--

title: Welcome

tags: linux os, filesystem

--

**1. Khái niệm File System trong Linux**

Filesystem là các phương pháp và các cấu trúc dữ liệu mà một hệ điều hành sử dụng để theo dõi các tập tin trên ổ đĩa hoặc phân vùng. Có thể tạm dịch filesystem là một hệ thống tập tin. Đó là các tập tin được tổ chức trên ổ đĩa. Thuật ngữ này cũng được sử dụng để chỉ một phân vùng hoặc ổ đĩa được sử dụng để lưu trữ các tập tin hoặc loại hệ thống tập tin.

Sự khác biệt giữa một ổ đĩa hoặc phân vùng và hệ thống tập tin được lưu trên đó là rất quan trọng. Một vài chương trình (bao gồm cả chương trình tạo ra các hệ thống tập tin) hoạt động trực tiếp trên các sector thô của một ổ đĩa hoặc phân vùng. Nếu có một hệ thống tập tin tồn tại, nó sẽ bị phá hủy hoặc hỏng hóc.

Để một phân vùng hoặc ổ đĩa có thể được sử dụng như một hệ thống tập tin, nó cần được khởi tạo và các cấu trúc dữ liệu của kiểu hệ thống tập tin đó cần phải được ghi vào ổ đĩa. Quá trình này được gọi là tạo hệ thống tập tin.

Hầu hết các loại hệ thống tập tin UNIX đều có cấu trúc chung tương tự nhau, mặc dù các chi tiết cụ thể khác nhau khá nhiều. Các khái niệm chủ chốt là superblock, inode, data block, directory block và indirection block.

* **Superblock**: chưa các thông tin về hệ thống tập tin một cách tổng thể, chẳng hạn như kích thước của nó (thông tin chính xác ở đây phụ thuộc vào hệ thống tập tin)
* **Inode**: chứa tất cả các thông tin về một tập tin, ngoại trừ tên của nó. Tên được lưu trữ trong thư mục, cùng với số lượng lớn các inode. Mục nhập thư mục bao gồm tên tập tin và các số lượng inode đại diện cho tập tin đó. Inode chứa khối lượng lớn các khối dữ liệu, được sử dụng để lưu trữ dữ liệu trong tập tin.
* **Data block**: đây là nơi dữ liệu được lưu trữ. Vì một thư mục chỉ đơn giản là một tệp được định dạng đặc biệt, các thư mục cũng được chứa trong các khối dữ liệu. Một khối dữ liệu được cấp phát có thể thuộc về một và chỉ một tệp trong hệ thống. Nếu một khối dữ liệu không được cấp phát cho một tệp, thì nó là miễn phí sẵn sàng cấp phát khi cần.

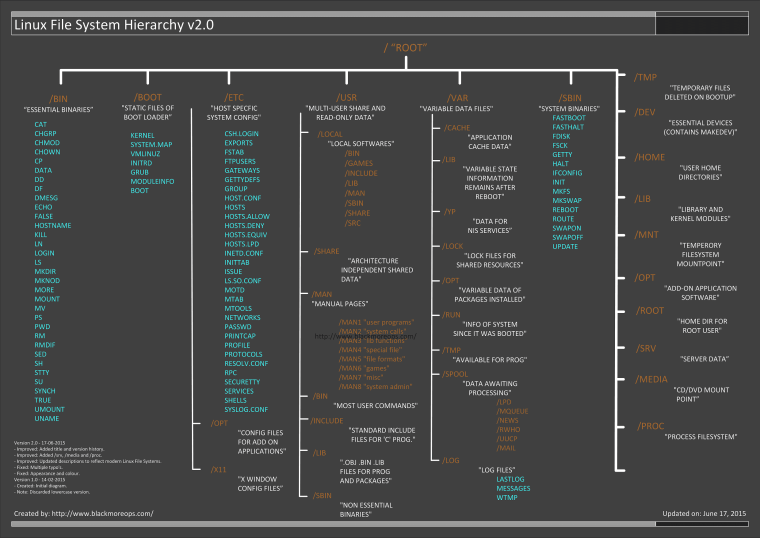
**2. Các kiểu File System trong hệ thống**

Linux có khá nhiều dạng file hệ thống, và mỗi loại sẽ được áp dụng với từng mục đích riêng biệt. Điều này không có nghĩa rằng những file hệ thống này không thể được áp dụng trong trường hợp khác, mà tùy theo nhu cầu, mục đích của người sử dụng, chúng ta sẽ đưa ra phương án phù hợp. Dưới đây là danh sách các filesystem phổ biến:

* **Minix**: là hệ thống lâu đời nhất và được cho là đáng tin cậy nhất, nhưng nó khá hạn chế về các tính năng (một số nhãn thời gian – time stamp bị thiếu, tối đa 30 kí tự / tệp tin)
* **Ext – Extended file system**: là định dạng file hệ thống đầu tiên được thiết kế dành riêng cho Linux. Có tổng cộng 4 phiên bản và mỗi phiên bản lại có một tính năng nổi bật. Phiên bản Ext đầu tiên là phần nâng cấp từ file hệ thống Minix được sử dụng tại thời điểm đó nhưng lại không đáp ứng được nhiều tính năng phổ biến ngày nay. Và tại thời điểm này, chúng ta không nên sử dụng Ext vì có nhiều hạn chế, không còn được hỗ trợ nhiều distribution.
* **Ext2**: thực chất không phải là file hệ thống journaling, được phát triển để kế thừa các thuộc tính của file hệ thống cũ, đồng thời hỗ trợ dung lượng ổ cứng lên 2TB. Ext2 không sử dụng journal cho nên sẽ có ít dữ liệu được ghi vào ổ đĩa hơn. Do lượng yêu cầu viết và xóa dữ liệu khá thấp nên nó phù hợp với các thiết bị lưu trữ gắn ngoài như thẻ nhớ, USB,…
* **Ext3**: về căn bản đây chỉ là Ext2 và đi kèm với journaling. Mục đích chính của Ext3 là tương tích ngược với Ext2, và do vậy, những ổ đĩa, phân vùng có thể dễ dàng được chuyển đổi giữa 2 chế độ mà không cần phỉa format như trước kia. Tuy nhiên, vấn đề về giới hạn của Ext2 vẫn còn nguyên trên Ext3. Ext3 không thực sự phù hợp để làm file hệ thống dành cho máy chủ bởi vì không hỗ trợ tính năng tạo disk snapshot và file được khôi phục sẽ rất khó để xóa bỏ sau này.
* **Ext4**: cũng giống như Ext3, lưu giữ được những ưu điểm và tính tương thích ngược với phiên bản trước đó. Như vậy, chúng ta có thể dễ dàng kết hợp các phân vùng định dạng Ext2, Ext3 và Ext4 trên cùng 1 ổ đĩa để tăng hiệu suất hoạt động.
* **BtrFS – Better FS**: được phát triển bởi Oracle và có nhiều tính năng giống ReiserFS. BtrFS hỗ trợ tính năng pool trên ổ cứng, tạo và lưu trữ snapshot, nén dữ liệu ở mức độ cao, chống phân mảnh dữ liệu nhanh chóng. Được thiết kế riêng dành cho các doanh nghiệp có quy mô lớn.
* **ReiserFS**: là một trong những bước tiến lớn nhất của file hệ thống Linux, lần đầu được công bố năm 2001 với nhiều tính năng mới mà file hệ thống Ext khó có thể đạt được. Năm 2004 được thay thế bởi Reiser4 với nhiều cải tiến nhưng do quá trình nghiên cứu, phát triển khá chậm chạp nên vẫn không hỗ trợ đầy đủ hệ thống kernel của Linux. Phù hợp với database và server email.
* **XFS**: được phát triển bởi Silicon Graphics năm 1994 để hoạt động với hệ điều hành riêng biệt của hãng, và sau đó chuyển sang Linux năm 2001. Khá tương đồng với Ext4 về một số mặt như: hạn chế được tình trạng phân mảnh dữ liệu, không cho phép các snapshot tự động kết hợp với nhau, hỗ trợ nhiều file dung lượng lớn, có thể thay đổi kích thước file dữ liệu,… nhưng không thể shrink – chia nhỏ phân vùng XFS. Phù hợp để áp dụng vào mô hình server media vì khả năng truyền tải file video rất tốt. Nếu hoạt động trên máy tính cá nhân thì đây không phải là lựa chọn tốt khi so sánh với Ext, vì hiệu suất hoạt động không khả thi, ngoài ra cũng không có tính năng gì nổi bật so với Ext3/4
* **JFS**: được phát triển bởi IBM năm 1990, sau đó chuyển sang Linux. Điểm mạnh rất dễ nhận thấy của JFS là tiêu tốn rất ít tài nguyên hệ thống, đạt hiệu suất hoạt động tốt với nhiều file dung lượng lớn, nhỏ khác nhau. Các phân vùng JFS có thể thay đổi kích thước được nhưng lại không thể shrink như ReiserFS và XFS, tuy nhiên, nó lại có tốc độ kiểm tra ổ đĩa nhanh nhất so với các phiên bản Ext.
* **Swap**: đây không thực sự là 1 dạng file hệ thống, bởi vì cơ chế hoạt động khá khác biệt, được sử dụng dưới 1 dạng bộ nhớ ảo và không có cấu trúc file hệ thống cụ thể. Không thể kết hợp và đọc dữ liệu được, nhưng lại chỉ có thể được dùng bởi kernel để ghi thay đổi vào ổ cứng. Thông thường, nó chỉ được sử dụng khi hệ thống thiếu hụt ram hoặc chuyển trạng thái của máy tính sang chế độ Hibernate.

**3. Filesystem Hierarchy Standard (FHS)**

Filesystem của hệ điều hành Linux được tổ chức theo tiêu chuẩn cấp bậc của hệ thống tập tin Filesystem Hierarchy Standard (FHS). Tiêu chuẩn này định nghĩa mục đích của mỗi thư mục.



Linux dùng ký tự “/” để tách các đường dẫn (khác với Windows sử dụng “\” để tách các đường dẫn). Tất cả các tập tin thư mục đều được bắt đầu từ thư mục gốc (/), cũng không có ký tự ổ đĩa giống Windows.

Chức năng của các thư mục:

* **/bin**: Các chương trình cơ bản
* **/boot**: chứa nhân Linux (kernel Linux) để khởi động và các filesystem maps cũng như các file khởi động giai đoạn 2.
* **/dev**: chứa các tập tin thiết bị (CDRom, HDD, FDD,…)
* **/etc**: chứa các tập tin cấu hình hệ thống.
* **/home**: thư mục dành cho người dùng không phải là root
* **/lib**: chứa các thư viện dùng chung cho các lệnh nằm trong /bin và /sbin. Thư mục này cũng chứa các module của kernel.
* **/mnt hoặc /media**: mout point mặc định cho những hệ thống file kết nối bên ngoài.
* **/opt**: thư mục chứa các phần mềm cài thêm
* **/sbin**: các chương trình hệ thống
* **/srv**: dữ liệu được sử dụng bởi các máy chủ lưu trữ trên hệ thống
* **/tmp**: thư mục chứa các file tạm thời
* **/usr**: thư mục chứa những file cố định hoặc quan trọng để phục vụ tất cả người dùng
* **/var**: dữ liệu biến được xử lý bởi daemon, điều này bao gồm các tệp nhật ký, hàng đợi, cache,…
* **/root**: các tệp cá nhân của người quản trị (root account)
* **/proc**: sử dụng cho nhân Linux, chúng được sử dụng bởi kernel để xuất dữ liệu sang không gian người dùng.