

SISTEMAS OPERATIVOS Y REDES II

Primer Semestre 2025

Trabajo Practico 1

Sistema de Archivos FAT

Objetivo

El objetivo de este trabajo práctico es explorar en profundidad el sistema de archivos FAT12, utilizando una imagen de disco (*test.img*). Los alumnos aprenderán a:

1. Leer y analizar el MBR (Master Boot Record).
2. Interpretar la estructura del Boot Sector de FAT12.
3. Leer y manipular el directorio raíz de FAT12.
4. Recuperar archivos borrados y entender cómo funciona la gestión de archivos en FAT12.

Instrucciones Generales

1. Formato de entrega:

- El informe debe estar en formato PDF.
- El código debe estar correctamente documentado.
- Fecha de entrega: miércoles 18 de septiembre de 2024, hasta las 23:59hs
- Cada integrante del grupo deberá subir al campus Moodle el informe y los archivos del TP. Es importante que todos suban el mismo trabajo para que quede registrado el aporte de cada uno.
- Todos los archivos (código, informe y test.img) deben estar comprimidos en un archivo con el nombre:
TP[Numero de Tp]_[Materia]_[año y semestre]_[Apellido y Nombre del que subió el archivo].tar.gz. Ejemplo: TP1_SOR2_2024_2s_PérezLucas.tar.gz.

2. Herramientas:

Editor hexadecimal: **Ghex** instalado en Ubuntu

```
sudo apt-get update  
sudo apt-get install ghex
```

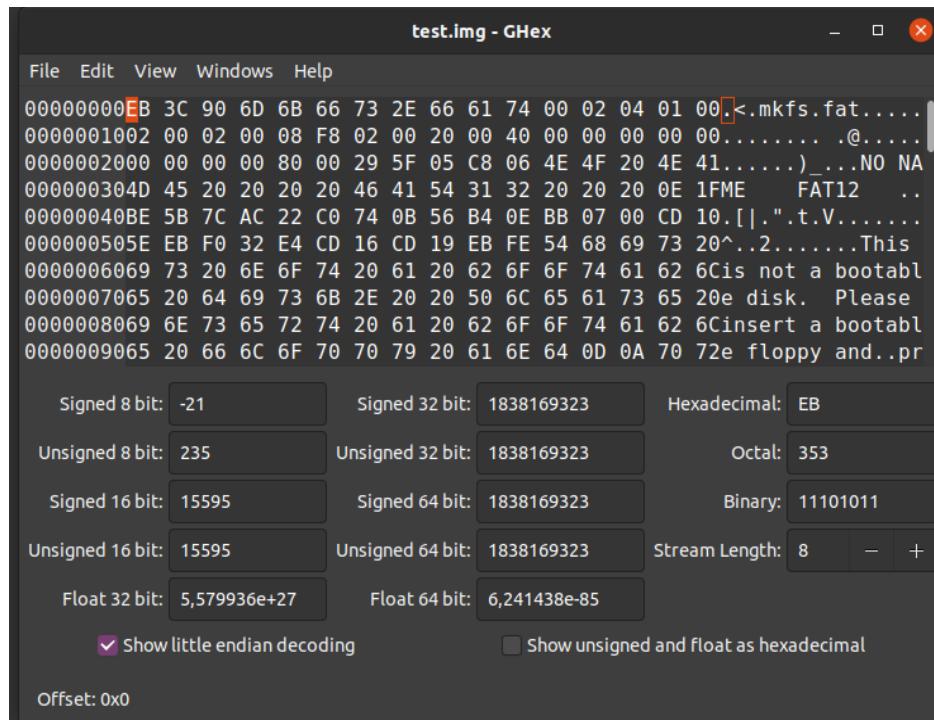
Montaje de la imagen:

```
sudo mount test.img /mnt -o loop,umask=000  
sudo umount /mnt
```

3. Código:

El código debe estar dividido en archivos separados:

- read_mbr.c: Para leer el MBR.
- read_boot.c: Para leer el Boot Sector.
- read_root.c: Para leer el directorio raíz.



Consignas:

1. Montaje del Sistema de Archivos

- ¿Para qué se ha puesto `umask=000` al montar el sistema de archivos? Explicar el propósito de este parámetro.

2. Cargando el MBR

a. Mostrando el MBR con el Hex Editor

- Abrir *test.img* con *Ghex*.
- Mostrar los primeros bytes y la tabla de particiones.
- Indicar cuántas particiones hay y en qué lugar se observa esta información.

b. Leer los datos del MBR utilizando código C

- Escribir un programa en C (*read_mbr.c*) que lea y muestre la tabla de particiones.
- Mostrar por pantalla:
 - El flag de bootable.
 - La dirección CHS (Cylinder-Head-Sector) de inicio y fin.
 - El tipo de partición.
 - El tamaño en sectores.

c. Verificar si la primera partición es bootable

- Usar *Ghex* para verificar si la primera partición es bootable.
- Explicar cómo se determina esto.

d. Mostrar información de la primera partición

Modificar el programa *read_mbr.c* para mostrar:

- El flag de booteable.
- La dirección CHS.
- El tipo de partición.
- El tamaño en sectores.

3. Cargando la Tabla de Archivos

a. Mostrar archivos en el filesystem

- Usar *Ghex* para mostrar cuántos y cuáles archivos tiene el filesystem.
- Escribir un programa en C (*read_root.c*) que lea y muestre los archivos en el directorio raíz.

b. Crear y borrar un archivo

- Montar el filesystem y crear un archivo en la carpeta raíz (/).
- Borrar el archivo y buscarlo en *Ghex*.
- Mostrar el archivo borrado usando el código en *read_root.c*.

c. Mostrar archivos borrados

- Explicar cómo se identifican los archivos borrados en *Ghex*.
- Modificar *read_root.c* para mostrar archivos borrados.

d. Recupero de archivos

- Explicar cómo funciona el recupero de archivos en FAT12.
- Escribir un programa en C (*read_root.c*) para recuperar un archivo borrado.

4. Leyendo Archivos

a. Crear un archivo y mostrarlo

- Montar el filesystem y crear un archivo llamado *lapapa.txt* en la carpeta raíz (/).
- Buscar el archivo en *Ghex* y mostrarlo usando *read_root.c*.

b. Mostrar el contenido de un archivo no borrado

- Usar *Ghex* para mostrar el contenido de un archivo no borrado.
- Modificar *read_root.c* para mostrar el contenido del archivo.

c. Recuperar un archivo borrado

- Escribir un programa en C (*read_root.c*) que busque un archivo borrado y lo recupere.

Entrega

1. Informe:

Explicar cómo se resolvió cada consigna.

Incluir capturas de pantalla de *Ghex* y salidas de los programas.

2. Código:

Entregar los archivos *read_mbr.c*, *read_boot.c*, *read_root.c*, y *test.img*.

3. Compresión:

Comprimir todos los archivos en un *archivo .tar.gz* con el formato especificado

Referencias:

- [Diseño del Sistema de Archivos FAT](#)
- [Estructuras de Datos de Sistemas de Archivos](#)