



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE INGENIERIA – SISTEMAS OPERATIVOS

Proyecto: (Micro) sistema de archivos multihilos

Alumnos:

- Francisco Daniel López Campillo
- Vigi Garduño Marco Alejandro

Profesor:

- Gunnar Eyal Wolf Iszaevich

Semestre: 2024-2

Fecha de entrega: 19-05-2024

Grupo: 06

Documentación

Introducción

Este documento explica de forma detallada el funcionamiento y lógica que desarrollamos para la implementación de un sistema de archivos con base en una especificación y un caso de referencia, que consiste en un archivo denominado *fiunamfs.img*. Se implementaron las funciones básicas que caracterizan al sistema de archivos, como agregar, eliminar, copiar o proporcionar un directorio que indique información sobre nuestros datos. Se emplearon algunos mecanismos de concurrencia para profundizar en la administración de procesos.

Requisitos

- Python 3.x
- Archivo de imagen **fiunamfs.img** (sistema de archivos simulado)

Estructura del Proyecto

El archivo principal del proyecto contiene todas las características necesarias para interactuar con el sistema de archivos llamado FiUnamFS titulado main:

```
|— fiunamfs.img
|— proyecto_sistop.py
```

Descripción del Código

Importación de Módulos

Los módulos que se emplearon para disponer de las herramientas que hicieron posible la implementación de las funciones que conforman a nuestro programa son los siguientes

- **os**: Permite interactuar con el sistema operativo, incluyendo operaciones de archivos y directorios.
- **struct**: Facilita la manipulación de datos binarios en Python, especialmente para leer y escribir enteros y cadenas en un formato específico.
- **threading**: Proporciona soporte para la programación concurrente, permitiendo que el programa maneje múltiples operaciones simultáneamente de manera segura.

Variables Globales

Las variables globales empleadas a lo largo del programa son las siguientes:

- **sector**: Representa el tamaño de un sector en bytes, que en este caso es de 512.
- **cluster**: Calcula el tamaño de un cluster en bytes, que al conformarse de cuatro sectores resulta en 2048.
- **direccionSistema**: Define la ruta del archivo de imagen **fiunamfs.img**.
- **semaphore**: Implementa un semáforo para controlar el acceso concurrente a recursos compartidos.
- **lock**: Implementa un mecanismo de bloqueo conocido como candado para asegurar que solo un hilo pueda acceder a un recurso crítico a la vez.

Funciones

Verificar Existencia del Archivo

Esta función verifica si el archivo de imagen **fiunamfs.img** existe en la ruta especificada. Si el archivo no se encuentra, el programa muestra un mensaje de error y se cierra.

Leer Entero desde el Archivo

Esta función lee un entero desde una posición específica en el archivo de imagen. Utiliza el módulo **struct** para convertir los datos binarios leídos en un entero.

Leer Cadena desde el Archivo

Esta función lee una cadena desde una posición específica en el archivo de imagen. Los datos binarios se convierten a una cadena de texto ASCII y se eliminan los espacios en blanco adicionales.

Validar Sistema de Archivos

Esta función recopila y muestra información sobre el sistema de archivos ubicado en el primer cluster (#0). La información incluye el nombre del sistema de archivos, la versión (24-2), la etiqueta del volumen ("Mi Sistema Favorito"), el tamaño del cluster (2048 bytes), la cantidad de clusters del directorio (4) y la cantidad de clusters de la unidad (720). Todos estos datos son preestablecidos, pero se comprueban en el archivo proporcionado a través de la lectura del archivo, empleando las funciones descritas anteriormente.

Listado de Archivos

Esta función lista el contenido del directorio del sistema de archivos simulado, ignorando las entradas vacías. Muestra información detallada de cada archivo, como el nombre, tamaño, cluster inicial, fecha de creación y fecha de modificación. Se basa principalmente en reconocer el primer byte de cada entrada, ya que se puede identificar su ocupación dependiendo si es / o -.

Copiar Archivo desde FiUnamFS al Directorio Local

Esta función permite copiar un archivo desde el sistema de archivos simulado al directorio local. Verifica si el archivo existe en **fiunamfs.img**, lee su contenido y lo escribe en el sistema de archivos del host. Se utiliza una comparación en formato ASCII, ya que la comprobación por caracteres dio muchos problemas. Datos como el tamaño del archivo o el número del cluster inicial en la sección de datos son cruciales para un correcto proceder.

Eliminar Archivo de FiUnamFS

Esta función inicia declarando un semáforo el cual nos garantiza que el acceso sea exclusivo y así evitamos que los hilos tengan problemas entre sí. Posteriormente hacemos uso de variables, la primera es auxiliar la cual nos permite cambiar el flujo de la función para validar que exista el archivo a eliminar y como segunda variable nos permite convertir el nombre a eliminar en ASCII para una comparación del archivo a eliminar. Se procede a iterar de forma que este reflejado en bloques de 64 bytes por lo que se obtiene la posición actual y en caso de que no sea el archivo '/' nos refleja que procedemos a leer el nombre del archivo y a eliminar los espacios en blanco contenidos. Seguido de esto validamos que el nombre haya sido encontrado o no, en caso de que si procedemos a abrir el archivo para escribir en este, escribiendo '/' que significa que está libre y se imprime en pantalla que si se pudo eliminar el archivo, caso contrario en que no se haya encontrado el archivo, se imprime de igual forma en pantalla que no

se encontró el archivo a eliminar, por ultimo ya sea que se haya encontrado al archivo o no, se libera el semáforo que se empleó al inicio para que otros hilos puedan hacer uso de la siguiente operación a emplear.

Mover Archivo desde el Directorio Local a FiUnamFS

Finalmente, al igual que en la función anterior, se inicia declarando un semáforo el cual nos garantiza que el acceso sea exclusivo y así evitamos que los hilos tengan problemas entre sí. Posteriormente se realiza la validación para ver si el archivo deseado de mover de nuestro sistema a FiUnamFS existe, en caso de que no exista se imprime en pantalla y termina la función, pero en caso de que este si exista, se obtiene el tamaño del archivo a mover, para posteriormente a realizar una validación acorde al tamaño anteriormente capturado y poder determinar así si es posible que el archivo a mover tenga un tamaño el cual sea posible mover para que quepa en FiUnamFS, en caso de que no quepa, se imprime en pantalla que no es posible por esta razón pero en caso de que si sea posible moverlo, se procederá a buscar el espacio suficiente para poder almacenar el archivo deseado, en caso de que no se encuentre este espacio disponible, se imprimirá en pantalla que no fue posible ya que no hay espacio suficiente, caso contrario en donde si exista el espacio, se procede a abrir el archivo FiUnamFS para poder escribir en este el archivo que se desea mover. Luego de esto se declaran las variables que nos van a permitir obtener los metadatos del archivo que se moverá, esto para poder escribir y mostrar esta información la cual será escrita sobre el mismo archivo de FiUnamFs. Por último, se realiza una impresión en pantalla que dadas las condiciones que fueron empleadas y que estas fueron validas, se procedió a realizar el correspondiente movimiento del archivo por lo que se hizo de forma correcta, esperando la tecla 'enter' para seguir con el flujo del programa y poder así liberar el semáforo que se empleó al inicio para que otros hilos puedan hacer uso de la siguiente operación a emplear.

Concurrencia implementada

Para poder garantizar la ejecución del programa y que pueda realizarse de forma segura, se han implementado varias primitivas de concurrencia:

- Uso del semáforo en la función *copiarArchivo*, *eliminarArchivo* y *moverArchivo*.

En esta implementación, se ocupa un semáforo que nos ayudará a impedir que en otro hilo de ejecución se comience una operación de esta naturaleza, ya que si se llegan a interrumpir podría resultar en pérdida o corrupción de información, que es claramente algo indeseado. La ubicación dentro de estas funciones abarca todas las operaciones que las involucran, ya que se realizan validaciones constantes sobre espacio disponible, obtención de cierta información o eliminación de datos. Es ciertamente sencillo reconocer que estas operaciones pueden impedir su realización efectiva al haber más de un hilo en ejecución. En el siguiente código se puede observar su implementación (resulta redundante mostrar la de las otras funciones, ya que es idéntica):

```
def copiarArchivo(nombre_archivo):
    semaphore.acquire()    # Adquirimos el semáforo antes de acceder al archivo
    try:
        # ... (operaciones de copiado)
    finally:
        semaphore.release()    # Liberamos el semáforo después de la operación
```

- Uso del candado en la función *moverArchivo*.

En este caso buscamos garantizar la atomicidad de la sección crítica de esta función, de manera que durante su ejecución no exista otro hilo que pueda acceder a la información compartida que se está manipulando:

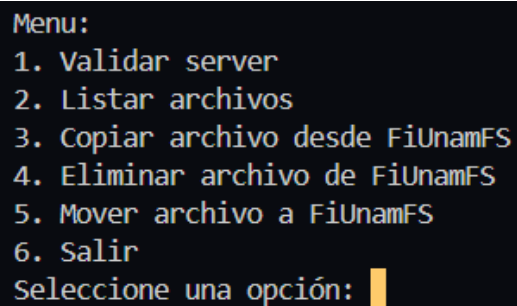
```
def moverArchivo(nombre_archivo):  
    # ... (operaciones de mover)  
    with lock:  
        with open(direccionSistema, "r+b") as archivo_fs:  
            archivo_fs.seek(cluster + cluster_libre)  
            archivo_fs.write(b'D')  
            archivo_fs.write(nombre_archivo_encoded)  
            archivo_fs.write(tamano_archivo_bytes)  
            archivo_fs.write(cluster_inicio_bytes)  
            archivo_fs.write(fecha_creacion_bytes)  
            archivo_fs.write(fecha_modificacion_bytes)  
            archivo_fs.seek(cluster_libre * cluster)  
            archivo_fs.write(contenido_archivo)
```

Menú Principal

El menú principal del programa permite al usuario elegir la operación que desea realizar, proporcionando una interfaz interactiva para validar el sistema de archivos, listar archivos, copiar archivos, eliminar archivos, y mover archivos teniendo una buena experiencia de usuario ya que es bastante intuitiva y evita la generación de texto manteniendo la consola siempre limpia.

Ejemplos y Capturas de Pantalla

Menú del proyecto



```
Menu:  
1. Validar server  
2. Listar archivos  
3. Copiar archivo desde FiUnamFS  
4. Eliminar archivo de FiUnamFS  
5. Mover archivo a FiUnamFS  
6. Salir  
Seleccione una opción: █
```

Validar Sistema de Archivos

1. Seleccione la opción 1 del menú.
2. La información del sistema de archivos se mostrará en pantalla.

```
El sistema se llama: FiUnamFS
La version es: 24-2
Etiqueta del volumen: Mi Sistema Favorito
Tamaño del Cluster: 2048 bytes
Numero de clusters que mide el directorio: 4
Numero de clusters que mide toda la unidad: 720

Presione 'enter' para continuar...
```

Listar Archivos

1. Seleccione la opción 2 del menú.
2. Se listarán los archivos presentes en el sistema de archivos simulado.

```
Listado de archivos en el FiUNAMFS:
Archivo: README.org    , Tamaño: 31222, Cluster Inicial: 6, Creado: 20240508131756, Modificado: 20240508131756
Archivo: logo.png      , Tamaño: 126423, Cluster Inicial: 22, Creado: 20240508131756, Modificado: 20240508131756
Archivo: mensaje.jpg   , Tamaño: 254484, Cluster Inicial: 84, Creado: 20240508131756, Modificado: 20240508131756

Ingrese 'enter' para continuar...
```

Copiar Archivo desde FiUnamFS a Directorio Local

1. Seleccione la opción 3 del menú.
2. Ingrese el nombre del archivo a copiar.

```
Listado de archivos en el FiUNAMFS:
Archivo: README.org    , Tamaño: 31222, Cluster Inicial: 6, Creado: 20240508131756, Modificado: 20240508131756
Archivo: logo.png      , Tamaño: 126423, Cluster Inicial: 22, Creado: 20240508131756, Modificado: 20240508131756
Archivo: mensaje.jpg   , Tamaño: 254484, Cluster Inicial: 84, Creado: 20240508131756, Modificado: 20240508131756

Ingrese el nombre del archivo a copiar a su directorio local: logo.png
Archivo 'logo.png' copiado exitosamente al directorio local.

Ingrese 'enter' para continuar...
```

Eliminar Archivo de FiUnamFS

1. Seleccione la opción 4 del menú.
2. Ingrese el nombre del archivo a eliminar.

```
Listado de archivos en el FiUNAMFS:
Archivo: README.org    , Tamaño: 31222, Cluster Inicial: 6, Creado: 20240508131756, Modificado: 20240508131756
Archivo: logo.png      , Tamaño: 126423, Cluster Inicial: 22, Creado: 20240508131756, Modificado: 20240508131756
Archivo: mensaje.jpg   , Tamaño: 254484, Cluster Inicial: 84, Creado: 20240508131756, Modificado: 20240508131756

Ingrese el nombre del archivo a eliminar de FiUnamFS: logo.png
Archivo 'logo.png' eliminado exitosamente de FiUnamFS.

Ingrese 'enter' para continuar...
```

Mover Archivo desde el Directorio Local a FiUnamFS

1. Seleccione la opción 5 del menú.
2. Ingrese el nombre del archivo a mover.

```
Menu:
1. Validar server
2. Listar archivos
3. Copiar archivo desde FiUnamFS
4. Eliminar archivo de FiUnamFS
5. Mover archivo a FiUnamFS
6. Salir
Seleccione una opción: 5
Ingrese el nombre del archivo a mover a FiUnamFS: hola.txt
Archivo 'hola.txt' movido exitosamente a FiUnamFS.

Ingrese 'enter' para continuar...|
```