Universidad Nacional Autónoma de México

Facultad de Ingeniería

Sistemas operativos

Grupo 6

Semestre 2024-1

Proyecto 3 “(Micro) Sistema de archivos”

Docente:

Ing. Gunnar Eyal Wolf Iszaevic

Alumnos:

- Ceniceros Mariaca Carlos
- Ramirez Martinez Luis Angel

Objetivos

Siguiendo las siguientes especificaciones que aparecen en la siguiente sección, se tiene que desarrollar un programa que pueda:

- Listar los contenidos del directorio
- Copiar uno de los archivos de dentro del FiUnamFS hacia tu sistema
- Copiar un archivo de tu computadora hacia tu FiUnamFS
- Eliminar un archivo del FiUnamFS
- Desfragmentación de FiUnamFS

Desarrollo

Para desarrollar este proyecto se hizo uso del lenguaje de programación Python y se implementaron las siguientes librerías:

```python
import os
import struct
from datetime import *
import io
```

Al tener cinco puntos que resolver se decidió dividir el problema en cinco funciones principales que explicaremos a continuación apoyándonos de una imagen del código además que se describirá brevemente su propósito y los parámetros que requiere la función.

Listar los contenidos del directorio

```python
def listar_archivos(fiunamfs_img_path, directorio_inicio, directorio_tamano, entrada_directorio_tamano):
    with open(fiunamfs_img_path, 'rb') as file:
        # Ir al inicio del directorio
        file.seek(directorio_inicio)

        # Leer el área completa del directorio
        directorio = file.read(directorio_tamano)

        # Procesar cada entrada del directorio
        for i in range(0, directorio_tamano, entrada_directorio_tamano):
            entrada = directorio[i:i + entrada_directorio_tamano]
            nombre_archivo = entrada[1:15].decode().strip()

            # Saltar entradas vacías o con nombres no válidos
            if not nombre_archivo or nombre_archivo == "---------------" or nombre_archivo.startswith('.'):
                continue

            tamano_archivo, cluster_inicial = struct.unpack('<I', entrada[16:20])[0], struct.unpack('<I', entrada[20:24])[0]
            fecha_creacion, fecha_modificacion = entrada[24:38].decode().strip(), entrada[38:52].decode().strip()

            # Verificar si hay información relevante del archivo antes de imprimir
            if tamano_archivo > 0 and cluster_inicial > 0 and fecha_creacion and fecha_modificacion:
                print(f"ARCHIVO: {nombre_archivo}, TAMAÑO: {tamano_archivo}, "
                      f"CLUSTER INICIAL: {cluster_inicial}, "
                      f"FECHA CREACION: {fecha_creacion}, "
                      f"FECHA MODIFICACION: {fecha_modificacion}\n")
```

La función “leer_archivos” tiene como propósito listar los archivos presentes en el sistema de archivos FiUnamFS. Como parámetros para funcionar requiere:

- “fiunamfs_img_path”
- “directorio_inicio”
- “directorio_tamano”
- “entrada_directorio_tamano”

Copiar uno de los archivos de dentro del FiUnamFS hacia tu sistema

```
def copiar_a_sistema_local(fiunamfs_img_path, directorio_inicio, directorio_tamano, entrada_directorio_tamano, tamano_cluster):
    print("\nLOS ARCHIVOS EXISTENTES SON:")
    listar_nombres_archivos(fiunamfs_img_path, directorio_inicio, directorio_tamano, entrada_directorio_tamano)
    nombre_archivo_destino = input("\nINGRESE EL NOMBRE DEL ARCHIVO A COPIAR: ")
    ruta_destino = input("INGRESE LA RUTA DE DESTINO EN SU SISTEMA *INCLUYENDO EL NOMBRE DEL ARCHIVO*: ")

    with open(fiunamfs_img_path, 'rb') as fiunamfs:
        fiunamfs.seek(directorio_inicio)
        directorio = fiunamfs.read(directorio_tamano)
        archivo_encontrado = False

        for i in range(0, directorio_tamano, entrada_directorio_tamano):
            entrada = directorio[i:i + entrada_directorio_tamano]
            nombre_archivo = entrada[1:15].decode().strip()
            tamano_archivo, cluster_inicial = struct.unpack('<I', entrada[16:20])[0], struct.unpack('<I', entrada[20:24])[0]

            if nombre_archivo == nombre_archivo_destino and tamano_archivo > 0:
                archivo_encontrado = True
                break

        if not archivo_encontrado:
            print(f"No se encontró el archivo '{nombre_archivo_destino}' en FiUnamFS.")
            return

        posicion_inicial = cluster_inicial * tamano_cluster
        fiunamfs.seek(posicion_inicial)
        datos_archivo = fiunamfs.read(tamano_archivo)

        if len(datos_archivo) != tamano_archivo:
            print("Error: El tamaño de los datos leídos no coincide con el tamaño del archivo.")
            return

    with open(ruta_destino, 'wb') as archivo_local:
        archivo_local.write(datos_archivo)

    print(f"\nARCHIVO '{nombre_archivo_destino}' COPIADO CON EXITO '{ruta_destino}'.")
```

La función “copiar_a_sistema_local” copia un archivo desde el sistema de archivos FiUnamFS al sistema local. Como parámetro requiere:

- “fiunamfs_img_path”
- “directorio_inicio”
- “directorio_tamano”
- “entrada_directorio_tamano”
- “tamano_cluster”

Copiar un archivo de tu computadora hacia tu FiUnamFS

```python
def copiar_a_fiunamfs_con_sobrescritura(fiunamfs_img_path, directorio_inicio, directorio_tamano, entrada_directorio_tamano, tamano_cluster, numer
    archivo_sistema = input("\n INGRESE LA RUTA DEL ARCHIVO COMPLETA A COPIAR: \n")
    if not os.path.exists(archivo_sistema):
        return "\n\tERROR: No fue posible encontrar la ruta"

    try:
        with open(archivo_sistema, "rb") as file:
            nombre = os.path.basename(archivo_sistema)
            if len(nombre) > 14:
                return "\tERROR: El nombre del archivo es demasiado largo para el sistema."

            tam = os.path.getsize(archivo_sistema)
            if tam > (numero_maximo_clusters - 5 - 1) * tamano_cluster:  # Ajuste aquí
                return "\tERROR: El tamaño del archivo es demasiado grande para el sistema."
            contenido = file.read()

        with open(fiunamfs_img_path, 'rb+') as fiunamfs:
            fiunamfs.seek(directorio_inicio)
            directorio = fiunamfs.read(directorio_tamano)

            for i in range(0, directorio_tamano, entrada_directorio_tamano):
                entrada = directorio[i:i + entrada_directorio_tamano]
                nombre_archivo = entrada[1:15].decode().strip()
```

```python
                if nombre_archivo in ["--------------", "."*14]:
                    # Escribir el archivo en esta posición
                    cluster_destino = 5 + (i // entrada_directorio_tamano)  # Calcula el cluster de destino
                    posicion_inicial = cluster_destino * tamano_cluster
                    fiunamfs.seek(posicion_inicial)
                    fiunamfs.write(contenido)

                    # Actualizar la entrada del directorio
                    fecha_modificacion = str(datetime.fromtimestamp(os.path.getmtime(archivo_sistema)))[0:19].replace("-","").replace(" ","").re
                    fecha_creacion = str(datetime.fromtimestamp(os.path.getctime(archivo_sistema)))[0:19].replace("-","").replace(" ","").replac
                    nueva_entrada = ('-' + nombre.ljust(14)).encode() + struct.pack('<I', tam) + struct.pack('<I', cluster_destino) + fecha_creac
                    nueva_entrada = nueva_entrada.ljust(entrada_directorio_tamano, b'\x00')
                    fiunamfs.seek(directorio_inicio + i)
                    fiunamfs.write(nueva_entrada)

                    return "ARCHIVO COPIADO EXITOSAMENTE EN FIUNAMFS."

            return "No hay espacio suficiente en FiUnamFS."

    except Exception as e:
        return f"\tERROR: No fue posible abrir el archivo (Error: {e})"
```

La función “copiar_a_fiunamfs_con_sobrescritura” copia un archivo del sistema local al FiUnamFS, con opción de sobrescribir un archivo existente si es necesario. Como parámetros requiere:

- “fiunamfs_img_path”
- “directorio_inicio”
- “directorio_tamano”
- “entrada_directorio_tamano”
- “tamano_cluster”
- “numero_maximo_clusters”

Eliminar un archivo del FiUnamFS

```python
def eliminar_archivo_fiunamfs(fiunamfs_img_path, directorio_inicio, directorio_tamano, entrada_directorio_tamano):
    print("\nLOS ARCHIVOS EXISTENTES SON:\n")
    listar_archivos(fiunamfs_img_path, directorio_inicio, directorio_tamano, entrada_directorio_tamano)
    nombre_archivo_eliminar = input("\nINGRESE EL NOMBRE DEL ARCHIVO A ELIMINAR: \n")

    with open(fiunamfs_img_path, 'rb+') as fiunamfs:
        # Leer el directorio
        fiunamfs.seek(directorio_inicio)
        directorio = fiunamfs.read(directorio_tamano)

        archivo_encontrado = False

        # Buscar el archivo en el directorio
        for i in range(0, directorio_tamano, entrada_directorio_tamano):
            entrada = directorio[i:i + entrada_directorio_tamano]
            nombre_archivo = entrada[1:15].decode().strip()

            if nombre_archivo == nombre_archivo_eliminar:
                archivo_encontrado = True

                # Marcar la entrada del directorio como vacía
                entrada_vacia = b'/' + b' ' * 14 + b'\x00' * (entrada_directorio_tamano - 15)
                fiunamfs.seek(directorio_inicio + i)
                fiunamfs.write(entrada_vacia)
                break

        if not archivo_encontrado:
            return "Archivo no encontrado."

        return "ARCHIVO ELIMINADO CON EXITO."
```

La función “eliminar_archivo_fiunamfs” elimina un archivo específico del sistema de archivos FiUnamFS. Como parámetros requiere:

- “fiunamfs_img_path”
- “directorio_inicio”
- “directorio_tamano”
- “entrada_directorio_tamano”

Desfragmentación de FiUnamFS

```python
def desfragmentar_fiunamfs(fiunamfs_img_path, directorio_inicio, directorio_tamano, entrada_directorio_tamano, tamano_cluster):
    with open(fiunamfs_img_path, 'rb+') as fiunamfs:
        # Leer el directorio
        fiunamfs.seek(directorio_inicio)
        directorio = fiunamfs.read(directorio_tamano)

        archivos = []
        # Extraer información de cada archivo
        for i in range(0, directorio_tamano, entrada_directorio_tamano):
            entrada = directorio[i:i + entrada_directorio_tamano]
            nombre_archivo = entrada[1:15].decode().strip()
            tamano_archivo, cluster_inicial = struct.unpack('<I', entrada[16:20])[0], struct.unpack('<I', entrada[20:24])[0]

            if nombre_archivo and nombre_archivo != "--------------" and nombre_archivo != "..............." and tamano_archivo > 0:
                archivos.append((nombre_archivo, tamano_archivo, cluster_inicial, i))

        # Ordenar los archivos por su cluster inicial
        archivos.sort(key=lambda x: x[2])

        # Desfragmentar cada archivo
        posicion_actual = directorio_inicio + directorio_tamano
        for i in range(len(archivos) - 1):
            archivo_actual = archivos[i]
            archivo_siguiente = archivos[i + 1]
            fin_archivo_actual = archivo_actual[2] + (archivo_actual[1] + tamano_cluster - 1) // tamano_cluster
```

```
        if fin_archivo_actual < archivo_siguiente[2]:
            # Mover archivo siguiente para que esté contiguo al archivo actual
            posicion_lectura = archivo_siguiente[2] * tamano_cluster
            fiunamfs.seek(posicion_lectura)
            datos_archivo = fiunamfs.read(archivo_siguiente[1])

            posicion_escritura = fin_archivo_actual * tamano_cluster
            fiunamfs.seek(posicion_escritura)
            fiunamfs.write(datos_archivo)

            # Actualizar la entrada del directorio del archivo movido
            nueva_entrada = entrada[:16] + struct.pack('<I', fin_archivo_actual) + entrada[20:]
            indice_dir = archivo_siguiente[3]
            fiunamfs.seek(directorio_inicio + indice_dir)
            fiunamfs.write(nueva_entrada)

    return "Desfragmentación completada con éxito."
```

La función “desfragmentar_fiunamfs” realiza un proceso de desfragmentación en el sistema de archivos FiUnamFS. Como parámetros requiere:

- “fiunamfs_img_path”
- “directorio_inicio”
- “directorio_tamano”
- “entrada_directorio_tamano”
- “tamano_cluster”

Con esto el programa realiza lo mínimo requerido para funcionar, pero con fines de mejorar el funcionamiento y hacer su uso más fácil se crearon las siguientes funciones.

Leer superbloque

```
def leer_superbloque(fiunamfs_img_path):
    with open(fiunamfs_img_path, 'rb') as file:
        file.seek(0)
        superbloque = file.read(1024)  # Leyendo el primer cluster que es el superbloque

        # Extraer información del superbloque
        nombre_fs = superbloque[0:8].decode().strip('\x00')
        version = superbloque[10:15].decode().strip('\x00')

        if nombre_fs != "FiUnamFS":
            raise ValueError("El sistema de archivos no es FiUnamFS")
        if version != "24.1":
            raise ValueError("Versión del sistema de archivos no compatible")

        # Extraer valores del superbloque
        tamano_cluster = struct.unpack('<I', superbloque[40:44])[0]
        numero_clusters_directorio = struct.unpack('<I', superbloque[45:49])[0]
        numero_maximo_clusters = struct.unpack('<I', superbloque[50:54])[0]
        directorio_inicio = 1 * tamano_cluster
        directorio_tamano = numero_clusters_directorio * tamano_cluster
        entrada_directorio_tamano = 64  # Asumiendo que este valor es fijo

        return (version, tamano_cluster, directorio_inicio, directorio_tamano, entrada_directorio_tamano, numero_maximo_clusters)
```

La función “leer_superbloque” lee y analiza el superbloque del sistema de archivos FiUnamFS. Como parámetros requiere:

- “fiunamfs_img_path”

Algo muy importante de esta función son los valores que regresa ya que estos son requeridos para el resto de las funciones los cuales son los siguientes:

- “versión”
- “tamano_cluster”
- “directorio_inicio”
- “directorio_tamano”
- “entrada_directorio_tamano”
- “numero_maximo_clusters”

Y por último para mejorar el manejo del programa se creó un menú en la consola que nos ayuda a elegir y ejecutar la función que deseamos realizar sobre el sistema de archivos.

Menú

```python
def menu_fiunamfs(fiunamfs_img_path):
    version, tamano_cluster, directorio_inicio, directorio_tamano, entrada_directorio_tamano, numero_maximo_clusters = leer_superbloque(fiunamfs_

    while True:
        print("\n----MENÚ FIUNAMFS----")
        print("*** 1. LISTAR ARCHIVOS ***")
        print("*** 2. COPIAR ARCHIVO DE FiUnamFs AL SISTEMA ***")
        print("*** 3. COPIAR ARCHIVO DEL SISTEMA A FiUnamFS ***")
        print("*** 4. ELIMINAR ARCHIVO ***")
        print("*** 5. DESFRAGMENTAR ***")
        print("*** 6. MOSTRAR INFORMACION DEL SUPERBLOQUE ***")
        print("*** 7. SALIR ***")


        opcion = input("SELECCIONE UNA OPCION: ")
```

```python
        if opcion == '1':
            print("\n")
            listar_archivos(fiunamfs_img_path, directorio_inicio, directorio_tamano, entrada_directorio_tamano)
        elif opcion == '2':
            print("\n")
            copiar_a_sistema_local(fiunamfs_img_path, directorio_inicio, directorio_tamano, entrada_directorio_tamano, tamano_cluster)
        elif opcion == '3':
            print("\n")
            resultado = copiar_a_fiunamfs_con_sobrescritura(fiunamfs_img_path, directorio_inicio, directorio_tamano, entrada_directorio_tamano,
            print(resultado)
        elif opcion == '4':
            print("\n")
            resultado = eliminar_archivo_fiunamfs(fiunamfs_img_path, directorio_inicio, directorio_tamano, entrada_directorio_tamano)
            print(resultado)
        elif opcion == '5':
            print("Iniciando desfragmentación...")
            resultado = desfragmentar_fiunamfs(fiunamfs_img_path, directorio_inicio, directorio_tamano, entrada_directorio_tamano, tamano_cluste
            print(resultado)
        elif opcion == '6':
            print("\nInformación del Superbloque:")
            print(f"Versión del sistema de archivos: {version}")
            print(f"Tamaño del Cluster: {tamano_cluster} bytes")
            print(f"Inicio del Directorio: {directorio_inicio} bytes")
            print(f"Tamaño del Directorio: {directorio_tamano} bytes")
            print(f"Tamaño de Entrada del Directorio: {entrada_directorio_tamano} bytes")
            print(f"Número Máximo de Clusters: {numero_maximo_clusters}")
        elif opcion == '7':
            print("Saliendo del programa...")
            break
        else:
            print("Opción no válida. Por favor, intente de nuevo.")
```

Esta función es un menú simple que invoca las funciones que deseemos usar en base a un número que pongamos en la consola.

Evidencia de funcionamiento

Listar los contenidos del directorio

```
----MENÚ FIUNAMFS----
*** 1. LISTAR ARCHIVOS ***
*** 2. COPIAR ARCHIVO DE FiUnamFs AL SISTEMA ***
*** 3. COPIAR ARCHIVO DEL SISTEMA A FiUnamFS ***
*** 4. ELIMINAR ARCHIVO ***
*** 5. DESFRAGMENTAR ***
*** 6. MOSTRAR INFORMACION DEL SUPERBLOQUE ***
*** 7. SALIR ***
SELECCIONE UNA OPCION: 1


ARCHIVO: README.org, TAMAÑO: 31209, CLUSTER INICIAL: 6, FECHA CREACION: 20231116130339, FECHA MODIFICACION: 20231116130339

ARCHIVO: logo.png, TAMAÑO: 170192, CLUSTER INICIAL: 21, FECHA CREACION: 20231116130339, FECHA MODIFICACION: 20231116130339

ARCHIVO: mensaje.jpg, TAMAÑO: 102657, CLUSTER INICIAL: 154, FECHA CREACION: 20231116130339, FECHA MODIFICACION: 20231116130339
```

Copiar uno de los archivos de dentro del FiUnamFS hacia tu sistema

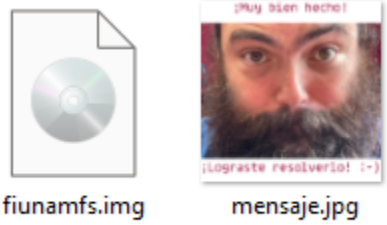
```
----MENÚ FIUNAMFS----
*** 1. LISTAR ARCHIVOS ***
*** 2. COPIAR ARCHIVO DE FiUnamFs AL SISTEMA ***
*** 3. COPIAR ARCHIVO DEL SISTEMA A FiUnamFS ***
*** 4. ELIMINAR ARCHIVO ***
*** 5. DESFRAGMENTAR ***
*** 6. MOSTRAR INFORMACION DEL SUPERBLOQUE ***
*** 7. SALIR ***
SELECCIONE UNA OPCION: 2



LOS ARCHIVOS EXISTENTES SON:
README.org
logo.png
mensaje.jpg

INGRESE EL NOMBRE DEL ARCHIVO A COPIAR: mensaje.jpg
INGRESE LA RUTA DE DESTINO EN SU SISTEMA *INCLUYENDO EL NOMBRE DEL ARCHIVO*: C:\Users\carlo\OneDrive\Escritorio\SO\Proyecto 3\mensaje.jpg

ARCHIVO 'mensaje.jpg' COPIADO CON EXITO 'C:\Users\carlo\OneDrive\Escritorio\SO\Proyecto 3\mensaje.jpg'.
```



Copiar un archivo de tu computadora hacia tu FiUnamFS

```
----MENÚ FIUNAMFS----
*** 1. LISTAR ARCHIVOS ***
*** 2. COPIAR ARCHIVO DE FiUnamFs AL SISTEMA ***
*** 3. COPIAR ARCHIVO DEL SISTEMA A FiUnamFS ***
*** 4. ELIMINAR ARCHIVO ***
*** 5. DESFRAGMENTAR ***
*** 6. MOSTRAR INFORMACION DEL SUPERBLOQUE ***
*** 7. SALIR ***
SELECCIONE UNA OPCION: 3



 INGRESE LA RUTA DEL ARCHIVO COMPLETA A COPIAR:
C:\Users\carlo\OneDrive\Escritorio\SO\Proyecto 3\UNAM.jpg
ARCHIVO COPIADO EXITOSAMENTE EN FIUNAMFS.
```

```
----MENÚ FIUNAMFS----
*** 1. LISTAR ARCHIVOS ***
*** 2. COPIAR ARCHIVO DE FiUnamFs AL SISTEMA ***
*** 3. COPIAR ARCHIVO DEL SISTEMA A FiUnamFS ***
*** 4. ELIMINAR ARCHIVO ***
*** 5. DESFRAGMENTAR ***
*** 6. MOSTRAR INFORMACION DEL SUPERBLOQUE ***
*** 7. SALIR ***
SELECCIONE UNA OPCION: 1


ARCHIVO: README.org, TAMAÑO: 31209, CLUSTER INICIAL: 6, FECHA CREACION: 20231116130339, FECHA MODIFICACION: 20231116130339

ARCHIVO: UNAM.jpg, TAMAÑO: 100664018, CLUSTER INICIAL: 838860800, FECHA CREACION: 02311242146182, FECHA MODIFICACION: 0231124214620

ARCHIVO: logo.png, TAMAÑO: 170192, CLUSTER INICIAL: 21, FECHA CREACION: 20231116130339, FECHA MODIFICACION: 20231116130339

ARCHIVO: mensaje.jpg, TAMAÑO: 102657, CLUSTER INICIAL: 154, FECHA CREACION: 20231116130339, FECHA MODIFICACION: 20231116130339
```

Eliminar un archivo del FiUnamFS

```
----MENÚ FIUNAMFS----
*** 1. LISTAR ARCHIVOS ***
*** 2. COPIAR ARCHIVO DE FiUnamFs AL SISTEMA ***
*** 3. COPIAR ARCHIVO DEL SISTEMA A FiUnamFS ***
*** 4. ELIMINAR ARCHIVO ***
*** 5. DESFRAGMENTAR ***
*** 6. MOSTRAR INFORMACION DEL SUPERBLOQUE ***
*** 7. SALIR ***
SELECCIONE UNA OPCION: 4



LOS ARCHIVOS EXISTENTES SON:

ARCHIVO: README.org, TAMAÑO: 31209, CLUSTER INICIAL: 6, FECHA CREACION: 20231116130339, FECHA MODIFICACION: 20231116130339

ARCHIVO: UNAM.jpg, TAMAÑO: 100664018, CLUSTER INICIAL: 838860800, FECHA CREACION: 02311242146182, FECHA MODIFICACION: 0231124214620

ARCHIVO: logo.png, TAMAÑO: 170192, CLUSTER INICIAL: 21, FECHA CREACION: 20231116130339, FECHA MODIFICACION: 20231116130339

ARCHIVO: mensaje.jpg, TAMAÑO: 102657, CLUSTER INICIAL: 154, FECHA CREACION: 20231116130339, FECHA MODIFICACION: 20231116130339


INGRESE EL NOMBRE DEL ARCHIVO A ELIMINAR:
UNAM.jpg
ARCHIVO ELIMINADO CON EXITO.
```

```
----MENÚ FIUNAMFS----
*** 1. LISTAR ARCHIVOS ***
*** 2. COPIAR ARCHIVO DE FiUnamFs AL SISTEMA ***
*** 3. COPIAR ARCHIVO DEL SISTEMA A FiUnamFS ***
*** 4. ELIMINAR ARCHIVO ***
*** 5. DESFRAGMENTAR ***
*** 6. MOSTRAR INFORMACION DEL SUPERBLOQUE ***
*** 7. SALIR ***
SELECCIONE UNA OPCION: 1


ARCHIVO: README.org, TAMAÑO: 31209, CLUSTER INICIAL: 6, FECHA CREACION: 20231116130339, FECHA MODIFICACION: 20231116130339

ARCHIVO: logo.png, TAMAÑO: 170192, CLUSTER INICIAL: 21, FECHA CREACION: 20231116130339, FECHA MODIFICACION: 20231116130339

ARCHIVO: mensaje.jpg, TAMAÑO: 102657, CLUSTER INICIAL: 154, FECHA CREACION: 20231116130339, FECHA MODIFICACION: 20231116130339
```

Desfragmentación de FiUnamFS

```
----MENÚ FIUNAMFS----
*** 1. LISTAR ARCHIVOS ***
*** 2. COPIAR ARCHIVO DE FiUnamFs AL SISTEMA ***
*** 3. COPIAR ARCHIVO DEL SISTEMA A FiUnamFS ***
*** 4. ELIMINAR ARCHIVO ***
*** 5. DESFRAGMENTAR ***
*** 6. MOSTRAR INFORMACION DEL SUPERBLOQUE ***
*** 7. SALIR ***
SELECCIONE UNA OPCION: 5
Iniciando desfragmentación...
Desfragmentación completada con éxito.
```

Información del superbloque

```
----MENÚ FIUNAMFS----
*** 1. LISTAR ARCHIVOS ***
*** 2. COPIAR ARCHIVO DE FiUnamFs AL SISTEMA ***
*** 3. COPIAR ARCHIVO DEL SISTEMA A FiUnamFS ***
*** 4. ELIMINAR ARCHIVO ***
*** 5. DESFRAGMENTAR ***
*** 6. MOSTRAR INFORMACION DEL SUPERBLOQUE ***
*** 7. SALIR ***
SELECCIONE UNA OPCION: 6

Información del Superbloque:
Versión del sistema de archivos: 24.1
Tamaño del Cluster: 2048 bytes
Inicio del Directorio: 2048 bytes
Tamaño del Directorio: 8192 bytes
Tamaño de Entrada del Directorio: 64 bytes
Número Máximo de Clusters: 720
```