



Implementación de un sistema de archivos - Proyecto 4

Sistemas Operativos

Integrantes:

- Manuel González
- Matías Parra

1. Introducción

Este proyecto consta del desarrollo de un programa en Java que simule un sistema de archivos. Nosotros elegimos un sistema de archivos con asignación indexada porque es el más eficiente respecto a la memoria (evita fragmentación y tiene perfecta consistencia en los datos), la única desventaja es que el archivo puede crecer hasta que se le acabe el espacio en el bloque de los índices, pero dado que este es un sistema pequeño, esa desventaja no resulta un problema.

2. Descripción del programa y su diseño

Dado que un sistema de archivos tiene varias partes, el volumen será un archivo de texto que lo simulará, y el programa tendrá cuatro clases importantes (sin contar la clase main que es la que inicializa el programa y muestra la interfaz por consola), estas son el disco (encargado de simular el volumen que es un archivo de texto y cargarlo al sistema o memoria), los bloques (que pueden ser N hasta 128 y son los que almacenarán los archivos en memoria y después guardarlos en el volumen a través de la clase disco), el directorio (almacenará la lista de de los archivos, sus nombres y su bloque índice), y finalmente está el FCB (se encarga de almacenar los datos de un archivo abierto). Estas decisiones se tomaron porque se tenía en mente la asignación indexada en disco, por ende, la relación entre las clases y sus atributos es muy importante, por ejemplo, en la clase directorio hay un hashMap que contiene el nombre de los archivos y un índice (el índice de la lista que está en disco para poder obtener el bloque índice que tiene todos los índices de los bloques que forman el archivo). Los bloques tienen el tamaño de bytes que acepta (512 bytes), una bandera que muestra si el bloque está ocupado o no, y una lista de los índices de otros bloques (en el caso del que éste bloque sea un bloque índice, sino, la lista estará vacía). El file control block tendrá el nombre del archivo, los índices de los bloques que componen el archivo y el tamaño de éste. El FCB solo muestra un archivo a la vez, es decir, que solo se puede abrir un archivo. El disco es el más general de todas las asignaciones de archivos, porque tiene una lista de bloques que será igual a la cantidad de sectores y realiza operaciones de escritura y lectura al volumen. Finalmente, tenemos la clase main que es la que inicializa el proyecto y muestra un formulario por consola para poder interactuar con el usuario, y también hace uso de funciones y clases del sistema.

Las operaciones que tiene el sistema son las siguientes:

- **Format:** lo que hace es crear el volumen nuevamente (sobreescribiendo el anterior) y creando el nuevo con el tamaño ingresado por el usuario.
- **Create:** se encarga de crear un archivo en el volumen, iniciando con su creación en memoria (dado su nombre y el tamaño en bytes) y posteriormente guardar el archivo en volumen con líneas vacías.
- **Remove:**
- **Open:** busca en disco el archivo que se desea abrir por el nombre, si lo encuentra lo carga en memoria en el FCB, de lo contrario muestra un mensaje de error.
- **Read At:** muestra el contenido del archivo desde el primer bloque hasta el que el usuario ingrese (el usuario ingresa la cantidad de bloques que quiere ver del archivo). El archivo debe de ser abierto previamente y ser almacenado en el FCB.
- **Write At:**
- **PrintFile:** muestra todo el contenido que tiene el archivo abierto y que está contenido en el FCB.
- **List:** imprime la lista de directorios, es decir, todos los archivos mostrando su nombre, el número del bloque índice y se le añade el tamaño del archivo total en bytes (no se cuenta el bloque índice en el tamaño del archivo).

3. Detalles del volumen

El volumen es el archivo de texto que simulará el disco duro de un sistema, por ende tiene que tener una estructura que será explicada a continuación:

DISCO.txt - Notepad

File Edit Format View Help

archivo-4/jopa-17/2560-10/hola-15/

[1->2->3->-1]

[5->6->7->8->9->-1]

Lorem ipsum dolor sit amet, consectetur turpis dui.

oooooooo

ppppppppp

33333333

[11->12->13->14->-1]

[16->-1]

La imagen adjunta muestra el contenido del volumen, en el cual:

- Cada línea es un bloque.
- La primera línea (bloque 0) es la encargada de almacenar el nombre del archivo y la dirección del bloque índice (guarda el directorio).
- De la línea 1 hacia abajo comienzan los bloques que contendrán la información o el contenido de cada archivo.
- Las líneas que tienen corchetes y números son los bloques índices que contienen los números de los bloques que contienen el contenido del archivo (en nuestro ejemplo, [1->2->3->-1] es el bloque índice del archivo llamado "archivo" y contiene espacios en blanco como contenido en las líneas sobre éste, que son la 1, 2 y 3)
- El ejemplo del archivo llamado "hola", tiene su bloque índice en la línea 15 ([11->12->13->14->-1]) y su contenido sobre la línea 15, desde la línea 11 (Lorem ipsum...) hasta la línea 14 (333...) con el número -1 que indica el final de los bloques que componen el archivo.

4. Conclusión

El sistema de archivo creado es con asignación indexada porque sus ventajas son mayores a las desventajas e incluso, en este sistema pequeño, la desventaja no es tan notoria. Las funcionalidades pedidas por el enunciado fueron completadas correctamente, pero hubieron varios problemas en el desarrollo, estos son en la funcionalidad write at y create que costó entenderlas y hacer que funcionaran usando el FCB y el disco sin que fallara. Finalmente, vimos y comprendimos de mejor manera cómo se manejan los archivos en un disco, los tipos de asignación y entender los componentes del sistema como lo es el file counter block, el disco y los bloques.