

# Clase de Filesystem + Taller

Sergio Romano

5 de octubre de 2014



Ext2

# Presentación Taller

- ▶ En el taller que nos trajo a todos aca, tienen que trabajar con el sistema de archivos *Ext2*
  - ▶ Popular, muy documentado, mejorado en *Ext3* y *Ext4*.
  - ▶ Empezó como el *filesystem* de Linux pero implementaciones existen para BSD, Windows, etc.
- ▶ Les damos una implementación parcial de un *driver* para *Ext2* y tienen que implementar unas funciones.
- ▶ Luego tienen que usar esas funciones para leer una imagen de disco y obtener unos datos.

# Un poco de historia

- ▶ Presentado en 1993 como un reemplazo de *ext* (*timestamps*, fragmentación, archivos de hasta 2TB).
- ▶ Basado en *ext* y en *ufs* (Unix File System o Berkeley Fast File System)

# Estructura

- ▶ El disco empieza con espacio para el *bootblock* de la partición.
- ▶ Después sigue un conjunto de *Block Groups*.
  - ▶ División lógica para reducir fragmentación y favorecer localidad
  - ▶ Empiezan con el *super block*, que tiene todos los datos de la configuración del *filesystem* (número de inodos y bloques totales y libres, cuando fue montado por última vez, etc.)
    - ▶ Replicado en cada grupo.
  - ▶ Sigue el *block group descriptors*, que tiene los datos de cada grupo.
    - ▶ También replicado como el *super block*.
- ▶ Luego sigue la información específica a ese grupo de bloques.
  - ▶ *Data block bitmap*
  - ▶ *Inode bitmap*
  - ▶ *Inode table*
  - ▶ *Data blocks*: Donde guardamos los datos efectivamente.



# Inodo de Ext2 en detalle.

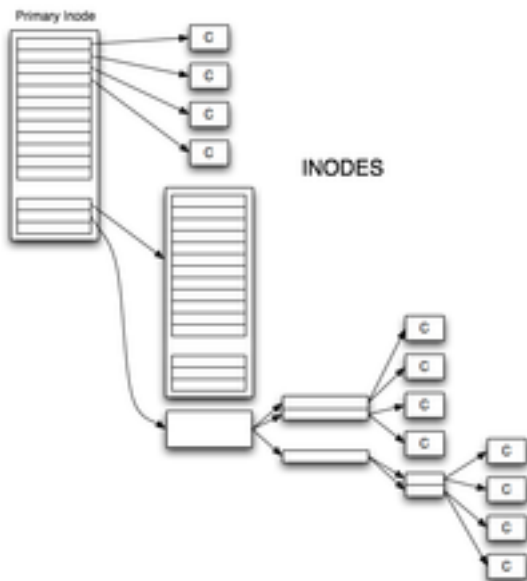
- ▶ Tiene la información del archivo: ubicación, tamaño, tipo, permisos y punteros a los bloques datos.
- ▶ Los inodos pueden ser desde directorios hasta pipes, sockets, devices, archivos regulares, etc.
- ▶ Punteros a datos
  - ▶ Cada inodo tiene 12 punteros a bloques con datos.
  - ▶ Cada inodo tiene un puntero indirecto.
    - ▶ Puntero a un bloque con punteros a bloques
  - ▶ Cada inodo tiene un puntero doble indirecto.
    - ▶ Puntero a un bloque con punteros a bloques con punteros.
- ▶ ¿Y el nombre de archivo?



# Inodo directorio

- ▶ En vez de datos crudos, tienen estructuras de directorio.
- ▶ Cada estructura guarda la longitud de la entrada, longitud del nombre, nombre (no necesariamente null-terminated) e inodo.
- ▶ Los pedazos se juntan con / en UNIX.
- ▶ Un inodo fijo (el 2) es el *root node* (/) del filesystem.

# Inodo esquemático



# Estructuras en el código

- ▶ Veamos como es cada estructura en el código.
- ▶ Las mismas son copia exacta de como se ubican en disco (asumiendo Little-Endian), ya que los `structs` y `clases` de C++ tienen contenido consecutivo en memoria.
- ▶ Miremos entonces:
  - ▶ MBR
  - ▶ Super bloque
  - ▶ Descriptor de grupo de bloques
  - ▶ Inodo
  - ▶ Entrada de directorio.

# Enunciado

# Enunciado Taller

1. Completar la implementación de los siguientes métodos:
  - 1.1 `get_block_address(inode,block_number)`
  - 1.2 `load_inode(inode_number)`
  - 1.3 `get_file_inode_from_dir_inode(from,filename)`
2. Hacer un programa que, utilizando el FS programado en el punto anterior, imprima los 17 caracteres que se encuentran guardados en el archivo `/grupos/gNUMERO/nota.txt` (de la imagen de disco `hdd.raw` provista) a partir de la posición 14000 inclusive.

# Ayudas para el taller

- ▶ Haganlos en el orden dado.
- ▶ Funciones utiles:
  - ▶ `read_block`: Lee un bloque de disco a un buffer.
  - ▶ `strcmp`: Devuelve 0 si dos strings de C son iguales.
  - ▶ `block_group`: Puntero a descriptor del blockgroup.
  - ▶ `blockgroup_for_inode`: Número de blockgroup del inodo.
  - ▶ `blockgroup_inode_index`: Offset dentro de la tabla de inodos para el inodo.
- ▶ Datos sobre el *filesystem* en general: En el superbloque.
- ▶ Datos sobre un *block group*: En su descriptor de blockgroup.
- ▶ Directorio: Un archivo común, CONTENIDO son entradas de directorio.
- ▶ Descompriman la imagen *hdd.raw.gz* en */tmp* para usarla.
- ▶ ¡Tienen estructuras para cada tipo necesario!
- ▶ ¡Hay funciones de filesystem para el punto 2!

# Documentación Taller

- ▶ <http://www.nongnu.org/ext2-doc/ext2.html>
- ▶ <http://e2fsprogs.sourceforge.net/ext2intro.html>
- ▶ <http://wiki.osdev.org/Ext2>
- ▶ <http://oreilly.com/catalog/linuxkernel2/chapter/ch17.pdf>