

FI UNAM

PROYECTO 2 - (MICRO) SISTEMAS DE ARCHIVOS

Nombre: De La Cruz López Carlos Odette

Grupo: 06

Profesor: Ing. Gunnar Eyal Wolf Iszaevich

Fecha: 05/Enero/2023

SISTEMAS OPERATIVOS

Proyecto 2: (Micro) sistema de archivos

Descripción del proyecto

Para la unidad de sistemas de archivos, creo que resulta natural que el proyecto sea implementar un sistema de archivos 😊 Para esto, lo harán trabajando sobre una *especificación* y sobre un *caso de referencia*.

¿Qué tengo que hacer?

Lo que ustedes deben desarrollar es un programa que pueda obtener, crear y modificar información en el micro-sistema-de-archivos de la Facultad de Ingeniería, **FiUnamFS**.

Siguiendo la especificación que aparece en la siguiente sección, tienen que desarrollar un programa que pueda:

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. Desafortunadamente, este sistema de archivos *simplote* es muy dado a la fragmentación externa. Generen también un programa que *desfragmente* al **FiUnamFS**. Ojo, la desfragmentación debe hacerse *dentro del sistema de archivos* (no creando un sistema de archivos nuevo y copiando hacia éste).

Sistema muestra

Para verificar su implementación, pueden [descargar un sistema de archivos ejemplo](#) que cumple con el planteamiento. Verifiquen que pueden realizar todas las tareas que les solicité con éste. Indíqueme — ¿Cuáles son los contenidos del «disco»?

Especificación de **FiUnamFS**

- El sistema de archivos cabe en un *diskette* tradicional. Claro, no espero que tengan acceso al hardware, por lo que lo simularemos representándolo en un archivo de longitud fija, de 1440 Kilobytes
- Por simplicidad, en todas las estructuras de **FIUnamFS**, las cadenas de texto deben ser ASCII 8-bit (*no* Unicode UTF-8).
- En las estructuras del disco, todos los enteros serán representados como valores de 32 bits, en formato *little endian*.
 - Esto es, por ejemplo: El valor 1354 se representará como los bytes (74, 5, 0, 0) (porque $((0 * 256 + 0) * 256 + 5) * 256 + 74 = 1354$). El valor 1234567890 se representará como los bytes 210, 2, 150, 7 (porque $((7 * 256 + 150) * 256 + 2) * 256 + 210 = 1234567890$).
 - Para hacer las conversiones desde o hacia este formato, les sugiero usar el formato <l con las funciones `pack()` y `unpack()`, disponibles en distintos lenguajes ([Python: struct.pack\(\)](#), [Ruby: Array#pack\(\)](#), [Perl: pack\(\)](#), [PHP: pack\(\)](#)), hay varias bibliotecas que pueden usar para Javascript, como [node-jspack](#) o [bufferpack](#)). Si van a usar C o C++, les sugiero revisar las

funciones [htons\(\)](#) y [ntohs\(\)](#). ¿Java? Si bien no forman parte del lenguaje, pueden encontrar [varias implementaciones](#) de `htons()` y `ntohs()` en la red.

- La superficie del disco se divide en sectores de 256 bytes. Cada clúster mide cuatro sectores.
- El pseudodispositivo no maneja *tabla de particiones*, hospeda directamente un *volumen* en la totalidad de su espacio.
- **FiUnamFS** maneja únicamente un *directorio plano*, no se consideran subdirectorios.
- El primer clúster (#0) del pseudodispositivo es el *superbloque*. Este contiene información en los siguientes bytes:
 - **0–8**
 - Para identificación, el nombre del sistema de archivos. ¡Debes validar nunca modificar un sistema de archivos que no sea el correcto! Debe ser la cadena `FiUnamFS`.
 - **10–14**
 - Versión de la implementación. Estamos implementando la versión 23.1 (como cadena, no como número). Deben validar que el sistema de archivos a utilizar sea de esta versión, para evitar la corrupción de datos.
 - **20–35**
 - Etiqueta del volumen
 - **40–44**
 - Tamaño del clúster en bytes
 - **45–49**
 - Número de clusters que mide el directorio
 - **50–54**
 - Número de clusters que mide la unidad completa
- El resto del superbloque puede quedar vacío (o puedes *sobrecargarlo* con otros datos — No es importante para la implementación.
- **Ojo:** Por confusión mía, al plantear el proyecto el último valor (número de clusters en la unidad) aparecía sobreponiéndose al anterior (número de clusters en el directorio), del carácter 48 al 51. El valor correcto es 50–54. ¡Agradezco a la compañera que me hizo ver el error, con excelente información para sustentarlo!
- El sistema de archivos es de *asignación contigua*. Toda la información de los archivos está en el directorio.

- El directorio está ubicado en los clusters 1 a 4. Cada entrada del directorio mide 64 bytes, consistentes en:
 - 0
 - Tipo de archivo. Dado que el sistema de archivos actual no tiene soporte para directorios, dispositivos, pipes, ni otros archivos especiales, siempre será el carácter - (0x2d, 45)
 - 1–15
 - Nombre del archivo
 - 16–20
 - Tamaño del archivo, en bytes
 - 20–23
 - Clúster inicial
 - 24–37
 - Hora y fecha de creación del archivo, especificando AAAAMMDDHHMMSS (p.ej. '20221108182600' para 2022-11-08 18:26:00)
 - 38-51
 - Hora y fecha de última modificación del archivo, especificando AAAAMMDDHHMMSS (p.ej. '20221109182600')
 - 52–64
 - Espacio no utilizado (¿reservado para expansión futura?)
- Las entradas no utilizadas del directorio se identifican porque en el campo de nombre llevan la cadena -----.
 - Los nombres de archivos pueden componerse de cualquier carácter dentro del subconjunto ASCII de 7 bits (no acentuados, no Unicode, sólo el *viejo y aburrido* US-ASCII)
- Es un sistema de archivos plano — No maneja subdirectorios.
- Después del directorio, todo el espacio restante es espacio de datos.

Desarrollo del proyecto

En este punto del desarrollo, solo se logró, listar los archivos, copiar un archivo del “diskette” a nuestra computadora, eliminar un archivo y finalizar el programa, la función dedicada a la desfragmentación no fue posible

```
#####
#####  Menú principal  #####
#####
#
#      ls          -> Listar contenido del directorio      #
#      cp          -> Copiar archivo                      #
#      rm          -> Eliminar archivo de FiUnamFS         #
#      defrag      -> Desfragmentar FiUnamFS               #
#      exit        -> Salir de FiUnamFS                   #
#
#####
Opcion:
```

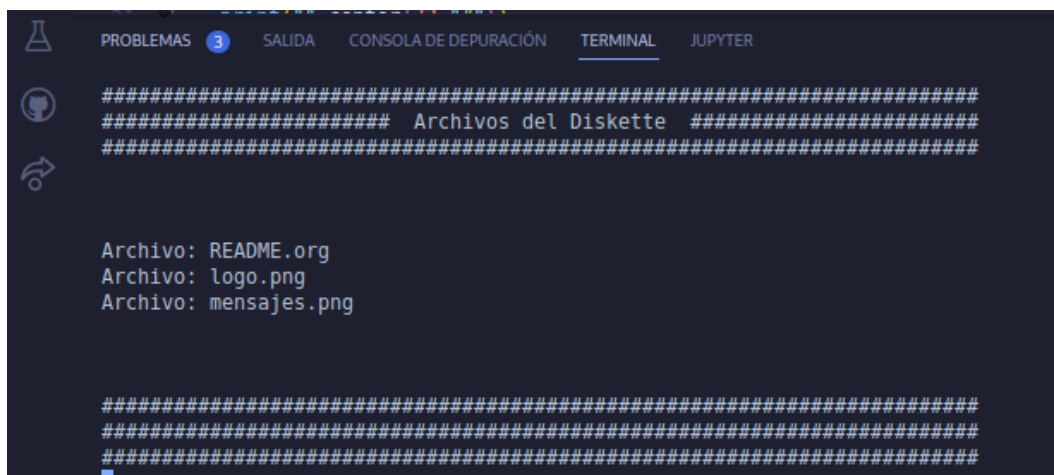
¿Como usar el programa?

Se escribe, que función se quiere realizar, se intentó reutilizar como lo sería en terminal de Linux,

Casos con nuestras instrucciones implementadas

Ejecutar instrucción “ls”

Al escribir ls y presionar enter, el programa procede a listar archivos, por el momento, se logró únicamente visualizar el nombre de los archivos almacenados actualmente en nuestro “diskette”



```
#####
#####  Archivos del Diskette  #####
#####

Archivo: README.org
Archivo: logo.png
Archivo: mensajes.png

#####
#####
#####
```

Ejecutar instrucción cp

Al escribir en nuestra ejecución la instrucción cp

```
#####
#####  Menú principal  #####
#####
#
#      ls              -> Listar contenido del directorio      #
#      cp              -> Copiar archivo                        #
#      rm              -> Eliminar archivo de FiUnamFS          #
#      defrag          -> Desfragmentar FiUnamFS                #
#      exit            -> Salir de FiUnamFS                    #
#
#####
Opcion: cp
```

Procede a desplegar una pantalla para ingresar que archivo se copiara de nuestro diskette a nuestra computadora, en nuestro ejemplo se intentara copiar en una carpeta creada para esta demostración el cual tiene por dirección y contenido el siguiente:

```
(carlosodette0803@ Carlos0detteDLCL) - [~/Descargas/ejemploParaElProyecto]
• $ pwd
/home/carlosodette0803/Descargas/ejemploParaElProyecto

• $ dir

• $ ls

○ $
```

Procedemos ejecutar la instrucción cp, en nuestro programa e ingresaremos los datos para que se almacene ahí, en este caso obtendremos la copia del archivo logo.png

```
PROBLEMAS 1 SALIDA CONSOLA DE DEPURACIÓN TERMINAL JUPYTER

Ingresa el nombre del archivo ubicado en el Diskette que deseas copiar
Copiar: logo.png
Ingresa la direccion donde se almacenará la copia
Direccion: /home/carlosodette0803/Descargas/ejemploParaElProyecto
```

En este caso se ejecutó correctamente

```
PROBLEMAS 1 SALIDA CONSOLA DE DEPURACIÓN TERMINAL JUPYTER

Ingresa el nombre del archivo ubicado en el Diskette que deseas copiar
Copiar: logo.png
Ingresa la direccion donde se almacenará la copia
Direccion: /home/carlosodette0803/Descargas/ejemploParaElProyecto
Copiado exitosamente al directorio /home/carlosodette0803/Descargas/ejemploParaElProyecto
```

Procedemos a ver que archivo existe en el directorio escogido para esta representación

```
PROBLEMAS 1 SALIDA CONSOLA DE DEPURACIÓN TERMINAL JUPYTER

(carlosodette0803@Carlos0detteDLCL) - [~]
$ cd Descargas/ejemploParaElProyecto

(carlosodette0803@Carlos0detteDLCL) - [~/Descargas/ejemploParaElProyecto]
$ ls
Logo.png
```



Ejecutar instrucción rm

Esta instrucción nos servirá para eliminar un archivo, en nuestro “diskette” al ser un almacenamiento sin carpetas, es posible solo eliminar colocando el nombre, siempre y cuando este exista, en este caso intentaremos eliminar únicamente a logo.png

```
PROBLEMAS 1 SALIDA CONSOLA DE DEPURACIÓN TERMINAL JUPYTER

#####
#####  Menú principal  #####
#####
#
#   ls           -> Listar contenido del directorio  #
#   cp           -> Copiar archivo                  #
#   rm           -> Eliminar archivo de FiUnamFS     #
#   defrag       -> Desfragmentar FiUnamFS           #
#   exit        -> Salir de FiUnamFS                 #
#
#####
Opcion: rm
```

```
PROBLEMAS 1 SALIDA CONSOLA DE DEPURACIÓN TERMINAL JUPYTER

Ingresa el archivo que deseas eliminar del diskette:  logo.png
El archivo: logo.png, fue eliminado exitosamente
```

Para verificar que se haya eliminado correctamente, se verá con ls los archivos después de esta instrucción

```
##### Archivos del Diskette #####
#####

Archivo: README.org
Archivo: mensajes.png

#####
#####
```

Instrucción exit

Y solo basta con colocar exit para que se cierre el programa

Recomendaciones (necesarias) para la ejecución del programa:

Debe colocar la dirección de la carpeta en la que se encuentra el programa, solo para evitar errores

En mi caso, se usó el siguiente directorio:

```
4 from struct import unpack, pack
5 #Este comando es mas para mi, ya que era "tedioso" escribir la carpeta a la cual se almacena los archivos cada vez que arrancaba el programa
6 directorio = '/home/carlosodette0803/sistop-2023-1/proyectos/2/DeLaCruzLopezCarlos0dette'
7 chdir(directorio)
8
```