresponden a archivos que contienen información de los usuarios. Estos archivos pueden ser ASCII o binarios. Los ASCII se denominan así porque su contenido se interpreta según la tabla ASCII (vista al finalizar el Unidad 3 de Programación Lógica) y pueden ser visualizados con cualquier editor de texto y son imprimibles. Los binarios no son imprimibles directamente y por lo general tienen un formato asociado; los archivos ejecutables, por ejemplo, son binarios.
- *Archivos directorios*: son archivos de sistema que se usan para mantener información respecto a la organización de archivos dentro del sistema. Este concepto se aclarará más cuando desarrollemos el tema Directorios, unas secciones más adelante.
- *Archivos especiales de caracteres*: Se usan para modelar dispositivos de E/S de tipo carácter, por ejemplo: terminales, teclados e impresoras. En algunos Sistemas Operativos como Unix, se utiliza este tipo de archivos para representar un dispositivo. Por ejemplo, se tiene un archivo de nombre TTY que representa una terminal y entonces para escribir algo en la misma, se utilizan las mismas llamadas al sistemas que las utilizadas para escribir en un archivo. Esta es una forma de simplificar el manejo de dispositivos.
- *Archivos especiales de bloque*: Se usan para modelar dispositivos de E/S que operan mediante bloques, como por ejemplo, los discos. Es la misma idea que el caso anterior, pero esta vez aplicado a dispositivos de bloque.

Accesos de archivos

El acceso de los archivos hace referencia a la forma de leer la información que contiene el archivo. Para ello imaginemos al archivo como un libro; si nuestra intención es leer la hoja número 20, tenemos dos caminos posibles: comenzar a hojear el libro desde el principio hasta llegar a la hoja 20 o bien abrir el libro exactamente en la hoja deseada.

Análogamente, en el acceso a la información contenida en un archivo tenemos esas dos posibilidades:

- **Acceso Secuencial:** En este caso, la información almacenada en un archivo se obtiene en orden, comenzando de lo primero a lo último. Si se desea algo almacenado en un lugar intermedio es necesario recorrer lo anterior para llegar al punto deseado. Este mecanismo es de fácil implementación pero no ofrece mucha flexibilidad y velocidad a la hora de localizar la información
- **Acceso Aleatorio o Directo.** En este caso, la información se puede obtener en forma directa, sin necesidad de leer lo que se encuentra antes. Un ejemplo de esto son las bases de datos. Normalmente este tipo de archivo requiere de información adicional para permitir localizar la información en forma directa, lo que hace que sea un poco más compleja su implementación, pero la facilidad que brinda lo hace mucho más práctico.

Atributos de archivos

El sistema de archivos aparte de permitir el nombramiento de archivos y su almacenamiento, mantienen información adicional de ellos. Como los archivos representan objetos, éstos siempre poseen características distintivas.

Así como en la vida real las cosas tienen sus dimensiones, olor, sabor, etc., los archivos poseen atributos que los diferencian entre ellos. De éstos, ya mencionamos el más importante: su nombre, pero además de él, el sistema guarda información, como por ejemplo el nombre del dueño del archivo, la fecha y hora en que se creó, cómo está protegido, tamaño actual, etc. Toda esta información adicional es la que se denomina *atributos de un archivo*.

Operaciones de archivos

El sistema de archivos proporciona un conjunto de operaciones a los usuarios que les permite almacenar información en archivos para luego recuperarla. Dichas operaciones son proporcionadas por el sistema mediante llamadas a sistema. Dentro de las operaciones más comunes presentes en los sistemas de archivos se tienen:

- | | |
|------------|------------------------|
| • Crear | • Agregar información |
| • Borrar | • Obtener atributos |
| • Abrir | • Establecer atributos |
| • Leer | • Renombrar |
| • Escribir | • etc. |

Directorios

Como se mencionó anteriormente, el sistema de archivo también proporciona mecanismos mediante los cuales los usuarios pueden organizar sus archivos. Los directorios son la herramienta que provee el sistema para realizar esta tarea. Así, armando una adecuada estructura de directorios, es posible organizar los archivos y así accederlos con mayor facilidad.

Algunos aspectos importantes de los directorios

Concepto

En esencia, un directorio es una tabla de símbolos en donde el sistema de archivos encuentra los nombres simbólicos de los archivos con los atributos asociados a cada uno de ellos. Esta tabla de símbolos se representa en un archivo especial denominado **archivo directorio**.

Imaginemos a los archivos como hojas sueltas, todas desparramadas, entonces, los directorios son como carpetas que nos sirven para agrupar la hojas que tienen relación entre sí. Por ejemplo, armar una carpeta con las evaluaciones; otra con las boletas a pagar, etc.. Así se van creando carpetas por voluntad del usuario, para organizar sus archivos según su criterio.

Subiendo un nivel más arriba en nuestra analogía, podríamos tener ahora todas las carpetas desparramadas, es entonces donde la presencia de un fichero o armario se hace imprescindible. Es el disco el que oficia de fichero o armario, conteniendo todas las carpetas y éstas conteniendo a las hojas y quizás otras carpetas.

Por esa razón, los directorios pueden llegar a tener otros directorios dentro de sí, denominados subdirectorios, razón por la que los sistemas operativos instrumentan los sistemas de archivos como una organización de tipo jerárquica. Que sea jerárquica significa que en un directorio se pueden encontrar archivos y directorios; y dentro de estos directorios, otros archivos y directorios, y así sucesivamente.

Anteriormente, cuando hablábamos de los tipos de archivos decíamos que existen algunos que son de tipo directorio. Estos no son más que directorios propiamente dichos, son manejados como archivo, ya que desde el punto de vista jerárquico un archivo directorio contiene los nombres de los archivos y archivos directorios que allí se encuentran con sus respectivos atributos.

Básicamente existen dos maneras de almacenar la información en un directorio. La primera consiste en que, en el mismo archivo directorio se encuentran tanto el nombre como los atributos de cada uno de los archivos y/o directorios que lo componen. La otra manera es tener el nombre del archivo o directorio y un puntero a otro espacio en el disco donde se encuentran sus respectivos atributos.

En casi todos los sistemas se encuentra un directorio llamado *raíz*, que corresponde al padre de todos los directorios.

Acceso a directorios

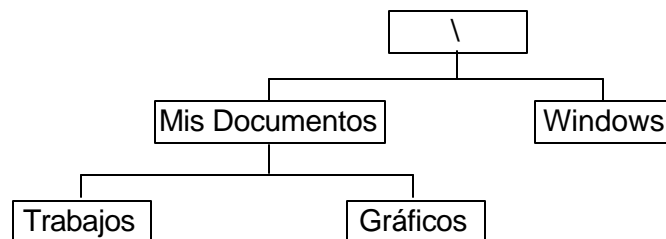
En el esquema planteado hasta el momento estamos manteniendo un excelente orden entre todas nuestros archivos, cada uno con sus atributos, en sus carpetas; éstas a su vez en otras carpetas hasta que todo queda muy bien guardado. Pero es imprescindible que todo este sistema de almacenamiento proporcione una forma de acceder a la información, una manera de poder llegar hasta el archivo y poder utilizarlo, sino todo el sistema fracasaría.

El acceso a los directorios se refiere a los mecanismos que proporcionan los sistemas para el acceso de los usuarios a los directorios y a archivos definidos en el sistema.

Básicamente se utilizan dos tipos de acceso:

- *Ruta de acceso absoluta*
- *Ruta de acceso relativa.*

Para comprender esto veamos el siguiente gráfico que representa una estructura de directorio típica:



El primer directorio representado con "\" es el denominado Directorio Raíz y del cual penden todos los demás. Por debajo en el primer nivel, tenemos además de archivos que no están representados, dos directorios, Mis documentos y Windows donde cada uno puede tener sus propios archivos y así sucesivamente según lo ya estudiado.

La ruta de acceso absoluta consiste en acceder a un directorio o archivo desde el directorio raíz; ejemplo: para un archivo que se llama MiFoto.bmp almacenado bajo el directorio Gráficos de la estructura anterior sería: \Mis Documentos\Gráficos\MiFoto.bmp. La primera "\" representa el Directorio raíz y las demás que aparecen después se utilizan como separadores para delimitar cada directorio de la estructura hasta llegar al archivo o directorio en cuestión.

En cambio, una ruta de acceso relativa consiste en acceder a un archivo o directorio considerando la posición actual. En este caso se maneja el concepto de directorio de trabajo; éste corresponde al directorio desde el cual se hace el acceso. En este caso, por ejemplo, si se desea hacer el acceso desde el directorio \ Mis Documentos basta con indicar Gráficos\MiFoto.bmp ya que parte de la ruta ya la tenemos recorrida.

De esto ejemplos y prácticas tendremos más en el desarrollo del módulo siguiente donde lo veremos aplicado a un Sistema Operativo real.

En muchos sistemas el directorio de trabajo se identifica con un ".", y también se conoce el directorio anterior y se identifica con "..".

Operaciones sobre directorios

Al igual que para el manejo de archivos, para el manejo de directorios el sistema también proporciona un conjunto de operaciones, mediante llamadas a sistema. Estas operaciones incluyen:

- Creación
- Eliminación
- Apertura
- Cierre
- Lectura
- Cambiar nombre
- Etc.

Implementación de Archivos

La segunda perspectiva que adoptamos para el análisis del sistema de archivos es la perspectiva de abajo hacia arriba, es decir cómo el sistema implementa todo lo antes expuesto y lo administra. Dentro de estas actividades veremos primero, cómo se trabaja con los archivos y luego con los directorios.

En general, los archivos se implementan en base a uno o más bloques que tienen su correspondiente ubicación en memoria secundaria.

Como sabemos, todo el espacio disponible de la memoria secundaria (el Disco) se organiza en bloques. Estos son todos del mismo tamaño y están numerados para poder identificarlos. Un archivo va estar almacenado en 1 o más bloques según su tamaño y un bloque le pertenece a un solo archivo, es decir, los bloques no se comparten.

De acuerdo a ésto se tienen tres implementaciones para los archivos. En general, la implementación dependerá de cada sistema en particular.

- Asignación adyacente
- Asignación mediante listas ligadas

- Nodos—i

Asignación adyacente

Este esquema consiste en asignar a cada archivo tantos bloques consecutivos como sean necesarios. Este esquema proporciona la ventaja de que es sencillo de implementar, pues para localizar un archivo sólo se necesita la dirección del primer bloque que utiliza y como siempre se almacena en bloques contiguos, con seguridad el resto del archivo estará a partir del primero en adelante.

La otra ventaja que proporciona es que es rápido pues todos los bloques están juntos.

Pero este esquema es muy rígido y tiene sus desventajas. Debido a que se almacena indefectiblemente en bloques contiguos, se necesita saber a priori el tamaño que tendrá el archivo para reservar bloques antes de almacenar otro al lado y que no pueda expandirse más tarde. Como habrá observado esta estimación por lo general no es posible.

La otra desventaja que caracteriza a este mecanismo es la fragmentación externa, pues cuando el espacio utilizado por el archivo es liberado queda un conjunto de bloques vacíos pero es difícil encontrar otro archivo que requiera la misma cantidad de disco para que lo pueda utilizar; por lo tanto, a medida que se va utilizando el sistema de archivo van quedando estas islas de bloques que no pertenecen a nadie y es difícil de utilizar llamadas Fragmentación externa.

Asignación mediante listas enlazadas

Para solucionar las desventajas de la asignación contigua, la primera idea que surgió fue utilizar los bloques donde haya espacio libre, sin importar si está contiguo o no. Este esquema consiste en tener una lista enlazada de todos los bloques que pertenecen al archivo. De manera que cada bloque (nodo de la lista) contiene la información perteneciente al archivo, más un puntero al siguiente bloque del archivo, y así sucesivamente.

Este esquema elimina el problema del almacenamiento contiguo y la fragmentación externa manteniendo la ventaja de saber la dirección del primer bloque.

Pero como nada es perfecto, este modelo también tiene sus desventajas. La principal es que el acceso es muy lento ya que los bloques no necesariamente están de seguido, sino que muy probablemente estén distribuidos en el disco, por lo cual para recorrer el archivo, el sistema tiene que navegar mucho por la superficie del disco; y la otra desventaja es que el espacio del bloque no puede ser utilizado íntegramente para el archivo sino que se tiene que dejar espacio para los punteros a bloque siguiente.

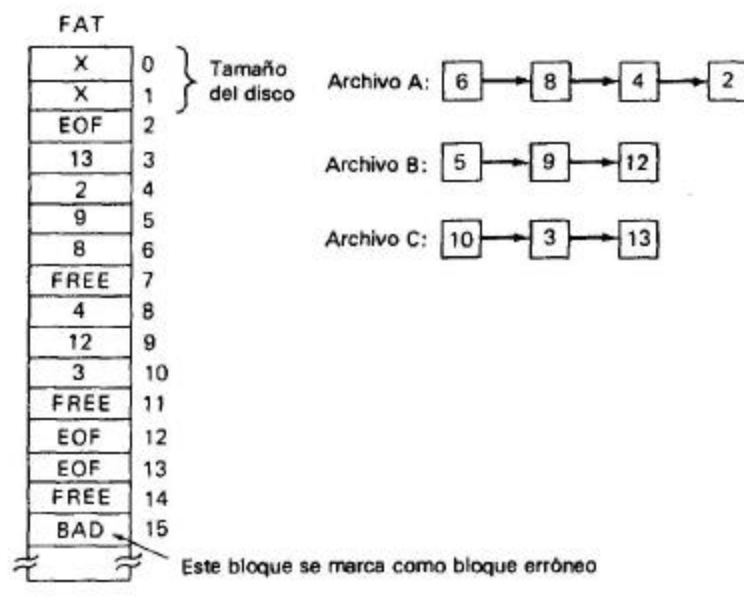
Asignación mediante lista enlazada con índice

Estas dos desventajas que se presentan pueden ser solucionadas parcialmente mediante una mejora del mismo esquema llamada Asignación mediante lista enlazada

con un índice. Este es el esquema utilizado por MS-DOS y consiste en tener en una tabla (llamada FAT sigla en inglés que significa Tabla de Localización de Archivos) almacenada en memoria con todos los punteros a bloque siguiente. Cada índice de la tabla indica una dirección de bloque, y el contenido de la posición apuntada por ese índice señala la dirección del siguiente bloque.

En la tabla, además de los punteros se almacena información adicional como ser: marcar un bloque que está malo para no utilizarlos, tamaño del disco, etc.

A continuación se presenta un ejemplo de cómo funciona esto.



Las ventajas de este esquema son: el acceso aleatorio de los bloques es más rápido que en el caso anterior ya que se puede localizar el bloque a leer mirando en la tabla que está en memoria y luego acceder directamente al bloque almacenado en el disco. Además, teniendo los índices almacenados en una tabla aparte, se puede utilizar todo el bloque para almacenar los datos del archivo.

La principal desventaja de este esquema es que la tabla puede ser muy grande y que se utilicen muchos recursos de memoria para esta acción.

Nodos-i

Este esquema es utilizado por Unix y consiste en tener una pequeña tabla llamada Nodo-i, la que contiene los atributos y direcciones de los bloques de archivo. Cada archivo tiene su correspondiente Nodo-i. El esquema con Nodos-i es utilizado por Unix de la siguiente manera.

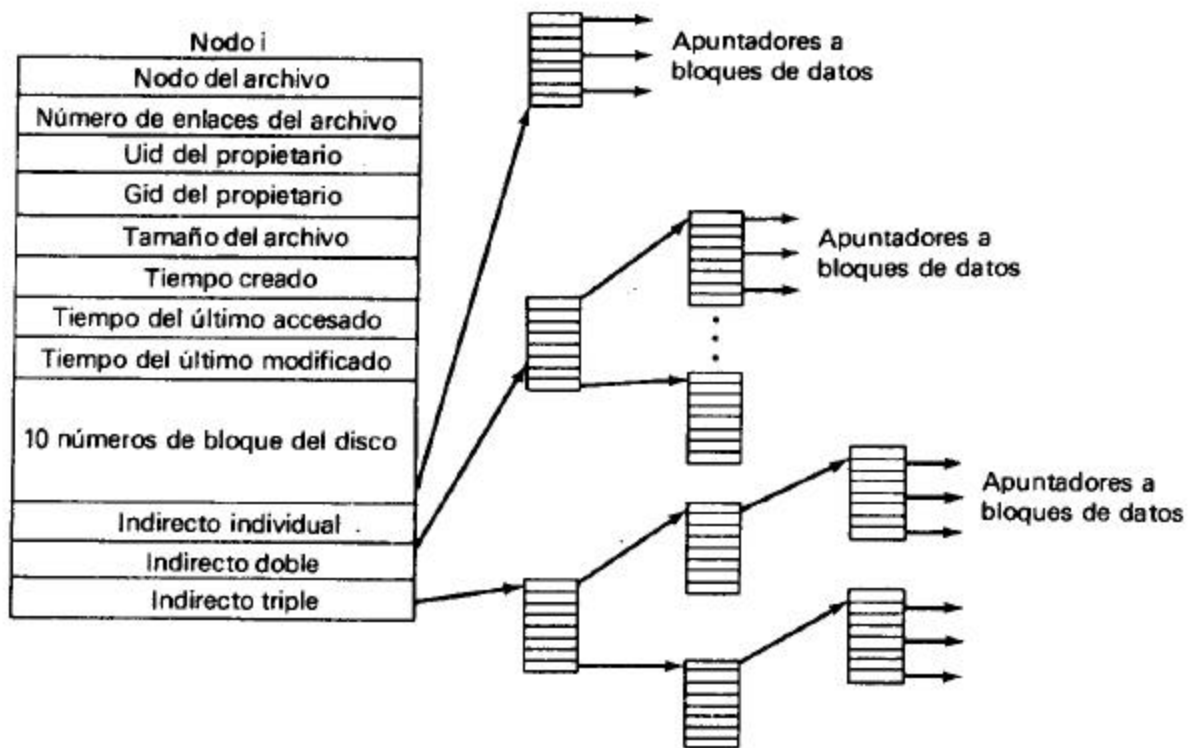
La tabla de Nodo-i asociada a un archivo posee 10 ó 12 entradas que contienen direcciones directas a bloques que ese archivo utiliza. Posteriormente, si el archivo es muy grande y utiliza todas estas entradas para marcar sus bloques, se tiene una entrada

que se denomina simplemente indirecta, que contiene una dirección de un bloque que en lugar de almacenar los datos del archivo almacena más direcciones de bloques de datos.

Luego, si aún no alcanza para almacenar todo el archivo se tiene una entrada denominada doblemente indirecta que apunta a un bloque en el cual se encuentran las direcciones a otros bloques los cuales almacenan las direcciones efectivas de más bloques de datos. Finalmente, se dispone de una tercera y última entrada en el nodo—i que tiene una entrada llamada triplemente indirecta, que agrega un nivel más de indirecciónamiento para obtener las direcciones efectivas de los bloques de datos, ampliando así de forma considerable la capacidad de direccionar bloques.

Este esquema que más bien parece un trabalenguas lo veremos más claro en el siguiente gráfico.

Dependiendo del número de bloques que necesite cada archivo se utilizan o no las entradas correspondientes de la tabla.

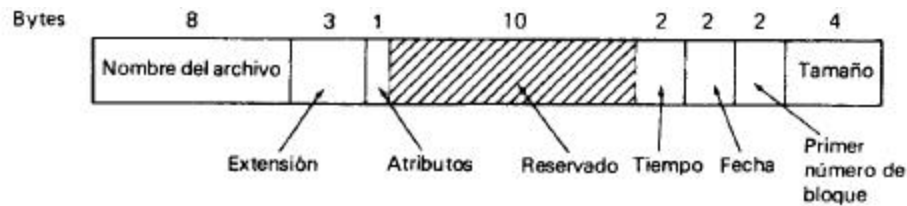


Implementación de Directorios

Como ya dijimos el directorio es un archivo especial, como una lista en donde cada renglón se conoce como entrada de directorio y contiene el nombre del archivo con sus respectivos atributos. Dichos atributos pueden estar definidos en la misma entrada; o bien, en la entrada existe un puntero a una estructura, ubicada en otro lugar, que contiene los

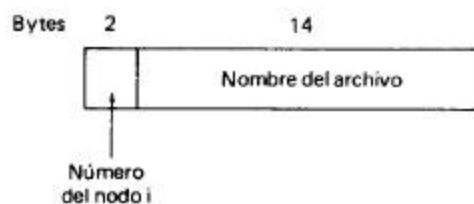
atributos. Por otro lado, si el sistema soporta una organización jerárquica de directorios, entonces una entrada del archivo directorio puede corresponder a otro archivo de directorio, el cual a su vez también puede contener una entrada que corresponde a otro directorio.

La implementación de directorios depende de cada sistema. Por ejemplo, en MS-DOS, que soporta una organización tipo jerárquica, los directorios están implementados de manera que cada entrada en el archivo directorio contiene el nombre del archivo o directorio y los atributos que le pertenecen. Además cada entrada contempla el número del primer bloque que le corresponde al archivo, el resto de los bloques se obtienen de la tabla FAT.



Entrada del directorio de MS-DOS.

Otro ejemplo de implementación es Unix, que también es de organización jerárquica: cada entrada en el archivo directorio contiene el nombre del archivo con el número de nodo-i asociado a ese archivo. Los atributos del archivo se encuentran en el nodo-i. Si el archivo corresponde a otro directorio, entonces dentro del nodo-i correspondiente a él, se encuentra un bloque que contiene información de otro archivo directorio. En cambio si el archivo corresponde a información, los bloques que se obtienen del nodo-i corresponden a datos de algún archivo.



Entrada del directorio de UNIX.

Confiabilidad del Sistema de Archivos

La confiabilidad de archivos tiene que ver con la garantía que poseen los usuarios de almacenar su información en memoria secundaria para luego recuperarla. Desde este punto de vista los sistemas multiusuarios son más complejos pues deben proporcionar a todos los usuarios dichas garantías. No es lo mismo para el caso de un sistema

monousuario como MS-DOS por ejemplo, donde la información que se almacena es sólo de un usuario a la vez.

Uno de los aspectos que se considera para el resguardo de la información de los usuarios tiene que ver con los discos que se utilizan para almacenar información. Estos por lo general, vienen con fallas desde su fabricación (es muy caro producirlos sin fallas). En general, se usan dos mecanismos para manejar los bloques malos de un disco; una es por hardware y otra, por software.

La solución por hardware consiste en dedicar un sector del disco para almacenar todos los sectores malos. Así, cuando el controlador de disco da el formato inicial del disco, determina los sectores malos y los inhabilita sacándolos de las listas de sectores disponibles, de manera que no puedan ser ofrecidos por el sistema de archivo para guardar información en ellos.

La solución por software considera un archivo en donde se guardan todos los números de sectores malos, eliminándolos de los disponibles.

Consistencia del Sistema de Archivos

La consistencia está relacionada con la validez de la información. Por ejemplo, cuando un usuario modifica un archivo, y el sistema por algún motivo no lo pudo modificar sin comunicárselo, se dice que la información que maneja el usuario no es consistente con la del sistema. Este detalle puede ser bastante terrible cuando la información es importante (por ejemplo un depósito en un banco). Esto cobra mayor importancia cuando se trata de información del propio sistema

En general, los sistemas tratan dos tipos de inconsistencias: inconsistencia de archivo, y de bloque.

Para entender la inconsistencia por archivo debemos previamente comentar el manejo de referencias a archivo que maneja, por ejemplo Unix. Supongamos que tenemos un archivo X en un determinado directorio. En cualquier otro directorio, Unix permite hacer una referencia a X, de manera que parezca que pertenece a este otro directorio, pero en realidad es nada más una referencia, existiendo un único archivo real almacenado y múltiples referencias a éste. Estas referencias se denominan Enlaces Reales

Entonces tomando como ejemplo Unix tenemos que: la consistencia por archivo la realiza a través de Enlaces Reales, en donde posee un contador por archivo enlazado, de manera que no permite su eliminación, mientras ese contador no esté en 0. Por su lado, la consistencia de bloques, la realiza mediante dos tipos de contadores: un contador de bloque ocupado y un contador de bloque libre. Estos contadores existen para cada número de bloque.

A continuación se presenta un esquema de cómo realiza la consistencia de bloques. En el primer caso, se presenta una configuración correcta; en las siguientes hay errores. Analícelas.

libres	1	1	0	0	0	1	1	0
ocupados	0	0	1	1	1	0	0	1

libres	1	0	1	0	1	0	0	2
ocupados	0	1	0	2	0	1	1	0

SEGURIDAD Y PROTECCIÓN DEL SISTEMA DE ARCHIVOS

Los términos de seguridad y protección están muy relacionados entre sí. El concepto de seguridad tiene una cobertura más amplia que el concepto de protección. Se dice que la seguridad engloba al sistema como un todo; considera aspectos del medio donde está inserto el sistema, como por ejemplo aspectos políticos, administrativos, legales y también técnicos. En cambio, la protección es un asunto más interno al sistema tiene que ver directamente con los aspectos técnicos que maneja el sistema para permitir el acceso controlado a los programas y datos que él maneja.

En forma más detallada se dice que la protección debe contemplar tanto las políticas como los mecanismos que regulan el acceso al sistema. Cuando se habla de políticas se refiere a *qué* se hará y cuando se habla de mecanismos, se habla de *cómo* se hará algo. Las políticas, por lo general, son determinadas por el diseño del sistema y deben ser de conocimiento de los administradores y de los usuarios el sistema. EL sistema debe diseñarse en forma flexible para estar preparados ante los cambios de las políticas en el tiempo.

Algunas de las facetas más importantes de la seguridad del sistema incluye la pérdida de datos y el ingreso de usuarios no autorizados. La pérdida de datos puede ocurrir de variadas maneras: pueden ser fallas del sistema (Hardware o software) o condiciones del ambiente no soportadas (como incendios, inundaciones, etc). El ingreso de intrusos al sistema, en general, es realizado por personas que desean romper la seguridad del sistema tomando la acción de hacerlo como un desafío.

Control de Acceso

Por lo general, para controlar el acceso de usuarios al sistema se utilizan mecanismos de autenticación de usuarios. Dentro de este esquema lo más conocido es el uso de contraseñas.

Contraseñas

Este mecanismo, básicamente, consiste en que un usuario define una contraseña que le permite autenticarse. Una vez definida esta contraseña los posteriores ingresos del usuario al sistema son permitidos si y sólo si ingresa la contraseña antes definida cuando el sistema se la pide. La contraseña es manejada por el sistema en forma encriptada. Esto significa que una vez que el usuario ha definido su clave de ingreso ésta es encriptada mediante algún algoritmo y es guardada en algún archivo especial. La información guardada en este archivo se utiliza para validar posteriormente la clave de ingreso de los usuarios.

En Unix, por ejemplo, se utiliza este mecanismo en donde las claves encriptadas son guardadas en el archivo *Password* del sistema. Unix permite definir claves de hasta 8 caracteres a las cuales le agrega un número de 12 dígitos y luego las encripta. El número se agrega para aumentar la cantidad de combinaciones de posibles claves para hacer más difícil la tarea de los intrusos de tratar averiguar las claves.

Respecto al tema de las contraseñas es muy importante lo especial que puede ser la clave del usuario, pues está comprobado que la mayor parte de las veces los usuarios eligen claves relacionadas con su vida privada; por ejemplo, colocan nombres de personas, fechas, números de licencias, etc.. Una investigación hecha al respecto, que consistió en construir una lista de posibles claves como las mencionadas, encriptarlas y compararlas con las claves encriptadas del sistema, arrojó como resultado una coincidencia del 85% aproximadamente.

Identificación física

Existen además otros mecanismos de validación de usuarios. Estos incluyen reconocimiento de características físicas de los usuarios. Dentro de este esquema se encuentran los reconocimientos de huellas digitales, tarjetas magnéticas con claves (como los cajeros), análisis del largo de los dedos, reconocimiento de orina, etc.

Atentados contra la seguridad

La seguridad de los sistemas puede ser atacada principalmente mediante dos conocidos tipos de software:

- Virus.
- Gusanos

Virus

Un virus consiste en un segmento de código que se agrega a un programa con la intención de infectarlo, es decir, evitar que se comporte correctamente. Por lo general, los

virus funcionan de la siguiente manera: la persona que construye un virus lo hace a través de un programa aparentemente inofensivo. Cuando este programa es ejecutado, empieza a analizar los programas ejecutables que encuentra en el sistema y si no están infectados los infecta. Esto lo hace agregando el virus al final de su código ejecutable y colocando en el inicio una sentencia que haga que antes de que se inicie el programa se ejecute el virus y luego continúe. De esa manera cuando este programa se ejecute, intentará infectar otros programas de la misma manera.

También existen virus que infectan el sector de arranque del sistema, por lo cual el computador queda inutilizado.

La mejor manera de prevenir los virus es mediante software garantizado.

Gusanos

Un gusano es un programa completo en sí, que actúa en forma independiente, no en forma de parásito, como el virus. Los gusanos generan copias de sí mismos y afectan el rendimiento del sistema sobreutilizando sus recursos. Los gusanos se propagan mediante redes de computadores. Una de los gusanos más famosos en la historia es el *gusano de Internet*, que entró en operación un día de 1988 y que en pocas horas se había propagado por muchas instituciones haciendo colapsar sus sistemas.

El gusano si lograba entrar en un sistema se ejecutaba; si ya existía el gusano allí, salía; si entraba por séptima vez, se volvía ejecutar (para asegurar su permanencia en el sistema). Esto produjo una enorme cantidad de gusanos en cada uno de los sistemas infectados que las máquinas no tenían más remedio que parar.

Mecanismos de Protección

En esta sección mencionaremos las técnicas más utilizadas por los sistemas para proteger la información de los usuarios y del propio sistema. Las técnicas que veremos hacen una distinción entre los objetos que se pueden o quieren proteger y los mecanismos que se utilizan para hacerlo. En general nosotros veremos en más detalle los mecanismos.

Dominios de protección

El dominio es un concepto que se define como un par (objeto, derechos), es decir que para un objeto en particular se definen ciertos derechos. En un sistema computacional se pueden distinguir variados objetos, sean éstos hardware como impresoras, discos, etc, o software como proceso, archivos, etc. El concepto de dominio surge por la necesidad de, por un lado, controlar el poder de los objetos en el sistema; y por otro lado, por la capacidad de utilización compartida de ciertos objetos.

En el sistema operativo Unix, por ejemplo, se definen dos dominios; el dominio UID y el GID. El primero abarca los objetos que pertenecen a un usuario particularmente y el

segundo abarca objetos definidos en un grupo (los grupos pueden ser, por ejemplo, los profesores, los alumnos).

Por otro lado, un proceso en Unix está formado por una parte de usuario y otra parte de sistema. La parte de usuario está restringida a ciertas operaciones. Cuando un proceso hace una llamada a sistema el proceso cambia de dominio usuario a dominio núcleo. En este último modo tiene permiso a efectuar funciones privilegiadas.

Matriz de acceso

El mecanismo de protección que concierne a los dominios de protección puede ser almacenada mediante una matriz de acceso ordenada de manera que los objetos se almacenen en las columnas y los dominios se almacenen en las filas, tal como muestra la siguiente figura. Allí manejamos cuatro objetos y cinco dominios, y en la intersección de cada uno de ellos, se establece cuáles son los derechos establecidos, pudiendo ser X (Ejecución), R (Lectura) y W (Escritura). Con las iniciales tomadas de los términos en inglés.

	Objeto 1	Objeto 2	Objeto 3	Objeto 4
Dominio 1	RW		X	
Dominio 2		RWX		
Dominio 3				
Dominio 4		RW		RX
Dominio 5	RX		R	

El problema que surge con este tipo de almacenamiento es que, por lo general, la matriz es rara, es decir, tiene muchos casilleros vacíos. Por esta razón se usan otras maneras de almacenar la información como las listas de control de acceso y listas de capacidades.

Listas de control de acceso (ACL, Access Control List)

Esta forma de almacenar la información de los dominios se caracteriza porque cada columna se puede almacenar como una lista de acceso para cada objeto, es decir se almacena la misma información de la matriz ordenada por columna. Este esquema permite obtener un conjunto de pares de (dominio, derechos) para cada objeto. Por ejemplo:

```
Objeto1 : (dominio1, RWX), (dominio2,R),(dominio3,WX)
```

Este esquema es utilizado por Unix, tal como se muestra en los siguientes ejemplos.

```
archivo1 : (Juan, estudiantes, RX), (Pedro, Profesor, RWX)
archivo2 : (Pepe, administrador, RWX)
```

La Lista de control de Acceso en Unix es manejable por el dueño del objeto mediante el comando *chmod* el cual permite modificar los bits de protección de derechos sobre los dominios.

Listas de Capacidades

La Lista de Capacidades, por su parte, consiste en almacenar la información de la matriz de dominios por filas. Esto significa que una lista de capacidades para un dominio es una lista de objetos y los derechos que se tienen para cada uno de ellos. La lista de capacidades es un objeto protegido, al cual el usuario tienen acceso a él en forma indirecta. La lista de capacidades se mantiene protegida por el sistema operativo y éste no permite que las capacidades se muevan al espacio de direccionamiento del usuario. Una lista de capacidades es de la siguiente manera:

```
dominio1 : (objeto1 , RWX), (objeto2,RX), (objeto5,X)
```

BIBLIOGRAFÍA

- SISTEMAS OPERATIVOS, Williams Stallings, Segunda Edición, Ed PRENTICE HALL
- SISTEMAS OPERATIVOS MODERNOS, Andrew Tanenbaum, Tercera Edición, Ed PRENTICE HALL
- Apuntes de la Cátedra COMPUTACION II, Universidad Tecnológica Nacional -FRR
- Apuntes de la Cátedra SISTEMAS OPERATIVOS I, Instituto Universitario Gastón Dachary
- <http://www.inf.udec.cl>

RESUMEN

Dos aspectos muy vinculados entre sí fueron analizados en este capítulo, el manejo de Entrada / Salida y el manejo de Archivos.

El sistema de E/S se puede dividir básicamente en dos partes: el componente Hardware y el componente Software. El hardware está determinado por el *dispositivo físico*, el cual consta de componentes mecánicos y componentes electrónicos, denominado *controlador*, que es el encargado de actuar sobre el dispositivo. Estos, los clasificamos en dispositivos de Bloque y dispositivos de Carácter según la manera en que tratan a los bit y un tercer tipo llamado Relojes.

El software de E/S es el encargado de acceder y usar los distintos dispositivos existentes, para lo cual, en general, su estructura tiene mínimamente funciones para solucionar errores, manejarse en forma independiente, tratar y reconocer distintos dispositivos, etc.. En general el software de E/S se compone de diferentes capas donde cada una va atacando y solucionando determinados problemas, y entre todas logran el manejo eficiente de los dispositivos.

De los distintos dispositivos de un sistema, el disco es quizás el de uso más frecuente, es por eso que los Algoritmos de planificación de discos que implementa el Sistema Operativo para manejarlos son muy importantes, entre ellos estudiamos el FCFS, SSTF, SCAN, C-SCAN, LOOK y C-LOOK, cada uno con sus características particulares.

El sistema de archivos es una de las partes más visibles del Sistema Operativo. Los usuarios utilizan los archivos para almacenar sus datos o programas en forma persistente. Esto quiere decir que la información no se pierde una vez que el computador es apagado. Para almacenar dicha información se utilizan dispositivos de memoria secundaria, como discos, cintas, etc. El sistema de archivos proporciona al usuario una interfaz sencilla, amigable y organizada, que le permite almacenar y organizar su información.

Dicha interfaz, conocida como la visión de Arriba–Abajo, el sistema la proporciona principalmente mediante archivos y directorios cada uno con características particulares y con operaciones definidas sobre ellos que permiten trabajarlos.

Por el otro lado, es decir en la Visión Abajo–Arriba, el Sistema Operativo administra dicha información para que la unidad de entrada / salida actúe sobre los dispositivos de almacenamiento secundario, trabajando con distintas forma de implementación: adyacente, listas ligadas, Nodos–i, etc.

Así también, una parte importante de las operaciones que el sistema otorga, tiene relación con los mecanismos de seguridad y protección donde trabajamos con contraseñas, listas de acceso, lista de capacidades, etc. controlando el acceso indebido de usuarios a la información almacenada.

EVALUACIÓN DEL MÓDULO 4

Alumno:

Desarrollar

1. Decimos que los dispositivos, según la manera en que tratan a los bits se clasifican en “De Bloque” y “De Carácter”. ¿Por qué cree necesario ésta separación, en lugar de tratarlos a todos por igual?
2. El software para manejar las Entradas/Salidas se organiza en capas, ¿cuáles son las ventajas de este tipo de organización?.
3. Si se fabrica un nuevo dispositivo, que el sistema no está manejando actualmente, de las capas del software del Sistema Operativo que maneja la E/S ¿cuáles serán las que con seguridad deberán ser actualizadas? ¿Por qué?
4. Ud. compra un disco rígido y en la etiqueta del mismo dice “Tiempo promedio de acceso = 300 ms” que sería el tiempo que se demora en acceder a la información almacenada, sea para escribir o sea para leerla. ¿Qué hace durante todo este tiempo el disco?
5. Hemos estudiado varios algoritmos de planificación de disco, FIFO, SSTF, etc. ¿Cuál es el objetivo de estos algoritmos?
6. Recordando un poco la unidad de estudio número dos, un proceso puede estar en tres estados: listo, ejecución y bloqueado. ¿En cuál de estos estados influyen los algoritmos de planificación de disco? ¿Por qué?.
7. Dada la siguiente serie de peticiones al disco que están almacenada en la cola, indicar en que orden serán atendidas si el algoritmo de planificación de disco es del tipo SSTF y la última solicitud servida fué en el cilindro 10 (si lo prefiere puede hacerlo en forma gráfica)

15; 32; 24; 21; 9; 60; 5; 55; 41; 30; 22; 12; 18; 64 y 8

8. ¿Por qué se dice que un directorio es un archivo especial?
9. Con la siguiente información de directorios, armar el árbol correspondiente

C:\windows

C:\dos

C:\windows\system

C:\mis documentos\graficos

C:\mis documentos\Word

C:\archivos de programa\Microsoft Office

¿Qué tipo de rutas son éstas? ¿Absolutas o relativas?

10. Si un archivo utiliza 8 bloques y para direccionarlo utilizamos nodos-¿Con cuántos accesos al disco leemos el último bloque del archivo?. ¿y cuantos serían si se almacena con una lista ligada? ¿Por qué?
11. Clasificar los siguientes conceptos en “ de Seguridad” y “de Protección” según corresponda.
 - ✓ Acceder a la red informática
 - ✓ Entrar en la sala de computadoras
 - ✓ Leer un sector de la memoria o del disco
 - ✓ Ejecutar una instrucción del Sistema Operativo
 - ✓ Borrar archivos del disco
 - ✓ Archivos de sólo lectura
12. Diferencie un Gusano de un Virus.