



UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO
FACULTAD DE INGENIERÍA



PROYECTO 1

”(Micro) sistema de archivos multihilos”

Autor: Pineda Galindo Ricardo Angel

Profesor: Ing. Gunnar Eyal Wolf Iszaevich

Asignatura: Sistemas Operativos

Grupo: 06

Fecha de entrega: 19 de mayo de 2024

Tabla de Contenidos

Requisitos	3
Estrategia	3
¿Cómo usarlo?	5
Función 1. Listar los contenidos del directorio.....	5
Función 2. Copiar un archivo del sistema FiUnamFS hacia tu computadora	6
Función 3. Copiar un archivo de tu computadora hacia el sistema FiUnamFS.....	6
Función 4. Eliminar un archivo del FiUnamFS	6
Función 5. Mostrar información del sistema	6
Función 6. Salir del programa	6
Sincronización de hilos	6
Ejemplos de uso	7
Función 1. Listar los contenidos del directorio.....	8
Función 2. Copiar un archivo del sistema FiUnamFS hacia tu computadora	8
Función 3. Copiar un archivo de tu computadora hacia el sistema FiUnamFS.....	10
Función 4. Eliminar un archivo del FiUnamFS	12
Función 5. Mostrar información del sistema	13
Función 6. Salir del programa	13
Entorno de Desarrollo	14

Requisitos

- Desarrollar un programa que pueda obtener, crear y modificar información en el micro-sistema-de-archivos de la Facultad de Ingeniería, FiUnamFS.

El programa debe poder realizar las siguientes funciones.

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. El programa que desarrollen debe contar, por lo menos, dos hilos de ejecución, operando concurrentemente, y que se comuniquen su estado mediante mecanismos de sincronización.

Estrategia

Para poder desarrollar este proyecto, lo primero que se realizó fue comprender las especificaciones propuestas. Entender que el sistema es un archivo de longitud fija, el cual ocupa cierta codificación (ASCII de 8 bytes) para las cadenas de texto y los números enteros tienen cierto formato (Little endian) para guardarse.

Una parte muy importante es entender cómo se puede trabajar con el espacio para guardar la información, es decir, el sistema se divide en clusters que están compuestos por 4 sectores de 512 bytes. Con esto se puede conocer el razonamiento para poder trabajar con los archivos al momento de copiarlos del sistema a la computadora o al revés sin corromper la información o no obtener toda la información del archivo deseado en el sistema.

Una vez que esto queda claro hasta cierto punto, la estrategia era sencilla, desarrollar primero de forma secuencial las funciones que solo leen la imagen, después las que escriben en la imagen y finalmente buscar la forma de implementar los hilos. Siempre tomando en cuenta que se hagan funciones con tareas sencillas para poder reutilizar el código en otras funciones.

Después de esta investigación en la fase inicial, sigue la fase de desarrollo, lo primero que se realizó fueron las funciones para poder leer bytes dentro de la imagen, tanto del directorio como de los datos. Fue necesario tomar en cuenta la codificación, el formato de los números enteros, y el desplazamiento de bytes dependiendo la situación, porque en el caso de los datos, se usa como referencia el cluster desde el cual se almacena la información.

Luego de poder validar la información del sistema, leer las entradas del directorio y convertir las cadenas o números enteros a su formato adecuado para su lectura, siguió la función para mostrar los archivos. Para esto, decidí crear otra función que lee el directorio y almacena la información en un diccionario donde se guarda el nombre como la llave y el valor es el resto de la información de la entrada del archivo. Esto se invoca siempre antes de realizar alguna de las funciones solicitadas donde se modifica la información de la imagen, para que se trabaje con la información actualizada.

En esta etapa de desarrollo me concentré en todo lo necesario con las funciones de lectura de archivos, para que posteriormente trabajara con las funciones que modifican o acceden a la imagen y así, poder estar seguro de que, en caso de haber algún error, no fuera por la forma en que se accede a la imagen o como se leen los bytes.

Para poder copiar archivos del sistema a la computadora primero se debían hacer algunas validaciones: escoger como opción solo alguno de los archivos que se imprimen en pantalla y que posteriormente se valide que la ruta destino exista para poder guardar ahí el archivo.

Para crear la siguiente función que borra los archivos del sistema, solo fue necesario que se escogiera el archivo a eliminar y después simplemente escribir ciertas cadenas de caracteres y números enteros en la entrada de directorio y los bytes que contienen la información del archivo para que estos datos basura no lleguen a interferir en la lectura de la imagen y puedan llegar a corromper un archivo. Cabe mencionar que esto también se podía manejar como simplemente borrarlo del directorio y perder referencia a los datos para que no se realice trabajo “extra” de volver a escribir en la imagen solo para sobrescribir todo, pero por comodidad decidí la primera opción.

Para la función de borrar archivos tuve que crear otras dos funciones que me permiten escribir en el directorio o en el espacio de datos, pasándoles ciertos parámetros que me permiten ubicar el apuntador en el lugar correcto para que pueda escribir. Gracias a esto pude aprovechar esas funciones para la siguiente tarea que es copiar un archivo de la computadora al sistema de archivos.

Para poder hacer eso, fue necesario hacer algunas validaciones para confirmar que exista el archivo a copiar, el tamaño del archivo quepa en el espacio de datos, que cumpla con la condición de que el nombre debe ser máximo de 14 bytes y que tenga la codificación ASCII de 7 bits para que no tenga caracteres especiales como los acentos. Una vez que se pasaron estos filtros, solo bastaba extraer información del archivo como las fechas de creación y modificación, el tamaño y el nombre del archivo para posteriormente escribirlos en la imagen.

Esto fue la parte sencilla, porque lo difícil fue asignar el espacio necesario para guardar sus datos e información o que en caso de que hubiera espacio suficiente pero no estuviera contiguo, tener que compactar los datos para ver si ya se podía guardar. Menciono que esto fue lo más difícil porque inicialmente estaba tomando en cuenta solo los bytes para poder guardar la información, en lugar de que usara los clusters como referencia para empezar a guardar la información. Las consecuencias de esto fue que cuando intentaba copiar un archivo a mi computadora, en algunas situaciones donde los archivos se guardan cerca, tenían el mismo cluster inicial y eso hacía que uno de los archivos se corrompiera porque se combinaba la información.

Después de varias horas de análisis, me di cuenta de que el error no fue trabajar desde un inicio con clusters para evitar la corrupción de datos y que los archivos no se llegaran a encimar por decirlo de alguna manera.

Respecto a la forma en que busca si hay espacio en el sistema, lo maneje de la siguiente manera: Primero se revisa que haya espacio suficiente para guardar los datos entre dos archivos ya existentes, si esto no se cumple, entonces busca en los extremos (en el inicio

del cluster 5 que es donde se empiezan a guardar los datos y en el siguiente cluster después del último archivo). Para realizar esto tuve que hacer que después de crear el diccionario con los archivos disponibles, se ordenara con base al cluster inicial para que las revisiones fueran de forma ascendente y así fuera más sencillo de leer en el código.

En esta función de copiar archivos de la computadora fue donde le dediqué más tiempo porque fue ahí donde me di cuenta de los errores al no manejar los clusters y su importancia al momento de leer los datos de la imagen.

¿Cómo usarlo?

Usar el programa es muy sencillo, primero debes indicar la ruta de la imagen del sistema de archivos, después se despliega un menú donde se presentarán las opciones para ver los archivos que contiene el sistema, copiar un archivo a tu computadora, copiar un archivo de tu computadora al sistema o borrar un archivo del sistema.

Para poder interactuar con el menú y la mayoría de las funciones, debes ingresar números para escoger la opción de tu preferencia. Si te llegases a equivocar en el menú principal o en cualquier opción, no te preocupes, el programa siempre te va a notificar que es lo que estás haciendo mal y posteriormente puedes corregirlo al acceder nuevamente a esa opción.

```
Manejador del Sistema de Archivos

1. Listar los contenidos del directorio
2. Copiar un archivo del sistema FiUnamFS hacia tu computadora
3. Copiar un archivo de tu computadora hacia el sistema FiUnamFS
4. Eliminar un archivo del FiUnamFS
5. Mostrar información del sistema
6. Salir del programa

Escribe el número asociado a la opción que deseas acceder: █
```

1. Menú principal del programa

Función 1. Listar los contenidos del directorio

Para esta opción solo debes ingresar el número 1, presionar enter y posteriormente se desplegarán los archivos y su información en una tabla para que los puedas apreciar de forma más cómoda y sencilla. Para salirte de esa opción bastará con que presiones la tecla enter y el programa te regresará al menú principal.

Función 2. Copiar un archivo del sistema FiUnamFS hacia tu computadora

Para esta opción debes ingresar el número 2, presionar enter y después se mostrarán los archivos disponibles en el sistema para poder copiarlos a tu computadora. En este punto debes ingresar el número asociado al archivo que deseas copiar y después poner la ruta donde quieras que se guarde. Si no hubo algún problema, te aparecerá un mensaje notificándote que se copió el archivo con éxito y deberás presionar enter para regresar al menú.

En caso de que, si hubiera algún problema, el programa te va a decir cuál fue el motivo del fallo y si está dentro de tus manos solucionarlo, puedes volver a intentarlo para que ahora el resultado sea positivo.

Función 3. Copiar un archivo de tu computadora hacia el sistema FiUnamFS

Para acceder a esta opción debes ingresar el número 3 y presionar enter, después te aparecerá un mensaje que te solicita la ruta del archivo a copiar y si no hay algún problema, te aparecerá un mensaje diciéndote que se copió el archivo con éxito. Al igual que con la función anterior, si hay alguna falla, te lo notificará el programa y después te regresará al menú principal.

Función 4. Eliminar un archivo del FiUnamFS

Esta función es similar a la función 2, seleccionas el archivo que quieras borrar, pulsas enter y si todo salió bien te aparecerá un mensaje diciendo que se borró con éxito.

Para acceder a la función debes ingresar el número 4 en el menú principal y después presionar enter.

Función 5. Mostrar información del sistema

Ingrese el número 5 en el menú principal y después presione enter.

Lo que aparecerá en pantalla es información exclusiva del sistema de archivos para poder ver el espacio de almacenamiento, su nombre y otras cosas. Para salir solo debe presionar enter y regresará al menú principal.

Función 6. Salir del programa

En caso de que quiera finalizar el programa, solo debe ingresar el número 6, presionar enter y ya podrá cerrar el programa.

Sincronización de hilos

Esta fue una de las partes difíciles pero interesantes de razonar para poder implementarlas de forma correcta.

Inicialmente la idea fue utilizar el patrón Jefe/Trabajador para que los archivos que necesitaran leer o escribir en el directorio y el espacio de datos se pudiera hacer a la par, pero esto ya no se concretó porque resultó complicado ajustar el código en todas las funciones donde se utilizan estas funciones.

Por lo que la opción más sencilla fue implementar un hilo para cada función principal del programa y que estos realicen todo lo de su función de forma secuencial, pero manteniendo comunicación con el hilo principal para que solo trabaje cuando sea necesario. La idea era implementar el patrón Rendezvous entre el hilo principal y cada uno de los hilos de las otras funciones, es decir, el hilo principal se iba a comunicar con cada hilo de forma particular solo cuando se requiriera esa función.

En el primer intento se usaron 4 semáforos, uno para el hilo principal, y los demás para su respectiva función. El funcionamiento solo constaba de que el main liberara el semáforo para un hilo y después se bloqueaba el semáforo del main para que el otro hilo hiciera su función, y una vez que terminara, liberaba el semáforo del hilo principal y después se bloqueaba nuevamente.

El problema con esto es que se liberaban los semáforos una vez que ya terminó de hacer toda su función, por lo que al ejecutar el programa se imprimían cosas que todavía no debían ser invocadas. Para evitar esto tuve que hacer que al inicio de la función se bloqueara el hilo hasta que fuera liberado por el hilo principal. El detalle con esto es que para realizar cualquier función 1 sola vez, corrían bien, pero en la segunda vez se bloqueaba el programa porque al liberar el semáforo de algún hilo se volvía a bloquear por la instrucción que bloqueaba el hilo al inicio para evitar que hiciera algo cuando aun no era su momento. La solución a esto no la pude encontrar utilizando los semáforos, por lo que mejor decidí crear un ciclo infinito para cada hilo donde al inicio se bloqueara y después de realizar su función liberara el semáforo del hilo principal para que al reiniciar el ciclo se volviera a bloquear solo. En esta nueva forma ya solo era necesario que un hilo liberara el semáforo del otro hilo y ya después se bloqueara en automático.

Desde el punto de vista de la función main si se ejecuta el patrón al pie de la letra, pero en el caso de las otras funciones se puede ver un poco diferente por la distancia que hay entre las instrucciones que liberan o bloquean los semáforos, pero es la misma idea, en el final se libera el semáforo del hilo principal después de haber realizado todo su trabajo y después se bloquea para volver a esperar en su punto de encuentro.

Ejemplos de uso

Inicialmente el programa va a solicitar ingresar la dirección de la imagen del sistema de archivos.

```
Ingrese la ruta donde se encuentra la imagen del Sistema de Archivos:
```

En caso de que no encuentre la imagen, te va a mostrar el siguiente mensaje

```
Ingrese la ruta donde se encuentra la imagen del Sistema de Archivos: opkp  
Error 5: La ruta proporcionada no existe, revisa que la hayas escrito bien.  
Ingrese la ruta donde se encuentra la imagen del Sistema de Archivos: 
```

Este mensaje no va a dejar de aparecer hasta que se ponga la ruta correcta y posteriormente te va a mandar al menú principal para poder manipular el sistema de archivos.

Para acceder a la primera opción se escribirá el número 1.

Función 1. Listar los contenidos del directorio

	Nombre	Tamaño(Bytes)	Cluster Inicial	Fecha de Creación	Fecha de Modificación
0	README.org	31222	6	2024/05/08 13:17:05	2024/05/08 13:17:05
1	logo.png	126423	22	2024/05/08 13:17:05	2024/05/08 13:17:05
2	mensaje.jpg	254484	84	2024/05/08 13:17:05	2024/05/08 13:17:05

PRESIONE ENTER PARA CONTINUAR

2. Función 1 - Listar contenidos

Como se puede observar, se muestra una tabla con la información esencial de cada archivo. Para regresar al menú basta con presionar enter.

Función 2. Copiar un archivo del sistema FiUnamFS hacia tu computadora

Ahora en el menú principal se presiona el número 2.

	Nombre	Tamaño(Bytes)	Cluster Inicial	Fecha de Creación	Fecha de Modificación
0	README.org	31222	6	2024/05/08 13:17:05	2024/05/08 13:17:05
1	logo.png	126423	22	2024/05/08 13:17:05	2024/05/08 13:17:05
2	mensaje.jpg	254484	84	2024/05/08 13:17:05	2024/05/08 13:17:05

Escribe el número del archivo que deseas copiar a tu computadora:

3. Función 2 - Copiar archivos a tu computadora

En general, todos los menús con opciones solicitan un número para escoger una opción y en caso de que te equivoques, te lo hace saber el programa y se regresa al menú principal.


```
Escribe el número del archivo que desees copiar a tu computadora: 4
Error 10: La opción seleccionada no pertenece a ningún archivo
```

```
PRESIONE ENTER PARA CONTINUAR
```

4. Ejemplo de error

Para este ejemplo se escogerá el archivo mensaje.jpg poniendo el número 2 y la ruta donde se almacenará.

```
Escribe el número del archivo que desees copiar a tu computadora: 2
```







```
Escribe la ruta hacia donde quieras guardar el archivo: D:\UNIVERSIDAD\Sexto Semestre\S0\proyecto
```

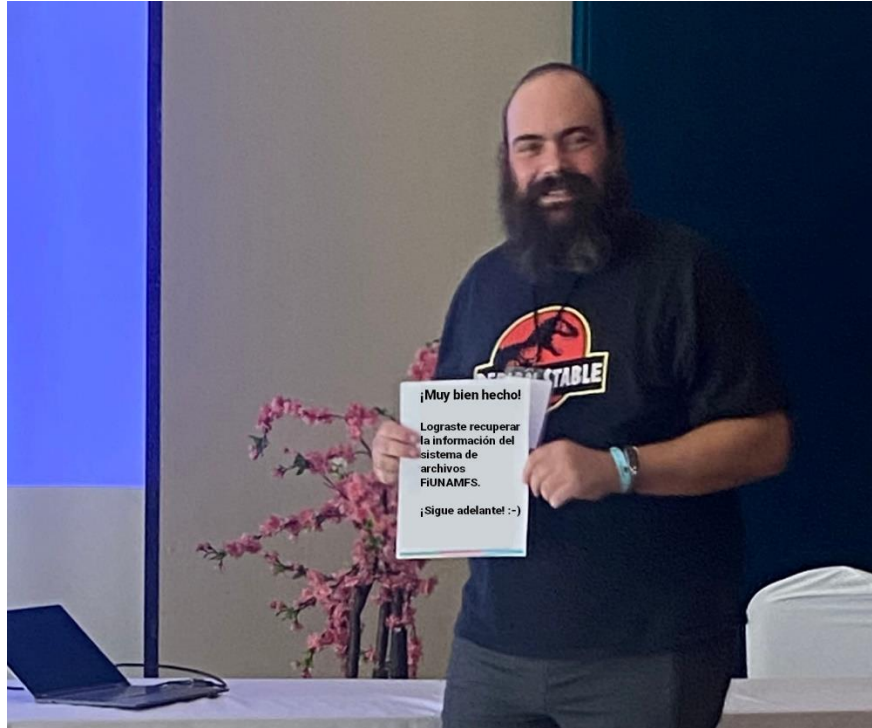
Si esta ruta existe, aparecerá un mensaje confirmando la acción y si no existe, aparecerá un error diciendo el motivo.

```
El archivo se ha copiado con éxito!!
```

```
PRESIONE ENTER PARA CONTINUAR
```

5. Ejemplo de error

	.vscode	12/05/2024 04:45 p. m.	Carpeta de archivos	
	.gitignore	12/05/2024 10:29 p. m.	Archivo de origen ...	1 KB
	fiunamfs.img	12/05/2024 10:30 p. m.	Archivo de image...	1,440 KB
	PinedaGalindoRicardo_proyecto1.py	16/05/2024 08:32 p. m.	Archivo de origen ...	23 KB
	Tarea 1_2.pdf	16/05/2024 07:50 p. m.	Microsoft Edge P...	1,013 KB
	mensaje.jpg	18/05/2024 02:33 p. m.	Archivo JPG	249 KB



6. Imagen sacada del sistema de archivos

Función 3. Copiar un archivo de tu computadora hacia el sistema FiUnamFS

Para esta función, lo único que solicita el programa es la ruta del archivo a copiar. En este caso se copiará un archivo llamado Tarea 1_2.pdf

```
Escriba la ruta del archivo que desea copiar al sistema de archivos:D:\UNIVERSIDAD\Sexto Semestre\S0\proyecto\Tarea 1_2.pdf
El archivo se copió con éxito!!

PRESIONE ENTER PARA CONTINUAR
```

7. Función 3 - Copiar archivos de computadora al sistema

Para comprobar esto, podemos volver a leer el contenido del sistema.

	Nombre	Tamaño(Bytes)	Cluster Inicial	Fecha de Creación	Fecha de Modificación
0	README.org	31222	6	2024/05/08 13:17:05	2024/05/08 13:17:05
1	logo.png	126423	22	2024/05/08 13:17:05	2024/05/08 13:17:05
2	mensaje.jpg	254484	84	2024/05/08 13:17:05	2024/05/08 13:17:05
3	Tarea 1_2.pdf	1036640	209	2024/05/15 13:37:02	2024/05/16 19:50:01

PRESIONE ENTER PARA CONTINUAR

Y como se puede observar, si se pudo ingresar con éxito. Para poder comprobar esto de otra forma, se copiará el archivo que se guardó en el sistema a una ruta de la computadora. Si se copió de forma correcta, se debería poder ver el contenido del pdf.







	Nombre	Tamaño(Bytes)	Cluster Inicial	Fecha de Creación	Fecha de Modificación
0	README.org	31222	6	2024/05/08 13:17:05	2024/05/08 13:17:05
1	logo.png	126423	22	2024/05/08 13:17:05	2024/05/08 13:17:05
2	mensaje.jpg	254484	84	2024/05/08 13:17:05	2024/05/08 13:17:05
3	Tarea 1_2.pdf	1036640	209	2024/05/15 13:37:02	2024/05/16 19:50:01

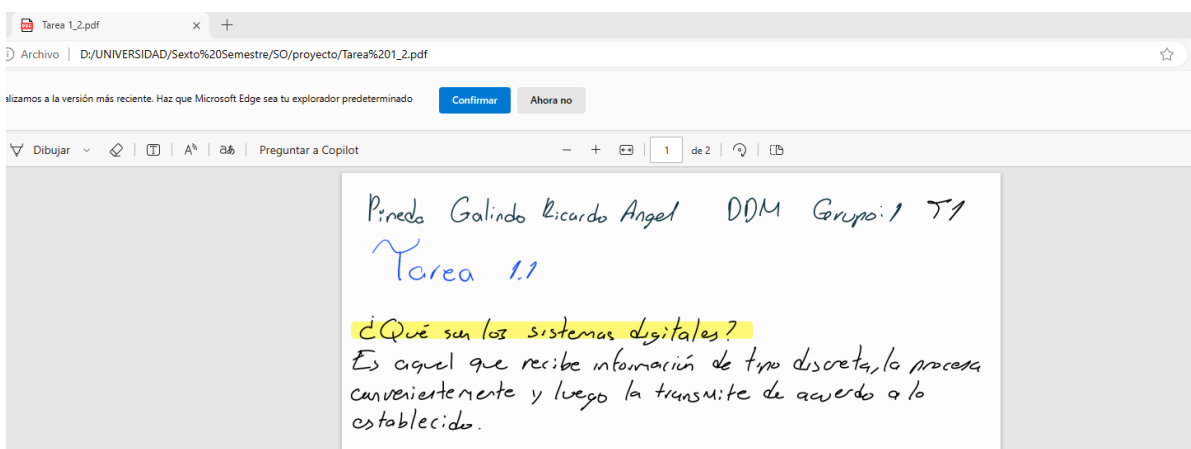
Escribe el número del archivo que deseas copiar a tu computadora: 3

Escribe la ruta hacia donde quieras guardar el archivo: D:\UNIVERSIDAD\Sexto Semestre\SO\proyecto
El archivo se ha copiado con éxito!!

PRESIONE ENTER PARA CONTINUAR

Como resultado se obtuvo lo siguiente:

	.vscode	12/05/2024 04:45 p. m.	Carpeta de archivos	
	.gitignore	12/05/2024 10:29 p. m.	Archivo de origen ...	1 KB
	fiunamfs.img	18/05/2024 02:35 p. m.	Archivo de image...	1,440 KB
	PinedaGalindoRicardo_proyecto1.py	16/05/2024 08:32 p. m.	Archivo de origen ...	23 KB
	mensaje.jpg	18/05/2024 02:33 p. m.	Archivo JPG	249 KB
	Tarea 1_2.pdf	18/05/2024 02:37 p. m.	Microsoft Edge P...	1,013 KB



8. Prueba de que se guardó un archivo en el sistema de archivos

En caso de que se hubiera puesto una ruta incorrecta, hubiera aparecido el siguiente mensaje:

```
Escribe el número del archivo que deseas copiar a tu computadora: 0

Escribe la ruta hacia donde quieras guardar el archivo: no_existe/aqui/
Error 5: La ruta proporcionada no existe, revisa que la hayas escrito bien.
Presiona enter para continuar...█
```

9. Ejemplo de error

Función 4. Eliminar un archivo del FiUnamFS

Aquí solamente se va a solicitar el archivo que quieras borrar y pulsar enter. En este caso se borrará el archivo llamado mensaje.jpg

	Nombre	Tamaño(Bytes)	Cluster Inicial	Fecha de Creación	Fecha de Modificación
0	README.org	31222	6	2024/05/08 13:17:05	2024/05/08 13:17:05
1	logo.png	126423	22	2024/05/08 13:17:05	2024/05/08 13:17:05
2	mensaje.jpg	254484	84	2024/05/08 13:17:05	2024/05/08 13:17:05
3	Tarea 1_2.pdf	1036640	209	2024/05/15 13:37:02	2024/05/16 19:50:01

```
Escribe el número del archivo que deseas borrar: 2
El archivo se ha eliminado con éxito!!

PRESIONE ENTER PARA CONTINUAR█
```

10. Función 4 - Eliminar archivos del sistema

Para comprobarlo debemos volver a leer los archivos del sistema.

	Nombre	Tamaño(Bytes)	Cluster Inicial	Fecha de Creación	Fecha de Modificación
0	README.org	31222	6	2024/05/08 13:17:05	2024/05/08 13:17:05
1	logo.png	126423	22	2024/05/08 13:17:05	2024/05/08 13:17:05
2	Tarea 1_2.pdf	1036640	209	2024/05/15 13:37:02	2024/05/16 19:50:01

Si hubiéramos seleccionado una opción que no corresponde a algún archivo, hubiera aparecido el siguiente mensaje:

```
Escribe el número del archivo que deseas borrar: 5
Error 10: La opción seleccionada no pertenece a ningún archivo
```

11. Ejemplo de error

Función 5. Mostrar información del sistema

Esta función va a recuperar cierto conjunto de bytes que contiene la imagen y la va a mostrar en pantalla, además de que mostrará información relacionada con el espacio disponible y archivos almacenados.

```
INFORMACIÓN DEL SISTEMA DE ARCHIVOS

Nombre: FiUnamFS
Versión: 24-2
Etiqueta del volumen: Mi Sistema Favorito
Tamaño del cluster: 2048 bytes
Tamaño del directorio: 4 clusters
Tamaño de la unidad completa: 720 clusters
Entradas Ocupadas: 3      || Entradas Disponibles: 125
Espacio Disponible: 15551 Bytes  || Espacio Ocupado: 1194285 Bytes

PRESIONE ENTER PARA CONTINUAR
```

12. Función 5 - Mostrar información del sistema

En este caso, solo se han almacenado 3 archivos, pero como Tarea 1_2.pdf está muy pesado ocupa casi todo el espacio.

Función 6. Salir del programa

Finalmente, para poder cerrar el programa de forma correcta, bastará con ingresar el número 6.

```
Cerrando Sistema de Archivos...
PS C:\Users\ricki>
```

13. Fin del programa

Entorno de Desarrollo

El lenguaje de programación utilizado fue Python, debido a su disponibilidad respecto a todos los módulos necesarios para poder desarrollar el proyecto. Inicialmente se planeaba utilizar el lenguaje C pero como no se logró encontrar las cabeceras para poder leer los bytes, se optó por cambiarlo a Python.

1. Lenguaje de Programación

- Lenguaje: Python
- Implementación: CPython
- Versión Mayor: Python 3.11.2

2. Principales Módulos y sus Versiones

Aquí se enumeran los módulos o paquetes de Python que son esenciales para que el programa funcione correctamente, algunos módulos tienen su versión declarada y otras solo indican que son compatibles con versiones de Python mayoreso igual a 3.

- os: 3.7
- datetime
- math
- tabulate: 0.9.0
- threading

3. Funciones principales utilizadas:

- path() del módulo os
- datetime() del módulo datetime
- ceil() del módulo math
- tabulate() del módulo tabulate
- Semaphore() y Thread del módulo threading
- Decode del modulo Bytes
- from_bytes y to_bytes del modulo int
-

En todos los módulos, solo se utilizan funciones específicas, por lo que en el programa solo se importan algunas funciones.

Para que el programa opere de forma correcta, se recomienda actualizar Python a su versión más reciente para evitar eventos que interfieran con el comportamiento normal.