

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERU
FACULTAD DE CIENCIAS E INGENIERÍA

SISTEMA DE ARCHIVO
exFAT

En el directorio Descargas se encuentra el archivo: *usb20242.img.zip* desempaquetelo.

El siguiente laboratorio será experimental.

En el material para el laboratorio que se le ha proporcionado se encuentra el siguiente fragmento de texto:

Theoretically, this change has the potential of speeding up I/O operations. When writing on a FAT32 file system the FAT must be accessed when each cluster is allocated or read. In exFAT this operation is flipping a bit in the Allocation Bitmap. As long as the file remains not fragmented, the FAT does not need to be updated. Even if there is data already in the FAT, those corresponding cells don't even need to be zeroed because there is a flag in the Stream Extension Directory Entry (Section 6.9) that indicates that the FAT is invalid. Read operations of a non-fragmented file stored in exFAT would not require access to the FAT or the Allocation Bitmap and reduces overall file system I/O overhead.

Para ello seguirá los siguientes pasos:

- 1) **(5 puntos)** Elabore un programa en C, con nombre *entryFat.c*, que muestre todas las entradas válidas de la FAT. La salida debe mostrarse como una secuencia de enteros (para final de archivo emplee -1).
- 2) **(5 puntos)** Elabore un programa en C, con nombre *tableBitMap.c*, que muestre la tabla de BitMap. Imprima su contenido como caracteres en hexadecimal. Emplee la secuencia de salida “%x”.
- 3) Redirija la salida de su programa *entryFat* hacia un archivo con nombre *salida1*
- 4) Redirija la salida de su programa *tableBitMap* hacia un archivo con nombre *salida2*
- 5) **(5 puntos)** Modifique el sistema de archivos añadiendo archivos (vea el procedimiento al final). Añada un archivo sin fragmentar y otro fragmentado. Para obtener un archivo fragmentado debe tener en cuenta que el S.O. tratará de asignarle *clusters* libres consecutivamente mientras se pueda. Es de suponerse que si hay otro archivo a continuación y el primero quiere crecer, deberá asignarle otros *clusters* no consecutivos, con lo cual el archivo quedará fragmentado.
- 6) Una vez terminado el paso 5 vuelva a obtener las salidas de los pasos 3 y 4, pero con nombres *salida3* y *salida 4*.
- 7) **(5 puntos)** En un archivo de texto con nombre *conclusiones.txt* escriba las conclusiones a las que llega basándose en las salidas obtenidas anteriormente.

Procedimiento para modificar el sistema de archivos.

En primer lugar procederemos a montarlo, debe conocer la ruta absoluta del archivo imagen. Asumamos que se encuentra en `/home/alulab/Documentos`, entonces

```
udisksctl loop-setup -f /home/alulab/Documentos/image.lab.usb
```

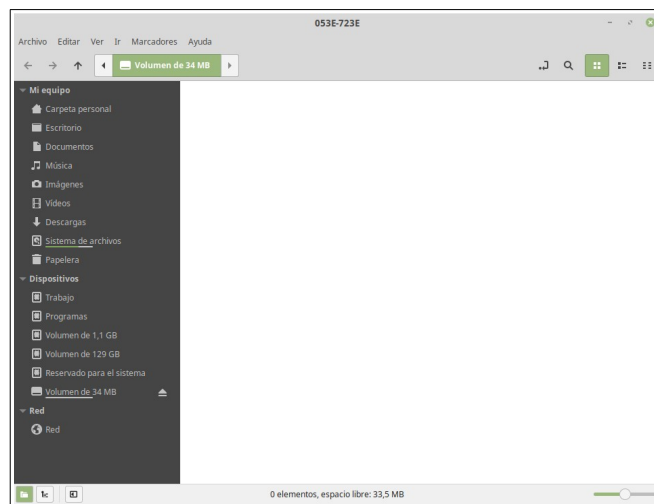
Como respuesta deberá aparecer en la terminal los siguiente:

```
Mapped file /home/alulab/Documentos/FAT16D1.img as /dev/loop0.
```

El sistema crea un directorio con nombre igual a algo como 053E-723E (el nombre puede variar) en `/media/alulab/` (asumiendo que el usuario es `alulab`) y monta sobre este directorio la imagen del disco. El directorio creado tiene como propietario al que ejecutó el comando (`alulab` en este caso), de forma que puede copiar y borrar archivos en esta imagen. Puede ver su contenido de forma acostumbrada:

```
ls -l /media/alulab/053E-723E
```

También se abrirá (en Linux Mint 20) una ventana del navegador de archivos (Nemo) mostrando el directorio donde se ha montado la imagen.



En adelante cuando ingrese a este directorio, estará accediendo a este disco virtual. Cuando ya no se desee acceder más a esta imagen se debe desmontarlo para que los cambios tomen efecto.

```
udisksctl unmount -b /dev/loop0
```

La idea de enseñarle este procedimiento, es que usted pueda crear y modificar su propia imagen, muy aparte de las imágenes que se le proporciona. De esta forma podrá experimentar con el sistema de archivos exFAT proporcionado.

Pando, 21 de noviembre de 2024

Prof: Alejandro T. Bello Ruiz