



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE MÉXICO



SISTEMAS OPERATIVOS

(Micro)sistema de archivos

Grupo 6

*Nombre:*

Barreiro Valdez Alejandro

Zepeda Baeza Jessica

**Proyecto 4**

**Profesor: Ing. Gunnar Eyal Wolf Iszaevich**

19 de mayo de 2022

# 1. Descripción

Este programa permite generar operaciones en un sistema de archivos proporcionado por el profesor. Este sistema de archivos es de asignación contigua y es de 1440 KB. La información de los archivos se encuentra en un directorio que ocupa los clústeres de 1 a 4. El clúster 0 pertenece al superbloque y almacena metadatos sobre el sistema de archivos. En este sistema de archivos se pueden realizar operaciones como creación, eliminación y consulta de los archivos que se almacenan en el sistema de archivos. Se generó un menú que le permite al usuario tener cinco opciones de las operaciones que puede realizar el programa: listar los contenidos del directorio, copiar un archivo de la computadora al sistema, copiar un archivo del sistema a la computadora, eliminar un archivo y desfragmentar el sistema de archivos. (Wolf, 2022) Para varias se debe especificar el nombre del archivo sobre el cual operar y el nombre de archivo que se va a generar. El menú también cuenta con una opción para salir. Los archivos que se generan se copian dentro del directorio donde se tiene guardado el código si no se especifica una ruta.

## 2. Estrategia para cada funcionalidad

### 2.1. Información del Sistema de Archivos

Lo primero que se realiza en el código es abrir el sistema de archivos con permisos de escribir y leer en binario. Se leen los primeros 60 bytes decodificados en UTF-8 para obtener una lectura con caracteres legibles y se va dividiendo la cadena en otras variables para separar la información que arrojan esos primeros 60 bytes. De manera que, como muestra la Figura 1 se imprime en pantalla: la identificación, la versión, la etiqueta, el tamaño de cluster, el número de clusters para el directorio y para la unidad. De esta información, se hace el casteo a *int* de los últimos tres datos para usarlos a lo largo del programa. También se muestra el menú con las opciones a realizar.

```
[jessicazepeda@MacBook-Pro-de-Jessica BarreiroAlejandro-ZepedaJessica % python3 proyecto4-so.py]

-----Microsistema de archivos-----

Identificación:      FiUnamFS
Versión:             1.1
Etiqueta:            FI-UNAM 2022-2
Tamaño cluster:      2048
Núm. de clusters directorio: 4
Núm. de clusters unidad: 720

-----Menú de opciones-----

Selecciona una opción:
1. Listar los contenidos del directorio
2. Copiar uno de los archivos de dentro del FiUnamFS hacia tu sistema
3. Copiar un archivo de tu computadora hacia tu FiUnamFS
4. Eliminar un archivo del FiUnamFS
5. Desfragmentar
6. Salir
```

Figura 1: Interfaz del programa.

## 2.2. Listar los contenidos del directorio

Para listar los contenidos del directorio se accedió al directorio que es donde se encuentra la información de cada uno de los archivos. El directorio se encuentra en los clusteres 1 a 4 y contiene información útil para la manipulación del archivo. Para el listado de los contenidos del sistema de archivo se realizó una función que lee estos clusteres y si existe un registro con un nombre diferente a los puntos suspensivos, se guarda la información del archivo. De manera que se hace una lista donde en cada elemento existe otra lista con los datos de cada archivo. Los índices de estas listas indican:

- 0: nombre del archivo
- 1: tamaño del archivo en bytes
- 2: cluster donde inicia el contenido
- 3: año de creación del archivo
- 4: mes de creación del archivo
- 5: día de creación del archivo
- 6: hora de creación del archivo
- 7: minuto de creación del archivo
- 8: cluster donde termina el contenido
- 9: segundos de creación del archivo

Posteriormente, en la función para esta opción, se devuelve esta lista con todos los datos de cada archivo. Y mientras se recorre se imprimen los datos siguiendo un cierto formato como se muestra en la Figura 2.

```
Selecciona una opción:
1. Listar los contenidos del directorio
2. Copiar uno de los archivos de dentro del FiUnamFS hacia tu sistema
3. Copiar un archivo de tu computadora hacia tu FiUnamFS
4. Eliminar un archivo del FiUnamFS
5. Desfragmentar
6. Salir

Ingresa la opción: 1

Listando los contenidos del directorio...

README.org          29565  B      5 - 20    2022-05-08    13:57
logo.png            52200  B      17 - 43    2022-05-08    13:57
mensajes.png       158520  B     353 - 431 2022-05-08    13:57
```

Figura 2: Lista del contenido del directorio.

## 2.3. Copiar un archivo del sistema a la computadora

Para copiar un archivo del sistema de archivos a la computadora primero se pide que se ingrese el nombre del nuevo archivo y posteriormente del archivo a copiar. Se busca el nombre en la lista de la información de los archivos y se obtiene su cluster y su tamaño. La variable que almacena el valor de su cluster inicial se inicializa con un valor de -1 de manera que si no se encuentra en nombre en la lista de los archivos existentes, este valor sirve como bandera para indicar que el archivo no existe y no continuar con la función.

Si se encuentra el nombre en la lista, se posiciona a la lectura en el cluster obtenido (haciendo la multiplicación de cluster por 2048), se genera una lectura del tamaño del archivo y se copia dicha lectura a un archivo nuevo, en la computadora del usuario, con el nombre que el usuario indicó. Esto se logra creando un nuevo archivo con modo de escritura binaria y escribiendo lo leído en el archivo. Una vez que se cierra el archivo se indica que la operación se realizó. La ejecución de esta operación se muestra en la Figura 3.

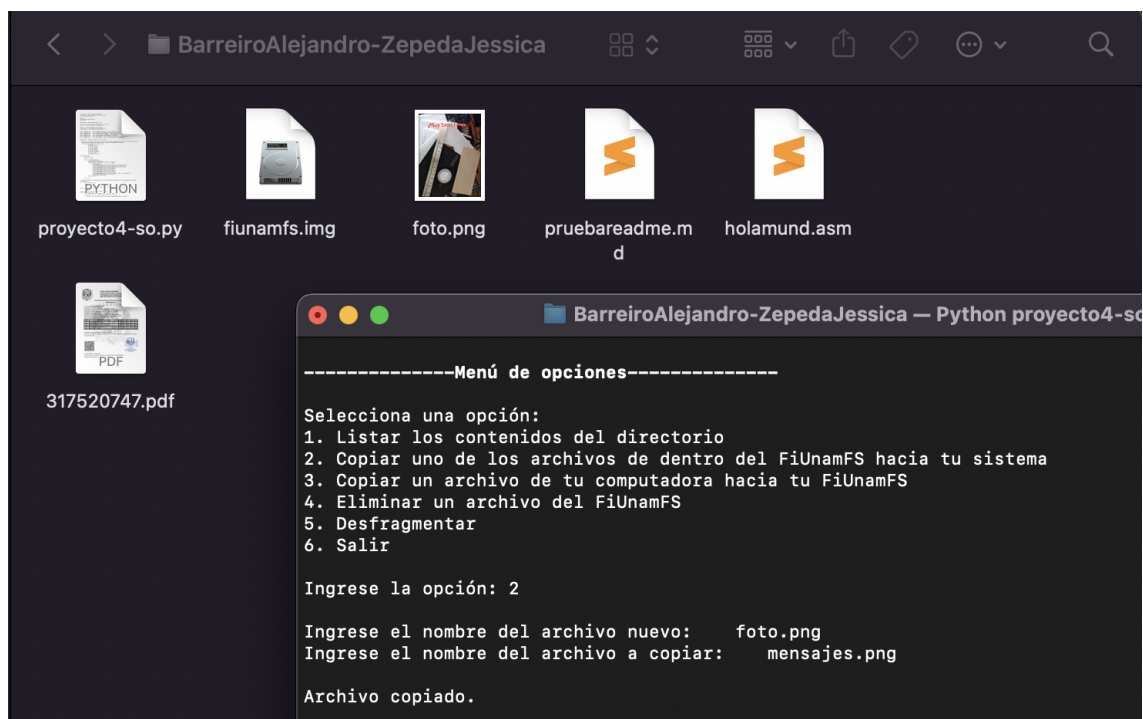


Figura 3: Copia de un archivo de sistema a la computadora.

## 2.4. Copiar un archivo de la computadora al sistema

Para copiar un archivo de la computadora al sistema de archivos primero se pide que se ingrese el nombre del archivo a copiar. Este archivo debe ubicarse en el directorio donde se encuentra el sistema de archivos y el programa. Se intenta abrir y leer el archivo indicado y se maneja una excepción *FileNotFoundException* si el archivo no existe. Haciendo una impresión igual a la de la Opción 2, como se muestra en la Figura 5.

Si se encuentra el archivo se obtiene su tamaño y se lee su contenido. Luego se pide el nombre del nuevo archivo. Después se llama la función que regresa la lista con los archivos existentes. Debido a que en el directorio puede que los archivos no se muestren en el orden como se encuentran en la unidad, se ordena esta lista con respecto al cluster inicial. Por lo tanto se obtiene cuál archivo está después de cuál y con esta información se genera una lista que guarda el espacio entre cada dos archivos. Este espacio se obtiene con la resta del cluster inicial de un segundo archivo y el cluster final de un primer archivo menos 1 y todo esto por el tamaño del cluster. Se resta 1 porque en caso de que sean clusteres continuos, se sabe que no se tiene espacio; sin embargo al hacer la resta y la multiplicación se tendría un espacio de 2048 y este no existe realmente.

Se recorre la lista para encontrar el primer espacio que puede almacenar el archivo y cuando se encuentra, se guarda el valor del cluster siguiente al cluster final del archivo que estará antes de este. Si no se encuentra un espacio significa que se colocará el archivo después del último archivo. Entonces con el cluster inicial definido, se posiciona el archivo del sistema en este lugar, se escribe lo leído del archivo de la computadora y se cierra este último archivo.

Lo último es ingresar los datos de este archivo nuevo en el directorio. Para ello se hace una cadena con los elementos que conforman el directorio; para la fecha de creación y modificación se pasa la fecha actual obtenida a través de la clase *datetime*. Se recorre la sección del directorio hasta encontrar una línea que sea igual a los puntos suspensivos. Como se conoce la dirección de esta línea, se coloca ahí el archivo del sistema y se escribe la cadena. La ejecución de esta función se muestra en la Figura 4.

```

BarreiroAlejandro-ZepedaJessica — Python proyecto4-so.py — 113x37

-----Menú de opciones-----
Selecciona una opción:
1. Listar los contenidos del directorio
2. Copiar uno de los archivos de dentro del FiUnamFS hacia tu sistema
3. Copiar un archivo de tu computadora hacia tu FiUnamFS
4. Eliminar un archivo del FiUnamFS
5. Desfragmentar
6. Salir

Ingrese la opción: 3

Ingrese el nombre del archivo a copiar: 317520747.pdf
Ingrese el nombre del archivo nuevo: prueba.pdf

Archivo copiado.

-----Menú de opciones-----
Selecciona una opción:
1. Listar los contenidos del directorio
2. Copiar uno de los archivos de dentro del FiUnamFS hacia tu sistema
3. Copiar un archivo de tu computadora hacia tu FiUnamFS
4. Eliminar un archivo del FiUnamFS
5. Desfragmentar
6. Salir

Ingrese la opción: 1

Listando los contenidos del directorio...

README.org          24576  B      5 - 17    2022-05-08    13:57
prueba.pdf          311695 B     44 - 197  2022-05-19    23:29
logo.png            52200  B     17 - 43   2022-05-08    13:57
mensajes.png        158520 B    353 - 431  2022-05-08    13:57

```

Figura 4: Copia de un archivo de la computadora al sistema.

```

Ingrese la opción: 3

Ingrese el nombre del archivo a copiar:   hola.txt

El archivo indicado no existe.

```

Figura 5: Impresión de un archivo que no existe.

## 2.5. Eliminar un archivo

Para eliminar un archivo primero se pide el nombre del archivo a eliminar. También se llama a la función que devuelve la lista de archivos existentes para después recorrer esta lista y encontrar el archivo a eliminar donde se busca obtener el cluster donde empieza. Se vuelve a emplear la metodología de asignarle un valor inicial de -1 a esta variable para saber cuando no se encuentra un archivo y terminar la función.

En caso de que si se encuentre el archivo a eliminar se obtiene su tamaño y su cluster inicial con lo que se posiciona el archivo del sistema al inicio del contenido del archivo a eliminar y se escribe el caracter 00x en cada byte ocupado por el archivo. También se actualiza el directorio guardando primero la cadena de registro vacío en una variable. Se recorre el directorio hasta encontrar un registro cuyo nombre coincida con el indicado por el usuario. En ese caso, se coloca el archivo del sistema en esta dirección y se sustituyen los 64 bytes del archivo por esta “cadena vacía”. La ejecución de esta función se muestra en la Figura 6.

```

BarreiroAlejandro-ZepedaJessica — Python proyecto4-so.py — 123x47

Ingrese la opción: 1

Listando los contenidos del directorio...

README.org          24576  B      5 - 17    2022-05-08    13:57
prueba.pdf          311695  B     44 - 197  2022-05-19    23:50
logo.png             52200  B      17 - 43   2022-05-08    13:57
codigo.asm           1251  B     198 - 199  2022-05-19    23:51
proyecto4.py         18477  B     280 - 286  2022-05-19    23:52
mensajes.png         158520  B    353 - 431  2022-05-08    13:57

-----Menú de opciones-----

Selecciona una opción:
1. Listar los contenidos del directorio
2. Copiar uno de los archivos de dentro del FiUnamFS hacia tu sistema
3. Copiar un archivo de tu computadora hacia tu FiUnamFS
4. Eliminar un archivo del FiUnamFS
5. Desfragmentar
6. Salir

Ingrese la opción: 4

Ingrese el nombre del archivo a eliminar:   proyecto4.py

Archivo eliminado.

-----Menú de opciones-----

Selecciona una opción:
1. Listar los contenidos del directorio
2. Copiar uno de los archivos de dentro del FiUnamFS hacia tu sistema
3. Copiar un archivo de tu computadora hacia tu FiUnamFS
4. Eliminar un archivo del FiUnamFS
5. Desfragmentar
6. Salir

Ingrese la opción: 1

Listando los contenidos del directorio...

README.org          24576  B      5 - 17    2022-05-08    13:57
prueba.pdf          311695  B     44 - 197  2022-05-19    23:50
logo.png             52200  B      17 - 43   2022-05-08    13:57
codigo.asm           1251  B     198 - 199  2022-05-19    23:51
mensajes.png         158520  B    353 - 431  2022-05-08    13:57

```

Figura 6: Eliminación de un archivo del sistema.

## 2.6. Desfragmentar

Para la desfragmentación se siguió la lógica de ir recorriendo los archivos hacia “delante” para eliminar espacios vacíos entre archivos. Se obtiene primero la lista de archivos existentes y se ordena según su cluster inicial. Después se recorre la lista analizando dos archivos: se compara si entre el archivo siguiente y el actual existe un espacio de dos o más clusteres. Si esto se cumple, se lee todo el archivo siguiente y se guarda en una variable. Se “borra” todo el espacio del contenido de este archivo siguiente (con el caracter 00x) y luego se escribe el contenido después del cluster final del archivo actual.

El mover un archivo implica actualizar la información de su cluster inicial y de su fecha de modificación en el directorio. Por lo que se hace una subcadena con estos datos que se escribe en la posición correspondiente al archivo que se movió en el directorio. Para ello se vuelve a recorrer el directorio y se compara el nombre de un cierto registro con el del archivo que se está moviendo. La ejecución de esta función se muestra en la Figura 7.

```
Ingrese la opción: 5
Desfragmentado.
-----Menú de opciones-----
Selecciona una opción:
1. Listar los contenidos del directorio
2. Copiar uno de los archivos de dentro del FiUnamFS hacia tu sistema
3. Copiar un archivo de tu computadora hacia tu FiUnamFS
4. Eliminar un archivo del FiUnamFS
5. Desfragmentar
6. Salir
Ingrese la opción: 1
Listando los contenidos del directorio...
README.org      24576 B      5 - 17      2022-05-08      13:57
prueba.pdf      311696 B     44 - 197     2022-05-19      23:50
logo.png         52200 B      17 - 43      2022-05-08      13:57
codigo.asm       1251 B      198 - 199     2022-05-19      23:51
mensajes.png     158520 B     200 - 278     2022-05-08      13:57
```

Figura 7: Desfragmentación del sistema de archivos.

## 3. Requisitos del programa

Se utilizó Python 3.6.8 para el desarrollo de este código. El código se ejecutó con esta versión de Python utilizando el comando *python3*. Se utiliza dicho comando y el nombre del archivo a ejecutar. Posteriormente, el programa debe desplegar la información del sistema de archivos y un menú.

## 4. Traslape entre archivos README.org y logo.png

Como se aprecia en la Figura 2 el cluster final del README.org se encuentra después de que empieza el archivo logo.png por lo que hubo una sobreescritura desde el cluster 17 hasta



el 20. Esto se comprobó al copiar el archivo README.org a la computadora ya que se observa que al final comienza un png, como muestra las Figuras 8 y 9. Para arreglar esto y no tener

```
Ingrese la opción: 2

Ingrese el nombre del archivo nuevo:   pruebareadme.md
Ingrese el nombre del archivo a copiar: README.org

Archivo copiado.
```

Figura 8: Copia del archivo README.org a la computadora.

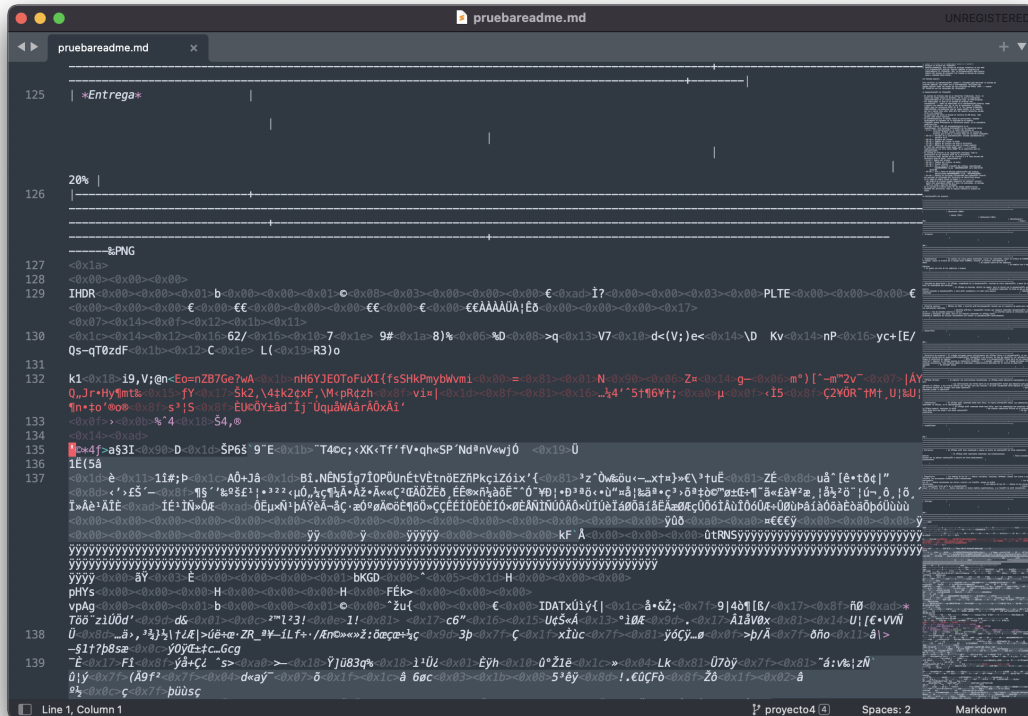


Figura 9: Final de la copia del archivo README.org.

problema al insertar y eliminar archivos se cambió el tamaño del archivo README.org en el directorio. Se indica un tamaño que termina en el cluster inicial de logo.png. Las instrucciones a estos cambios se encuentran “escondidas” en la opción 7 del menú. De manera que si se tiene una nueva imagen del disco con ejecutar esta opción, el sistema se vuelve consistente.

## Referencias

Wolf, G. (2022). Proyecto 4: (micro) sistema de archivos. Descargado de <https://github.com/unamfi/sistop-2022-2/tree/main/proyectos/4>