PHẦN MỀM HỆ THỐNG
HỆ THỐNG TỆP ẢO

Nguyễn Hữu Đức

Viện Công nghệ thông tin và Truyền thông
Trường Đại học Bách khoa Hà Nội

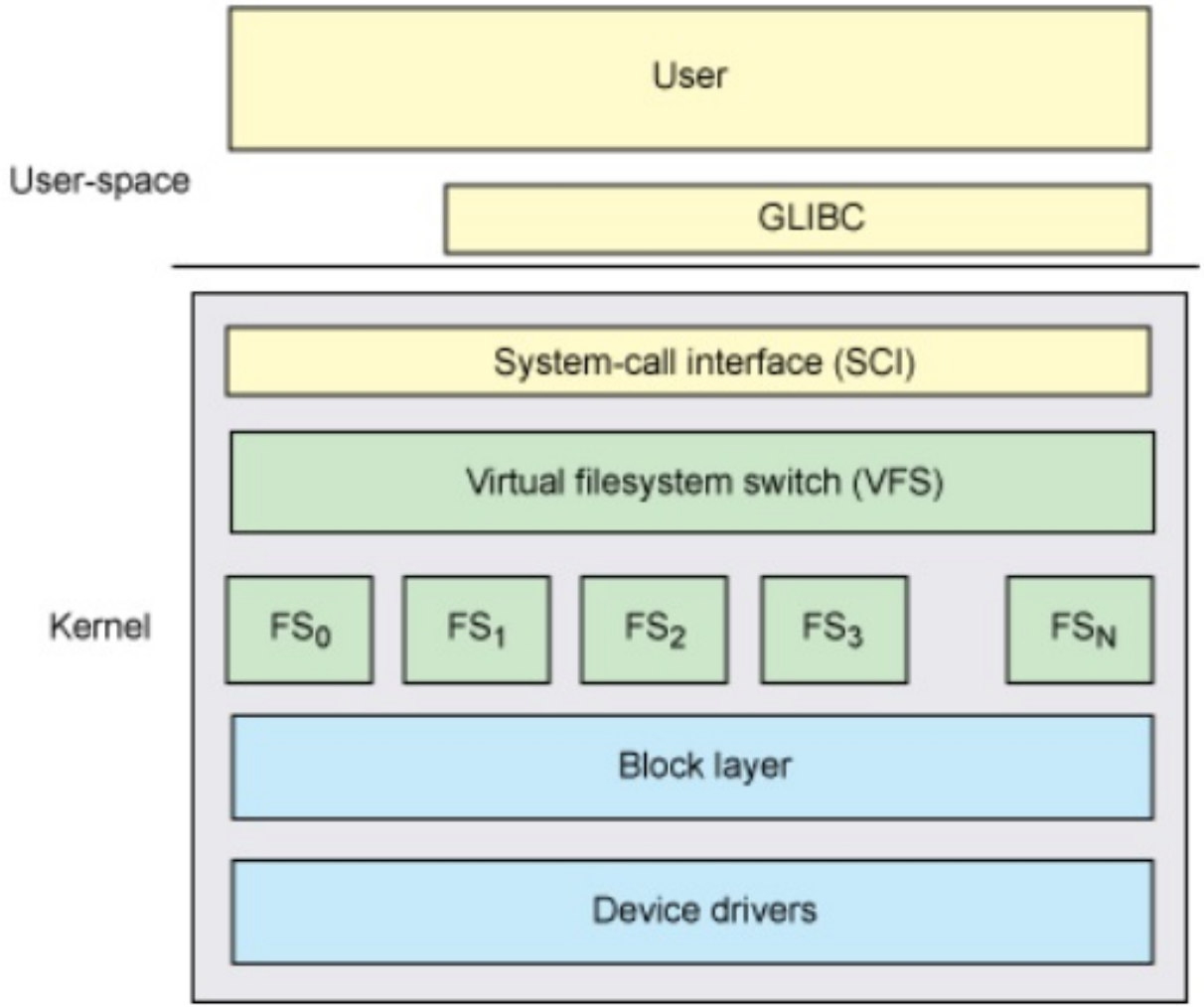
Nội dung

- Tổng quan về hệ thống tệp ảo
- Các thành phần chính của một hệ thống tệp
- Một số hệ thống tệp cơ bản

Tổng quan về hệ thống tệp ảo - VFS

- Mục đích: cho phép hệ điều hành đồng thời sử dụng nhiều hệ thống tệp khác nhau.
  - Các hệ thống tệp lưu trữ dạng đĩa: ext2, ntfs, fat, isofs, ...
  - Các hệ thống tệp mạng: nfs, sambafs, coda, ...
  - Các hệ thống tệp đặc biệt: procfs, ramdisk, ...
- Ý tưởng:
  - Sử dụng mô hình thống nhất cho các hệ thống tệp. Mỗi hệ thống tệp sẽ cung cấp cài đặt cụ thể cho từng thành phần của mô hình thống nhất này.
  - Cung cấp một phương pháp truy nhập thống nhất. Ứng dụng có thể truy nhập vào bất cứ hệ thống tệp nào thông qua các giao diện lời gọi hệ thống, sau đó một cơ cấu VFS switch sẽ chuyển hướng các lời gọi hệ thống đến cài đặt cụ thể của từng hệ thống tệp.
  - Cung cấp các cơ chế quản lý quyền truy nhập và tối ưu hóa truy nhập (vd caching) thống nhất.
  - Trên linux, các hệ thống tệp có thể được nạp một cách linh hoạt như các LKM

Truy nhập vào hệ thống tệp



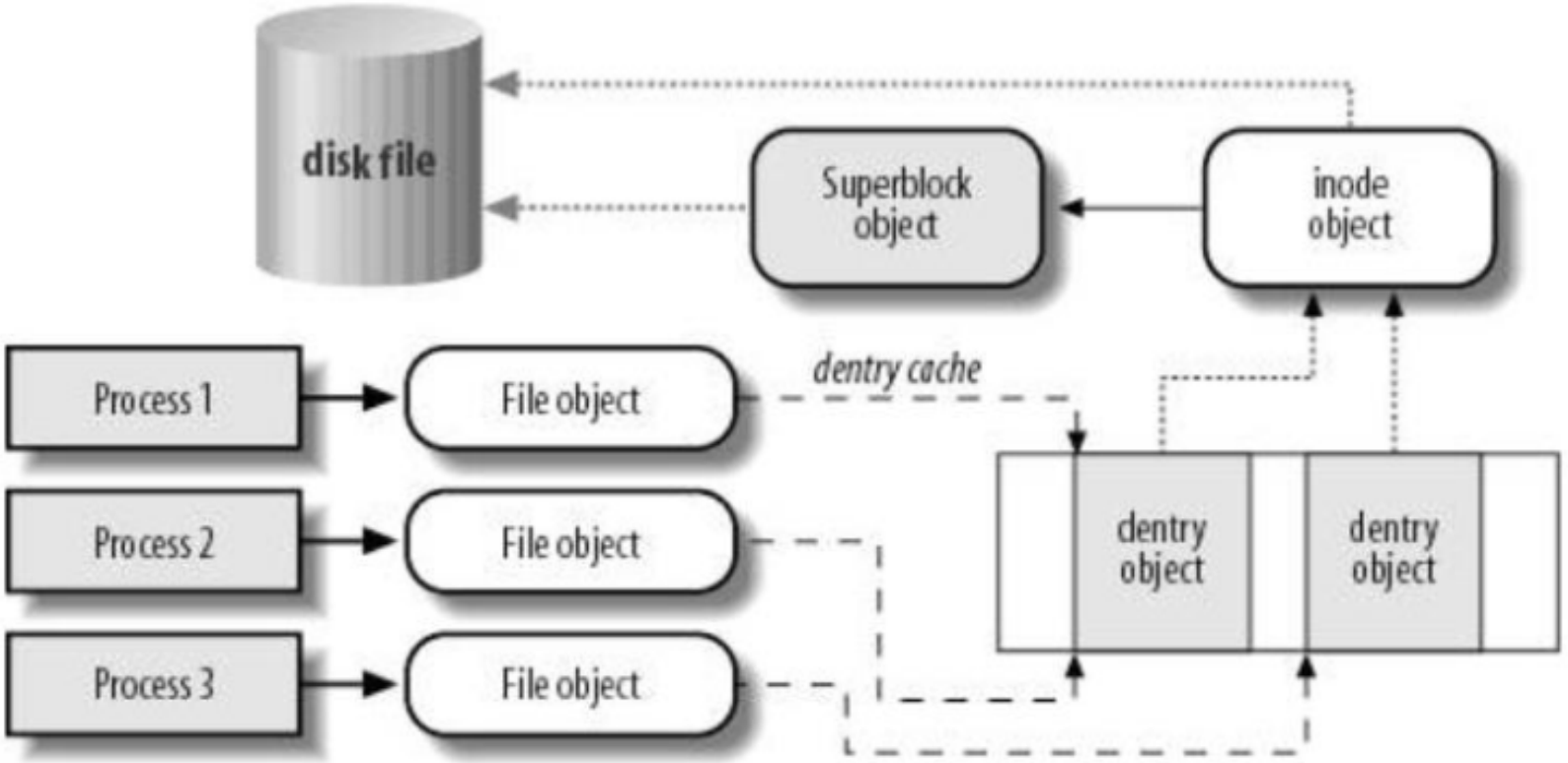
Tổng quan về hệ thống tệp ảo

- Nhiệm vụ của VFS
  - Chuyển hướng lời gọi hệ thống
  - Lưu trữ metadata của hệ thống tệp trong bộ nhớ, phối hợp với page cache
  - Quản lý truy cập thống nhất
  - Cung cấp một số cài đặt chuẩn như path lookup, quản lý file handle,...
- Nhiệm vụ của FS
  - Cài đặt các đối tượng/phương thức trừu tượng của hệ thống tệp
  - Chuyển đổi đối tượng chuẩn sang thiết bị lưu trữ, sử dụng giao diện của các block device.
  - Đọc/ghi page file

Các thành phần chính của một hệ thống tệp

- superblock: thông tin chung về hệ thống tệp
- inode: siêu dữ liệu về một tệp (vd. vị trí của dữ liệu tệp)
- dentry: ánh xạ tên tệp và inode tương ứng
- file: thông tin về một tệp đang được mở (dentry và con trỏ tệp)

Mối quan hệ giữa các thành phần của FS

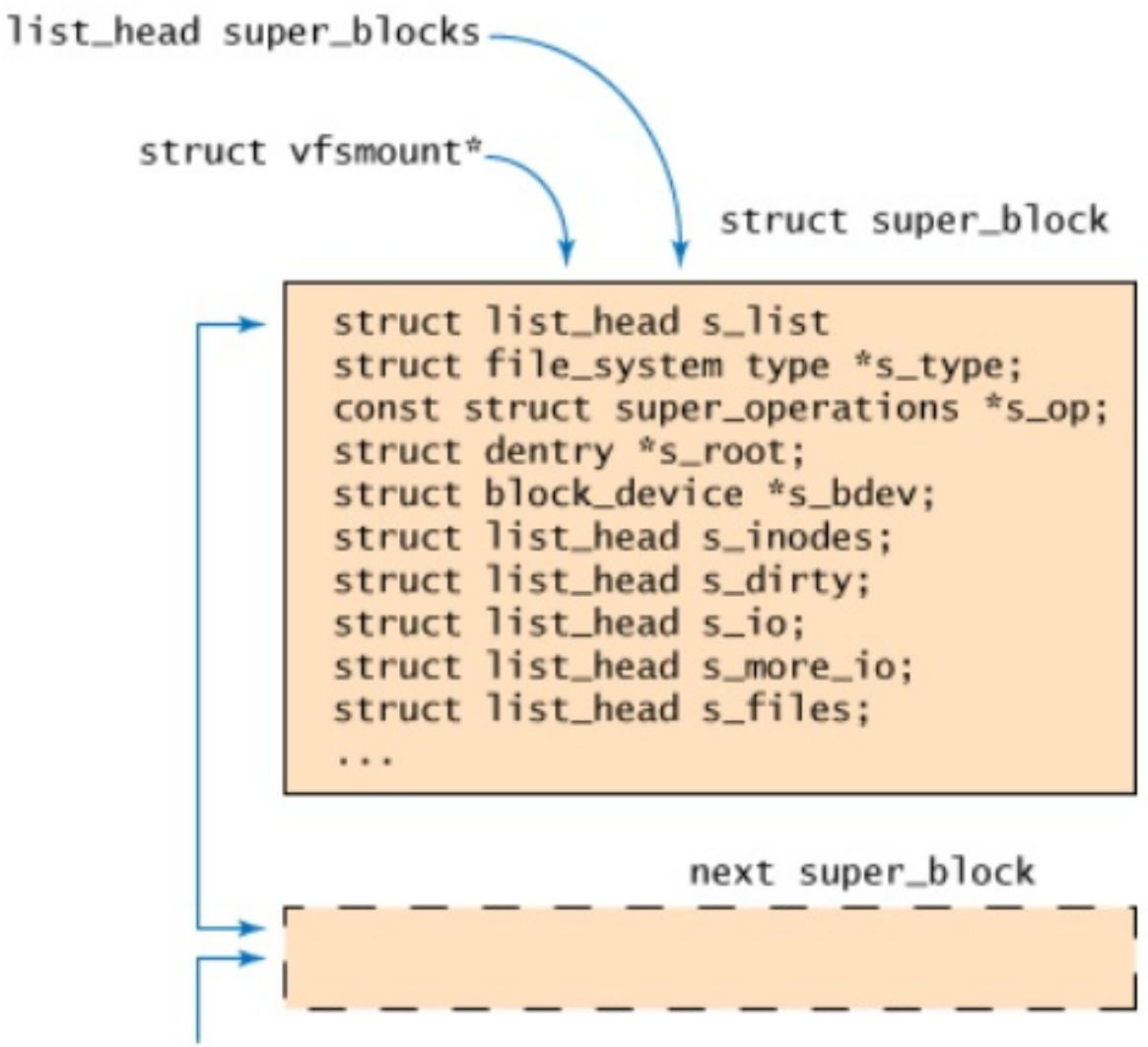


From *Understanding Linux kernel, 3rd Ed*

Superblock

- thông tin chung về hệ thống tệp
- các hệ thống tệp cụ thể cài đặt nhiều phương thức trừu tượng như tạo/hủy inode
- quản lý cờ đồng bộ với thiết bị lưu trữ
- kernel quản lý một danh sách hệ thống tệp được sử dụng

Superblock


list_head super_blocks
struct vfsmount*
struct super_block
struct list_head s_list
struct file_system type *s_type;
const struct super_operations *s_op;
struct dentry *s_root;
struct block_device *s_bdev;
struct list_head s_inodes;
struct list_head s_dirty;
struct list_head s_io;
struct list_head s_more_io;
struct list_head s_files;
...
next super_block

inode

- index node - virtual node: chứa siêu dữ liệu về tệp
  - thuộc tính tệp: quyền truy nhập, kích thước, thời gian thay đổi,...
  - nội dung tệp: vị tri lưu trữ tệp trên bộ nhớ và trên đĩa (thường là vị trí của các block dữ liệu)
  - cờ trạng thái (vd. dirty inode, dirty data, ...)
- Một inode giúp xác định duy nhất một tệp dữ liệu (không phụ thuộc vào tên tệp)
- Thông thường lưu trữ thông tin số lượng tham chiếu (vd. khi có một ánh xạ từ tên tệp tới inode).
  - Nếu tham chiếu $= 0$ thì inode sẽ bị xóa (unlink)
  - Hard link: tệp mới tham chiếu đến cùng inode
- TRICK: tự động dọn dẹp file tạm khi chương trình bị lỗi
  - create (1 link)
  - open (1link, 1 ref)
  - unlink

inode

struct inode

```
struct list_head i_dentry;
struct timespec i_atime;
struct timespec i_mtime;
struct timespec i_ctime;
gid_t i_gid;
uid_t i_uid;
loff_t i_size;
const struct file_operations *i_fop;
const struct inode_operations *i_op;
struct address_space *i_mapping;
struct address_space *i_data;
...
```

dentry

- Lưu trữ ánh xạ từ tên tệp đến inode tương ứng
  - tên tệp
  - tham chiếu tới inode
  - tham chiếu tới dentry cha (null nếu như là root của hệ thống tệp)
- Thường là cấu trúc dữ liệu tổng quát cho mọi hệ thống tệp
- Thường được cached trong bộ nhớ.

dentry

struct dentry

```
struct super_block *d_sb;
struct dentry *d_parent;
struct list_head d_subdirs;
struct dentry_operations *d_op;
unsigned char d_iname[];
struct inod *d_inode;
...
```

file

- Lưu trữ thông tin về tệp đang được mở
  - dentry và con trỏ tệp
  - mỗi tiến trình quản lý một bảng các tệp đang được mở, file handler trả về từ open() chính là index của tệp trong bảng này.
- Thường là cấu trúc dữ liệu tổng quát cho mọi hệ thống tệp
- Có reference count.
  - lệnh fork copy cả các file handles và nếu tiến trình con đọc file handle này, con trỏ tệp sẽ dịch chuyển
  - dup() sẽ tạo 2 entry trỏ tới cùng một cấu trúc file (tăng reference count).

file

struct file

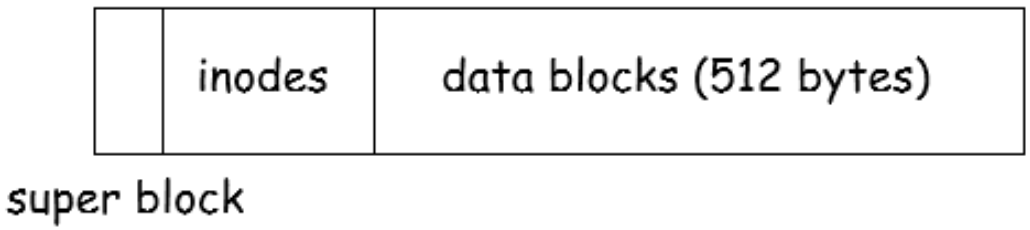
```
struct path f_path;
struct dentry (f_path.dentry);
const struct file_operations *f_op;
unsigned int f_flags;
fmode_t f_mode;
lodd_t f_pos;
...
```

Một số hệ thống tệp điển hình

- Unix File System
- Unix Fast File System
- Linux Second Extended File System

Unix File System

- Bell Labs
- Hệ thống tệp đơn giản
- Vấn đề: Chậm
  - Kích thước block nhỏ
  - Thứ tự block của cùng một tệp thường không có tổ chức
  - Thiếu tính cục bộ

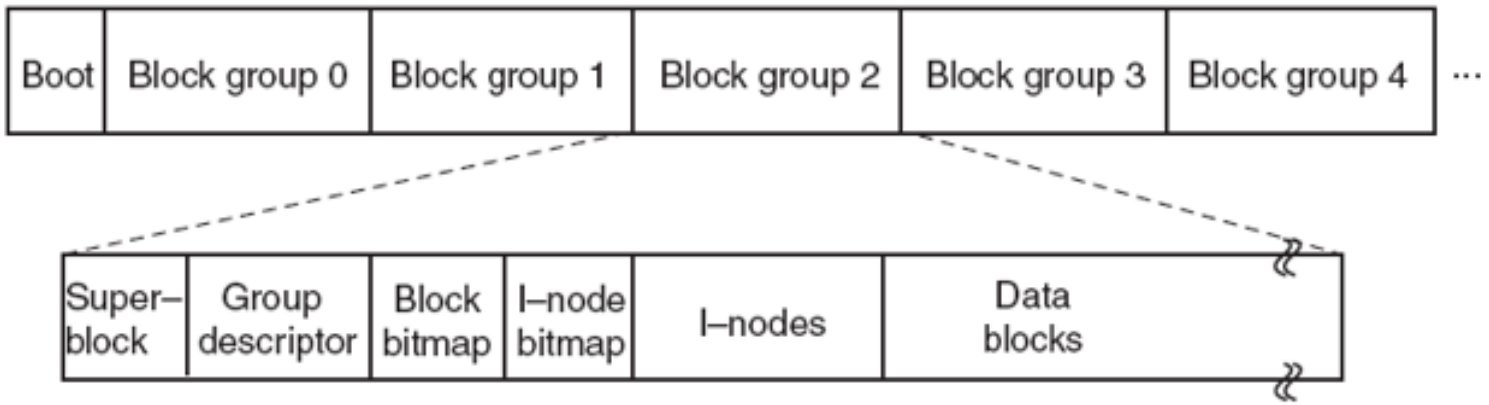


Unix Fast File System

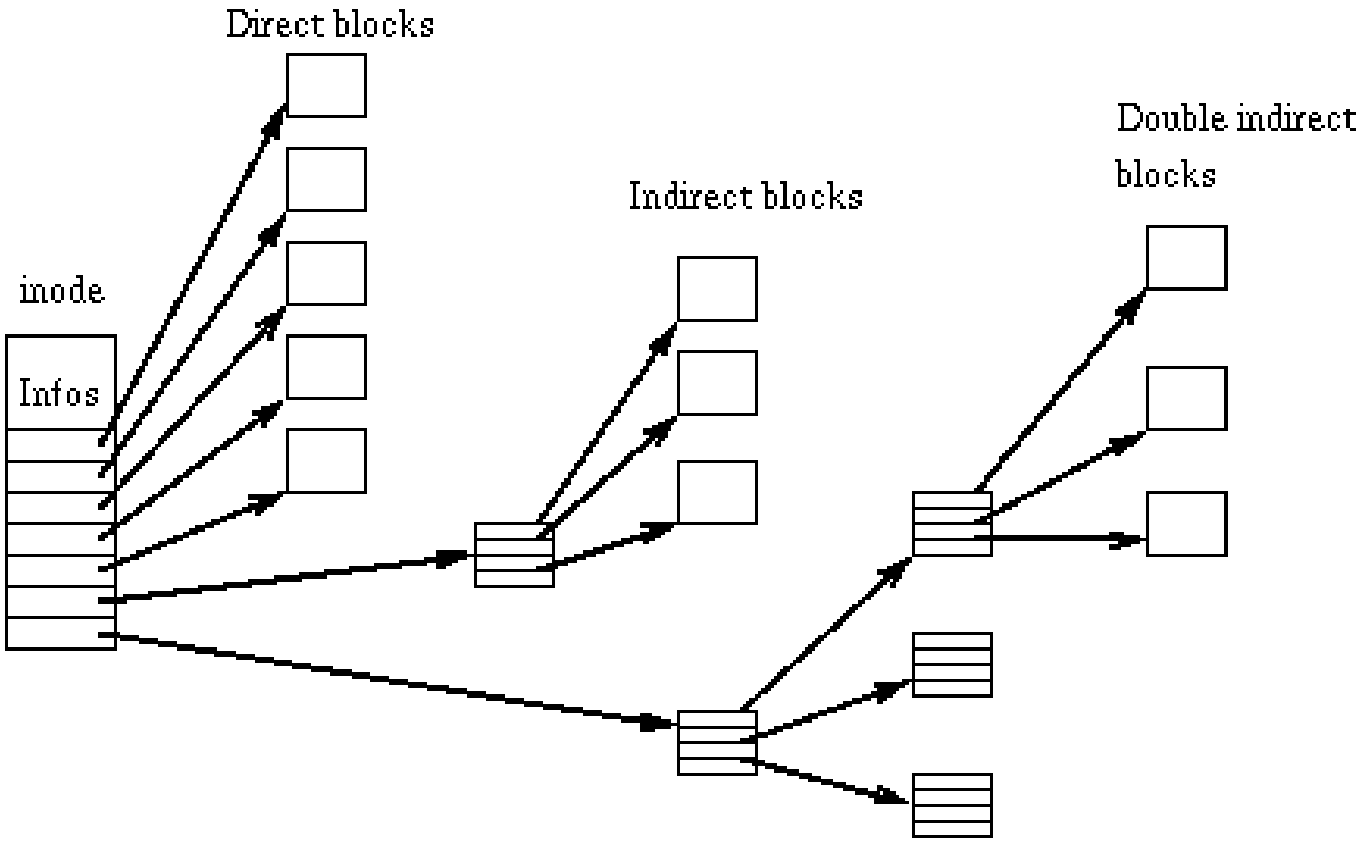
- BSD Fast File System
- Tăng kích thước block (4096)
- Lưu trữ bitmap của free block, để dễ dàng lựa chọn chuỗi block liên tiếp cho tệp
- Lưu trữ các dữ liệu liên quan gần nhau (vd. cùng một thư mục), các dữ liệu không liên quan được phân tán ra xa để dành chỗ cho các dữ liệu liên quan.

ext2

- Hệ thống tệp "chuẩn" của linux
- ext3 = ext2 + journaling
- Sử dụng layout lưu trữ gần giống FFS
- inode chứa các con trỏ (32bits) tới blocks.
  - direct, indirect, double indirect, triple indirect
  - Kích thước file lớn nhất 4.1TB (4k block)
  - Kích thước hệ thống tệp lớn nhất 16TB



ext2 inode


Direct blocks
Double indirect
blocks
Indirect blocks
inode
Infos