

## TRABAJO PRÁCTICO 7

### EJERCICIO 7.1

Las operaciones básicas que debería implementar son creación, escritura, lectura, reposición, eliminación y truncado.

### EJERCICIO 7.2

Los atributos que usualmente se le asignan a un archivo, son:

Nombre, tipo, locación, tamaño, protección, hora, día e identificación de usuario

### EJERCICIO 7.3

### EJERCICIO 7.4

#### **Mandatorio u obligatorio**

Una vez que un proceso adquiere un candado obligatorio, el sistema operativo se encargará de imponer las restricciones correspondientes de acceso a todos los demás procesos, independientemente de si éstos fueron programados para considerar la existencia de dicho bloqueo o no.

#### **Consultivo o asesor**

Este tipo de bloqueos es manejado exclusivamente entre los procesos involucrados, y depende del programador de cada uno de los programas en cuestión el solicitar y respetar dicho bloqueo.

### EJERCICIO 7.5

Permite que tanto el usuario como el sistema operativo puedan saber de que tipo de archivo se trata con solo visualizar el nombre.

### EJERCICIO 7.6

Un archivo es almacenado en disco en una secuencia de bloques de tamaño fijo. El último bloque tendrá siempre un espacio inutilizado, causando fragmentación interna. Cuanto más grande sea el tamaño de los bloques, habrá mayor fragmentación interna.

### EJERCICIO 7.7

16 KB = 16384 bytes

A)

41600 bytes => 3 bloques

$3 * 16384 - 41600 = 7552$

Fragmentación interna de 7552 bytes

B)

640000 bytes => 40 bloques

$40 * 16384 - 640000 = 15360$

Fragmentación interna de 15460 bytes

C)

4064000 bytes => 249 bloques

$249 * 16384 - 4064000 = 15616$

Fragmentación interna de 15616 bytes

### EJERCICIO 7.8

Una ruta absoluta comienza en la raíz del disco y sigue bajando el camino hasta el archivo especificado.

Una ruta relativa define un camino desde el directorio actual, permitiendo subir y bajar en la jerarquía de árbol.

### EJERCICIO 7.9

Para que un archivo pueda ser abierto, un sistema de archivos debe ser montado previamente para estar disponible para ser procesado por el sistema. El sistema operativo recibe el nombre de un dispositivo y el lugar de la estructura de archivos a donde debe adjuntar el sistema de archivo (llamado punto de montaje). A continuación, el SO verifica que el

dispositivo posea un sistema de archivos válido. Finalmente, refleja en la estructura de directorios que el dispositivo fue montado.

A)

B)

#### EJERCICIO 7.10

#### EJERCICIO 7.11

FTP es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP (Transmission Control Protocol), basado en la arquitectura cliente-servidor. Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo.

A)

B)

#### EJERCICIO 7.12

Es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. LDAP también se considera una base de datos (aunque su sistema de almacenamiento puede ser diferente) a la que pueden realizarse consultas.

#### EJERCICIO 7.13

Lectura, escritura, ejecución, anexar, eliminar, listar.

#### EJERCICIO 7.14

El principal problema de las listas de accesos es la longitud. Si se desea que todos puedan leer un archivo, se deben listar todos los usuarios con acceso a lectura. Esto provoca dos consecuencias indeseables:

- Construir tal lista puede ser tedioso, especialmente si no se conoce la lista de usuarios del sistema
- La lista de directorios que previamente tenía un tamaño fijo ahora deberá tener un tamaño variable, resultando en una administración de espacio más complicada.

Estos problemas se pueden resolver con una lista de acceso reducida. Suelen utilizarse tres clasificaciones

- Propietario: el usuario que creó el archivo
- Grupo: Un conjunto de usuarios que comparten el archivo y necesitan acceso similar
- Universo: Cualquier otro usuario del sistema

#### EJERCICIO 7.15

##### **Lista Lineal**

Se utiliza una lista lineal de nombres de archivos con punteros a los bloques de datos. Una lista lineal de entradas de directorios requiere una búsqueda lineal para encontrar una entrada particular.

##### **Tabla Hash**

En este método, una lista lineal guarda las entradas de directorios, pero también se utiliza una estructura de datos Hash. La tabla hash toma el valor computado del nombre de archivo y devuelve un puntero al nombre de archivo en la lista lineal.

#### EJERCICIO 7.16

Una partición *raw* es una parte de un disco físico al que se accede al nivel más bajo posible. Se crea una partición sin procesar cuando se crea una partición extendida y se asignan a ella particiones lógicas sin ningún formato. Una vez completado el formateo, se denomina partición *cooked*.

### EJERCICIO 7.17

El *Virtual File System* es un subsistema del kernel que posibilita la interoperabilidad de distintos FS en el mismo kernel. Una misma máquina accede a varios sistemas de archivos porque soporta distintos dispositivos de almacenamiento secundario, cada uno con su file system propio.

Se aísla la funcionalidad básica de las llamadas al sistema de los detalles de implementación de cada operación I/O en cada FS.

La implementación del file system está compuesta de tres niveles:

- El primer nivel es una interfaz del file system basada en los descriptores de archivo y las llamadas al sistema (open, read, write y close)
- El segundo nivel es llamado File System Virtual. Cumple dos funciones importantes
  - Separa las operaciones genéricas del sistema de archivos con respecto a su implementación definiendo una interfaz clara, uniforme que es capaz de representar cualquier función o comportamiento de un sistema de archivos concebible.
  - Proporciona un mecanismo para representar de forma coherente los archivos a través de una red. Para ello utiliza una representación de archivos llamada VNODO.
- El tercer nivel lo conforman los protocolos de cada file system montado.

### EJERCICIO 7.18

A)

1 para todos los casos

B)

500, 200, 1, 20

C)

5, 3, 1, 1

### EJERCICIO 7.19

	Contiguo	Enlazada	Indexada
A	201	1	1
B	101	52	1
C	1	3	1
D	198	1	0
E	98	52	0
F	0	100	0

### EJERCICIO 7.20

Si un puntero a un bloque requiere 4 bytes, un bloque puede direccionar hasta  $8 \text{ KB} / 4 \text{ bytes} = 2048$  bloques

Los 12 bloques de disco directo pueden referir a 12 bloques

El disco indirecto simple puede referir a 2048 bloques

El disco doble refiere a  $2048 * 2048$  bloques

El disco triple refiere a  $2048 * 2048 * 2048$  bloques

$(12 + 2048 + 2048^2 + 2048^3) * 8 \text{ KB} = 64 \text{ TB}$

### EJERCICIO 7.21

Si todas las extensiones son del mismo tamaño y el tamaño está predeterminado, simplifica el esquema de asignación de bloques. Un mapa de bits simple o lista libre para extensiones sería suficiente.

Si las extensiones pueden ser de cualquier tamaño y se asignan dinámicamente, se requieren esquemas de asignación más complejos. Puede ser difícil encontrar una extensión del tamaño adecuado y podría haber fragmentación externa. Se podría usar el asignador de sistema Buddy.

Cuando las extensiones pueden ser de unos pocos tamaños fijos, y estos tamaños están predeterminados, uno tendría que mantener un mapa de bits independiente o una lista libre para cada tamaño posible. Este esquema es de complejidad intermedia y de flexibilidad intermedia en comparación con los esquemas anteriores.

#### EJERCICIO 7.22

Mantener el vector de bits en memoria mejoraría la performance, pero solo es posible con discos pequeños. Un disco de 1.3 gb con bloques de 512 bytes podría necesitar un vector de bits de hasta 310 kb para llevar registro de los bloques libres.