Chương 03. FileSystem_Phần 1

3.1. Tổ chức hệ thống tập tin (File system) Linux, File system Ext2

- Một hệ thống tập tin (File system) là các phương pháp và cấu trúc dữ liệu mà một hệ điều hành sử dụng để lưu trữ các thông tin của tập tin hay phần chia trên đĩa.
- Linux được thiết kế và thực thi một file system mới được chứa trong nhân của hệ điều hành Linux chuẩn. File system này được gọi là Second Extended File System (Ext2 fs)

3.1.1. Các đặc tính "tiêu chuẩn" của Ext2

- Hỗ trợ các kiểu tập tin chuẩn của Unix: các tập tin thường, thư mục, các tập tin dành riêng thiết bị và các symbolic link.
- Ext2 fs có khả năng quản lý những hệ thống tập tin được tạo trên những phần chia (partition) lớn.
  - *Mã nhân của hệ điều hành gốc bị giới hạn kích thước tối đa của hệ thống tập tin là 2GB, cơ cấu hiện tại trong lớp VFS (Virtual File System) đã nâng giới hạn này lên 4TB, do vậy có khả năng sử dụng đĩa có dung lượng lớn mà không cần thiết phải tạo nhiều partition*

- Ext2 fs hỗ trợ tên tập tin dài đến 255 ký tự. Giới hạn này có thể được nâng lên 1012.
- Ext2 dành riêng một số khối (block) cho người dùng cao cấp (root). Thông thường 5% số lượng các block là dành cho root. Điều này cho phép quản trị hệ thống khôi phục dễ dàng từ những tính huống mà xử lý của người dùng làm đầy hệ thống tập tin.

3.1.2. Một số đặc tính "nâng cao" của Ext2 fs

- Người dùng có thể thiết lập các thuộc tính cho một tập tin hay một thư mục. Trong trường hợp thiết lập thuộc tính cho thư mục, các tập tin mới được tạo trong thư mục sẽ thừa hưởng các thuộc tính này.
- Cho phép người quản trị hệ thống lựa chọn kích thước khối (block) logic khi tạo hệ thống tập tin. Kích thước khối có thể là 1024, 2048 và 4096. Sử dụng kích thước khối lớn có thể làm tăng tốc độ nhập xuất cần được thực hiện để truy nhập một tập tin do có ít yêu cầu nhập xuất hơn và do đó có ít thao tác truy tìm vị trí trên đĩa hơn nhưng sẽ lãng phí không gian đĩa hơn.

- Ext2 thực thi các symbolic link nhanh. Một symbolic link không sử dụng bất kỳ khối dữ liệu nào trên hệ thống tập tin. Nội dung của tập tin liên kết không được lưu trữ trong một khối dữ liệu mà được lưu trữ trong i-node của bản thân nó. Chính sách này có thể tiết kiệm được một số không gian đĩa (không cần cấp phát khối dữ liệu) và tăng tốc độ thao tác liên kết (không cần đọc khối dữ liệu khi truy nhập một liên kết. Chiều dài tối đa của tên tập tin liên kết trong một symbolic link là 60 ký tự
- Ext2 lưu trữ vết của trạng thái hệ thống tập tin có tác dụng kiểm tra hệ thống khi có các thay đổi.

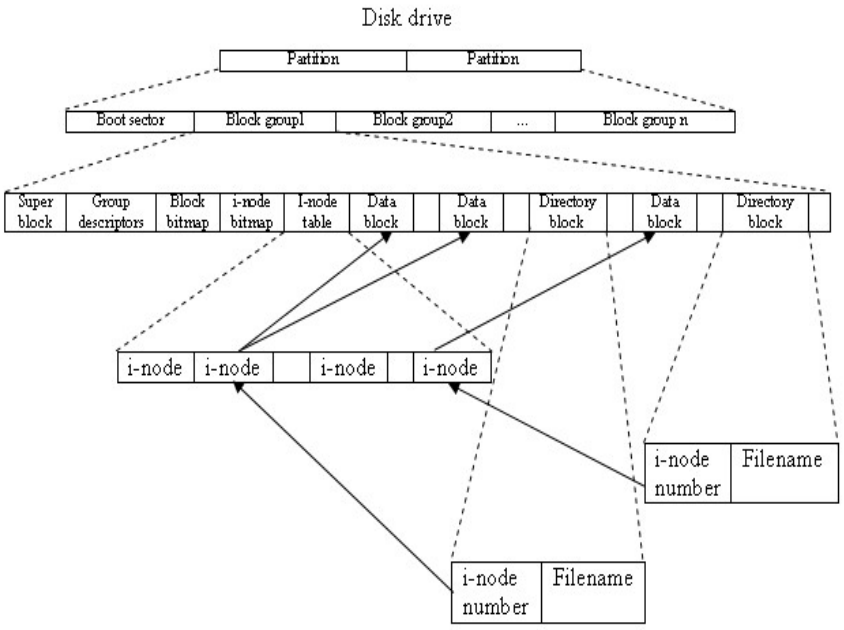
3.1.3. Cấu trúc vật lý

- Tất cả các cấu trúc dữ liệu được đặt kích cỡ dựa trên kích thước một block. Kích thước một block phụ thuộc vào kích thước của file system. Ví dụ: đĩa mềm kích thước block là 1KB (2 sector), kích thước block trên partition 10GB có thể là 4KB.
- File system được chia thành các block group, số lượng các block group cũng phụ thuộc vào kích thước của file system. Ví dụ: đĩa mềm chỉ có một block group, một partition 10GB có thể được chia thành 30 block group.

- Tại đầu mỗi block group có chứa các thông tin xác định vị trí, số block và các thông tin mô tả trạng thái hệ thống hiện hành. Các thông tin này bao gồm
  - **Super block**: Chứa các thông tin cơ bản nhất và các thuộc tính của file system. Ví dụ: tổng số i-node, tổng số khối, trạng thái của file system…
  - **Group descriptors**: Là một mảng cấu trúc, mỗi cấu trúc mô tả một block group, vị trí bảng i-node của nó, bảng đồ khối (block bitmap) và bảng đồ i-node (i-node bitmap)…

  - **Block bitmap**: Thường được đặt tại block đầu tiên của block group. Mỗi bit đại diện cho trạng thái hiện hành của một block trong block group. Trong đó giá trị 1 nghĩa là block đang được sử dụng và giá trị 0 là block chưa được sử dụng. Block đầu tiên của block group được đại diện bằng bit thứ 0 của byte thứ 0, block thứ 2 của block group được đại diện bằng bit thứ 1 của byte 0, block thứ 8 đại diện bằng bit thứ 7 của byte 0 và block thứ 9 của block group được đại diện bằng bit 0 của byte thứ 1…

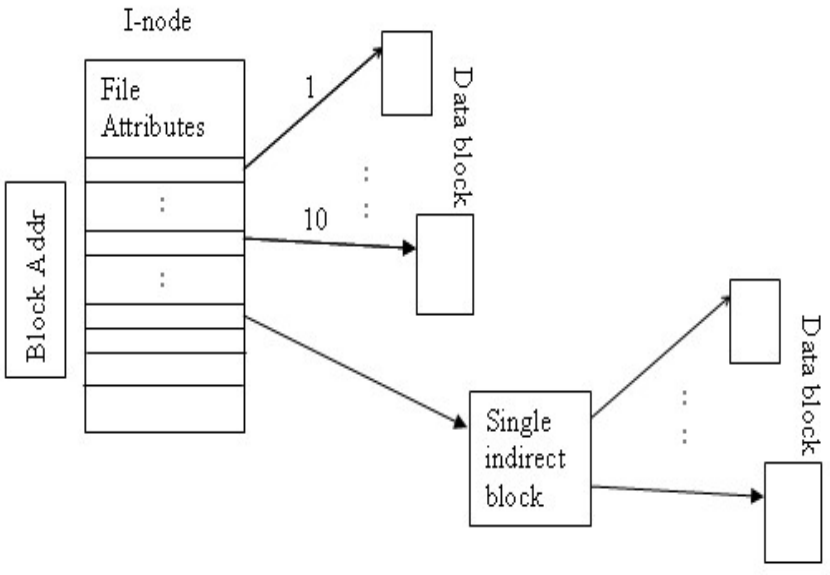
- **i-node bitmap**: Tương tự như block bitmap, mỗi bit đại diện cho một i-node trong i-node table. Mỗi một block group có một i-node bitmap.
- **i-node table**: Được sử dụng để lưu vết tất cả các tập tin: vị trí, kích thước, kiểu và quyền truy nhập của tập tin đều được lưu trữ trong các i-node. Tên tập tin không được lưu trữ trong i-node do trong bảng i-node tất cả các tập tin đều được tham chiếu bởi số i-node (i-node number). Số i-node là một chỉ số trong bảng i-node chỉ tới một cấu trúc i-node. Mỗi block group chứa một i-node table
- **Data block**: Được sử dụng để lưu trữ nội dung tập tin, các danh sách thư mục, các thuộc tính mở rộng, các symbolic link…



- **I-node**
  - Mỗi tập tin được đại diện bằng một cấu trúc gọi là i-node. I-node là một bảng có kích thước cố định được sử dụng để lưu trữ tất cả các thông tin về một tập tin và mỗi tập tin chỉ có một i-node duy nhất. Những thông tin này bao gồm: chủ nhân tập tin, thời điểm thay đổi nội dung tập tin, thời gian tập tin được truy nhập sau cùng, kích thước, các quyền trên tập tin, số lượng tập tin liên kết…

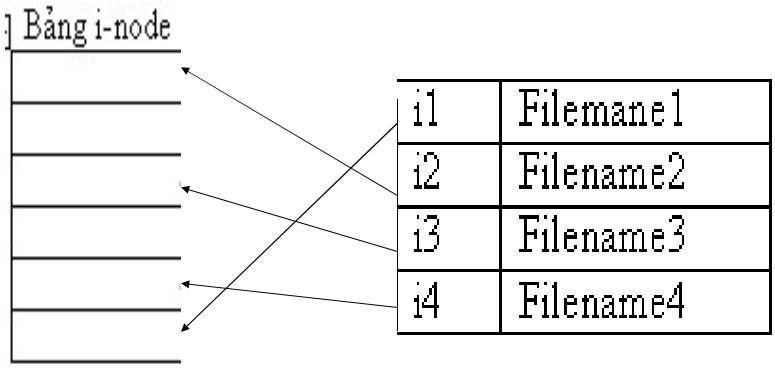
  - Địa chỉ của các khối dữ liệu đã cấp phát cho một tập tin được lưu trữ trong i-node của nó. I-node chứa một tập hợp các con trỏ. Những con trỏ này trỏ đến các khối dữ liệu của tập tin.
  - 10 con trỏ đầu tiên được tìm nạp từ đĩa đến bộ nhớ chính khi tập tin được mở. Đối với tập tin lớn hơn con trỏ tiếp theo trong i-node sẽ chỉ đến một block trên đĩa gọi là ***single indirect block***. Block này chứa các địa chỉ dữ liệu (block) bổ sung.

– Nếu vẫn không đủ, con trỏ thứ 12 trong i-node trỏ tới ***double indirect block***. Block này có chứa địa chỉ của block chứa danh sách các single indirect block. Mỗi single indirect block trỏ tới hàng trăm khối dữ liệu. Nếu vẫn không đủ một ***triple indirect block*** cũng có thể dược dùng…



Cấu trúc của i-node

- **Thư mục (directory)**
  - Các thư mục được cấu trúc trong một cây có thứ bậc. Mỗi một thư mục có thể chứa các tập tin và các thư mục khác.
  - Các thư mục được xây dựng như là một tập tin có kiểu đặc biệt. Thực tế, một thư mục là một tập tin có chứa một danh sách các mục từ, mỗi mục từ có cấu trúc đơn giản bao gồm hai trường: số i-node của tập tin và tên tập tin đó. Khi một tiến trình có tham chiếu đến thư mục, mã nhận hệ điều hành tìm kiếm trong các thư mục để tìm kiếm số i-node tương ứng. Sau khi tên đã được chuyển đổi thành số i-node, i-node được nạp vào trong bộ nhớ và được sử dụng bởi những yêu cầu tiếp theo.



Cấu trúc của thư mục

3.2. Hệ thống tập tin Ext3

- Bắt đầu từ phiên bản Red Hat Linux 7.2 hệ thống tập tin mặc định là Ext3.
- File system Ext3 là một phiên bản mới của Ext2, nó cải thiện một số đặc tính chính của Ext2 như sau:
  - **Tính sẵn có**: Nhờ vào nhật ký được cung cấp bởi file system Ext3, trong trường hợp hệ thống bị tắt đột ngột, Ext3 không cần thực hiện ngay thao tác kiểm tra tính nhất quán của hê thống khi file system được mount.

Ext3 chỉ kiểm tra tính nhất quán trong một số trường hợp đặt biệt khi có lỗi về đĩa cứng xảy ra. Thời gian để khôi phục một file system Ext3 không phụ thuộc vào kích thước của nó mà chỉ phụ thuộc vào kích thước của nhật ký được dùng để bảo trí tính nhất quán của hệ thống tập tin.

- **Tính toàn vẹn dữ liệu**: File system Ext3 cung cấp một cơ chế đảm bảo tính toàn vẹn dữ liệu trong trường hợp hệ thống bị tắt đột ngột. Ext3 cho phép người dùng chọn kiểu và mức độ bảo vệ dữ liệu

– **Tốc độ**: Ext3 có tốc độ cao hơn Ext2

– **Hỗ trợ phục hồi dữ liệu hiệu quả hơn ext2** (sử dụng chương trình e2fsck để phục hồi lại những hư hỏng dữ liệu)

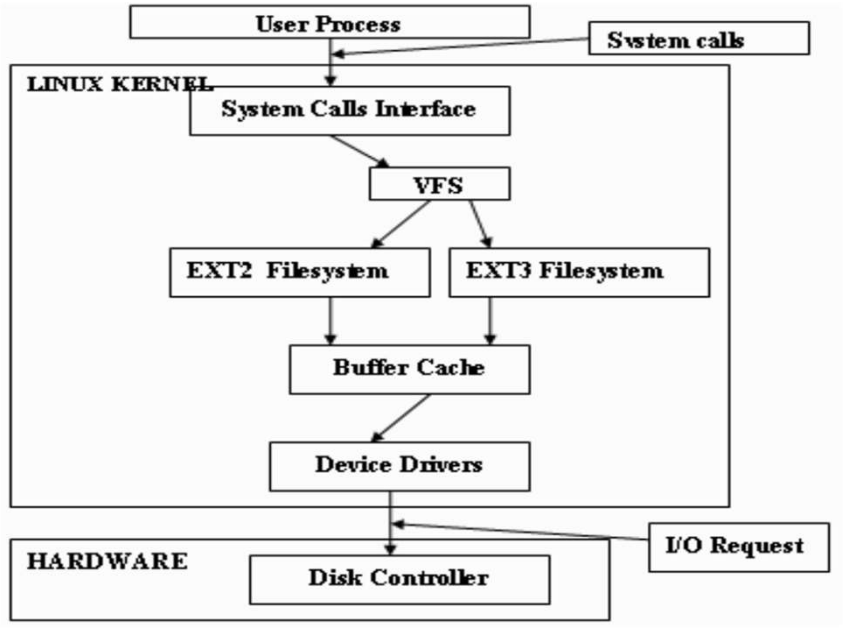
– **Dễ dàng chuyển đổi từ ext2 sang ext3**

3.3. Virtual File System-VFS

3.3.1. Nguyên lý

- Nhân hệ điều hành Linux có chứa một lớp hệ thống tập tin ảo (VFS). Được sử dụng trong suốt quá trình hệ thống gọi kích hoạt các tập tin. VFS là một lớp gián tiếp quản lý các lời gọi hệ thống tập tin và gọi các chức năng cần thiết trong mã hệ thống tập tin vật lý để thực hiện nhập xuất.
- Cơ chế gián tiếp này được sử dụng thường xuyên trong các hệ điều hành Linux để dễ dàng hoà nhập và sử dụng được nhiều kiểu file system khác nhau.

- Khi một tiến trình phát sinh lời gọi file system, nhân hệ điều hành sẽ gọi một chức năng được chứa trong VFS. Chức năng này quản lý các thao tác độc lập cấu trúc và chuyển hướng lời gọi tới một chức năng chứa trong mã hệ thống tập tin vật lý. Mã hệ thống tập tin sử dụng các chức năng lưu trữ bộ đệm (buffer cache) để yêu cầu nhập/xuất trên các thiết bị



3.3.2. Cấu trúc

- VFS định nghĩa một bộ các chức năng mà hệ thống tập tin được thực thi.
- VFS biết về các kiểu file system được hỗ trợ trong nhân của hệ điều hành bằng cách là sử dụng một bảng được định nghĩa trong quá trình cấu hình nhân của hệ điều hành. Mỗi mục từ trong hệ thống này mô tả một kiểu file system nó có chứa tên của kiểu file system và một con trỏ hàm đã gọi trong quá trình thao tác mount file system vào hệ thống.

- Khi một file system được mount hàm mount tương ứng được gọi. Chức năng của hàm này là đọc super block từ đĩa để khởi nạp các biến nội của nó và trả về một bộ mô tả file system đã mount cho VFS. Các hàm của VFS sử dụng bộ mô tả này để truy nhập file system.
- Bộ mô tả file system được mount chứa nhiều kiểu dữ liệu: kiểu dữ liệu thông tin là phổ biến với mọi file system, kiểu con trỏ tới hàm được cung cấp bởi mã nhân hệ điều hành file system vật lý và dữ liệu riêng được bảo quản bởi mã file system vật lý. Các con tỏ hàm trong bộ mô tả hệ thống tập tin cho phép VFS truy nhập file system.

3.4. Tổ chức logic của File system Linux

3.4.1. Các kiểu tập tin trong Linux

- **Regular**: Là tập tin chỉ chứa dữ liệu thông thường. Dữ liệu có thể là chương trình, tập tin dạng text, mã nguồn...
- **Directory**: Tương tự như kiểu regular, nó chứa một bộ các byte dữ liệu là danh sách các tập tin khác. Không có giới hạn về kiểu tập tin mà một thư mục có thể chứa.

- **Character device và block device**: Các chương trình Linux giao tiếp với các thiết bị phần cứng thông qua hai kiểu tập tin đặc biệt gọi là character device và block device. Các tập tin character device tham chiếu đến các trình điều khiển thiết bị muốn thực hiện các thao tác nhập/xuất bộ đệm như là terminal. Các tập tin block device được liên kết với các trình điều khiển thiết bị thực hiện các thao tác nhập xuất chỉ trong phạm vi những đoạn lớn 512 hay 1024 byte. Một số kiểu phần cứng như là đĩa có thể đại diện cho cả hai kiểu tập tin character device và block device.

- **Domain socket**: Được kết nối giữa các tiến trình, cho phép chúng truyền thông với nhau một cách nhanh chóng và tin cậy. Có nhiều kiểu khác nhau của domain socket, phần lớn chúng được sử dụng trong giao tiếp mạng.
- **Name pipes**: Cho phép truyền thông giữa hai tiến trình không có quan hệ chạy trên cùng một máy.
- **Hard link**: Thực sự không phải là một tập tin mà là một tập tin liên kết. Mỗi tập tin có ít nhất một hard link.

- Khi một hard link mới được tạo cho một tập tin, một tên hiệu (alias) cho tập tin đó sẽ được tạo ra. Nội dung của hard link và tập tin nó liên kết tới luôn giống nhau. Khi thay đổi nội dung của hard link, nội dung của tập tin mà nó liên kết tới cũng thay đổi theo và ngược lại.
- **Symbolic link**: Là một tập tin chỉ chứa tên của tập tin khác. Khi nhân hệ điều hành mở hoặc duyệt qua các symbolic link, nó được dẫn đến tập tin mà symbolic link chỉ đến thay vì chính bản thân symbolic link. Sự khác nhau quan trọng giữa symbolic link và hard link là hard link là tham chiếu trực tiếp trong khi đó symbolic link là một tham chiếu bởi tên tập tin.

3.4.2. Tổ chức cây thư mục

- Mọi hệ điều hành điều có một phương pháp để lưu trữ các thư mục và tập tin của chúng để tạo điều kiện cho các công tác bổ sung sửa đổi và các thay đổi khác về sau. Trong Linux mọi tập tin đều được lưu trữ trên hệ thống với một tên duy nhất, trong các thư mục cũng có thể chứa các tập tin và thư mục khác nữa.
- Hệ thống tập tin Linux là hệ thống có cấu trúc cây. Cây được đặt tại một vị trí gốc gọi là root, được biểu diễn bằng dấu /. Mọi phần tử trong cây hệ thống tập tin là các thư mục và tập tin chúng đều được nối tới root.

– ***Chú ý***: *Red Hat Linux sử dụng thuật ngữ root trong nhiều trường hợp khác nhau. Có tài khoản root (người có quyền làm mọi tác vụ trên hệ thống), có thư mục chủ của tài khoản đăng nhập root (/root), và thư mục root trong toàn thể file system (/).*

- Để mô tả một vị trí xác định trong cây hệ thống tập tin ta phải chỉ ra đường dẫn (path). Đường dẫn đến một vị trí có thể được định nghĩa như một đường dẫn tuyệt đối từ điểm root, hoặc như một đường dẫn tương đối được bắt đầu từ vị trí hiện hành.
- Khi chỉ ra một đường dẫn ta chỉ cần thực hiện "theo vết đường đi" trong cây hệ thống tập tin bằng cách liệt kê tuần tự những thư mục ta đi qua từ điểm này đến điểm khác. Mỗi thư mục được liệt kê trong thứ tự này phải được phân cách nhau bằng dấu /.

- Tổ chức cơ bản của hệ thống thư mục trong Linux bao gồm những thư mục chính sau:
- /
  - **dev**
  - **etc**
  - **lib**
  - **mnt**
  - **opt**
  - **proc**
  - **sbin**
  - **usr**
  - **var**

- **/dev**: Chứa những thư mục từ file system đại diện cho các thiết bị được gắng với hệ thống. Những tập tin này cần thiết cho hoạt động của hệ thống.
- **/etc**: Chứa những tập tin cấu hình của các dịch vụ trên máy tính.
- **/lib**: Chứa những thư viện cần thiết để thi hành các tập tin nhị phân được chứa đựng trong /sbin và /bin. Những thư viện được chia sẻ (dùng chung) này là rất quan trọng để khởi động hệ thống và thi hành các lệnh trong hệ thống tập tin root.

- **/mnt**: Tham chiếu và chứa các file system được mount tạm thời vào hệ thống. Ví dụ: CD-ROM, đĩa mềm…
- **/opt**: Chứa đựng các gói phần mềm ứng dụng tĩnh. Các gói phần mềm này được chứa đựng trong từng thư mục con riêng rẽ. Trong trường hợp các gói lớn chứa nhiều gói con nhỏ hơn, mỗi gói con này có thể thực hiện một chức năng hoàn chỉnh nào đó, mỗi gói con này cũng được lưu trữ riêng trong từng thư mục con trong thư mục của gói lớn. Với cách thức tổ chức này, người quản trị hệ thống sẽ dễ dàng xác định được vai trò của mỗi tập tin trong một gói cụ thể nào đó.

- **/proc**: Chứa các tập tin đặc biệt đại diện cho trạng thái hiện tại của nhân hệ điều hành.
- **/sbin**: Thư mục chứa các tập tin thi hành dành riêng cho người dùng root sử dụng và các tập tin cơ bản để khởi động hệ thống thêm vào cùng với các tập tin nhị phân có trong thư mục /bin.
- **/usr**: Chứa các tập tin có thể được dùng chung trên toàn hệ thống. Thư mục /usr thường được cài đặt riêng trên một partition độc lập và được mount vào root với quyền chỉ đọc. Trong usr có chứa nhiều thư mục con: bin chứa các tập tin thi hành, doc chứa các tài liệu văn bản, etc chứa các tập tin cấu hình, game chứa các trò chơi…

- **/usr/local**: Được dành cho người quản trị hệ thống sử dụng khi cài đặt phần mềm cục bộ. Thư mục này cần được bảo vệ để tránh bị ghi đè lên khi phần mềm hệ thống được cập nhật. Nó có thể được sử dụng cho các chương trình và dữ liệu cho phép dùng chung trên mạng mà không có trong /usr.
- **/var**: Chứa các tập tin dữ liệu khả biến như là tập tin, thư mục “đường ống”, dữ liệu nhật ký (log), các tập tin tạm thời…

3.5. Getting Started

3.5.1. Đăng nhập hệ thống

- Terminal hay PC có cài HĐH Unix/Linux và một tài (account) và mật khẩu (password)
- Đăng nhập (login)
- Những loại người dùng : superuser (root), user thường.
- Mật khẩu : phân biệt chữ hoa, thường; được mã hóa

3.5.2. Giao diện người sử dụng

- ***Dòng lệnh*** **(Shell prompt - command line)** : thường dùng bởi root để thao tác nhanh.
- ***Giao diện đồ họa*** : trực quan, dễ nhớ, là giao diện ưa thích của người dùng thường

3.5.3. Giao tiếp qua dòng lệnh

- Sau khi đăng nhập, Linux sẽ chuyển đến home directory của user và chạy một chương trình gọi là shell. Shell là giao tiếp giữa người sử dụng và HĐH hay nói cách khác shell là một chương trình được thiết kế để nhận lệnh của người sử dụng và thi hành lệnh.
  - Red Hat Linux có nhiều loại shell: sh (Bourne); Korn (ksh); C (csh); bash (Bourne Again Shell)…

- **Dấu nhắc shell (dấu nhắc đợi lệnh)**

  # khi ta là root (uperuser), ở bất kỳ shell nào

  % dấu nhắc khi chạy C shell

  $ dấu nhắc khi chạy Bourne shell hoặc Korn shell

  ~ dấu nhắc khi chạy tcsh shell

- Shell mặc định của Linux là bash, file bash trong /bin

- Nhận diện máy UNIX đang dùng : tên, phiên bản, tên máy, kiến trúc bộ vi xử lý, …

  uname –a

- Nhận diện shell đang làm việc : lệnh echo

  echo $SHELL

- **Thay đổi shell làm việc**
  - Thay đổi *vĩnh viễn* : dùng lệnh usermod

    usermod –s /bin/sh thuan
  - Thay đổi *tạm thời* (chuyển tạm thời qua shell khác) : **tên shell**

    $ bash

    [nam@localhost nam] **$ exit**

    (hoặc Ctrl-D)

    $

3.5.4. Giao diện đồ họa

- Có tên gọi khác nhau tùy thuộc HĐH:
  - Solaris : CDE (Common Desktop Environment)
  - Linux : X Window, phổ biến là KDE, GNOME
- X Window bao gồm : hệ thống cung cấp dịch vụ cửa sổ, thư viện đồ họa , …

- **Quy ước đặt tên cho tập tin**
  - Dài tối đa 255 ký tự
  - Phân biệt chữ hoa và chữ thường : taptin khác với Taptin
  - Tránh dùng các ký tự đặc biệt : *, ?, -, SPACE

3.6. Sử dụng các lệnh cơ bản

3.6.1 Dùng lệnh man để tìm trợ giúp

- **Cú pháp**

  man <command>

- **ví dụ**

  man useradd

3.6.2. Thêm user mới

- Muốn thực hiện lệnh này, người sử dụng phải là superuser (root)
- **Cú pháp**

  useradd <username>

- Một user mới tạo phải được gán password trước khi có thể đăng nhập bằng lệnh passwd

  passwd <username>

3.6.3. ls

- **ls** lists information about files on the system. By default ls shows you the names of unhidden files in the current directory. A common usage of ls is ls -la, which lists files in a longer format, and lists all files, including hidden files.
- **Cú pháp**

  ls [OPTION] [FILE]
- **Các Option**

-a: Liệt kê tất cả các tập tin có trong thư mục kể cả tập tin ẩn.

-l: Hiển thị thông tin chi tiết về nội dung tập tin.

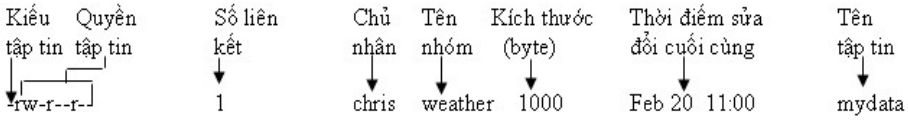
-R: Liệt kê cả nội dung của các thư mục con trong thư mục muốn xem.

-S: Sắp xếp danh sách tập tin theo kích thước giảm dần.

-t: Sắp xếp danh sách tập tin theo thứ tự ngày tháng thay đổi tập tin.

-F: Bổ sung thêm một ký tự vào cuối mỗi mục từ: /là thư mục; @ là symbolic link tới tập tin khác; * tập tin thi hành.

- Trong trường hợp liệt kê nội dung chi tiết thư mục. Kết quả bao gồm nhiều trường như sau:



- **Các ký tự mô tả kiểu tập tin**
  - Regular -
  - Directory d
  - Character device c
  - Block device b
  - Domain socket s
  - Name pipe p
  - Symbolic link l

- **3.6.4. Lệnh mkdir**

  Tạo thư mục mới
- **Cú pháp**

  mkdir [option] directory
- **Option -p**: Cho phép tạo cả cây thư mục (tạo cả thư mục cha nếu chưa có) như yêu cầu

3.6.5. Lệnh cd

Thay đổi thư mục hiện hành

- **Cú pháp**

cd directory
cd ..
cd /
pwd

3.6.6. Lệnh cp

Thực hiện sao chép các tập tin thư mục

- **Cú pháp**

  cp [option] source dest

  Sao chép tập tin *source* đến tập tin *dest*

  cp [option] source directory

  Sao chép nhiều tập tin từ thư mục source đến thư mục directory

- **Các option**

  -l: Tạo hard link cho các tập tin thay vì sao chép.

  -s: Tạo symbolic link cho các tập tin thay vì sao chép.

  -i: Xuất hiện lời nhắc trước khi ghi đè lên tập tin hiện có.

3.6.7. Lệnh mv

Thực hiện di chuyển hay đổi tên tập tin.

- **Cú pháp**

  mv [option] source dest

  Thực hiện đổi tên tập tin từ source thành dest

  mv [option] source …directory

  Thực hiện di chuyển các tập tin trong thư mục source đến thư mục directory

- **Các option**

  -i: xuất hiện thông báo trước khi ghi đè

  -f: không xuất hiện lời nhắc trước khi ghi đè.

3.6.8. Lệnh rmdir

Xóa thư mục rỗng

- **Cú pháp**

  rmdir [option] directory

- **Option**

  -p: Cho phép xóa bỏ cả cây thư mục (xóa bỏ các thư mục cha nếu nó rỗng)

3.6.9. Lệnh rm

Thực hiện xóa bỏ tập tin/ thư mục. Mặc định thì lệnh này không xóa các thư mục.

- **Cú pháp**

  rm [option] file/directory

- **Option**

  -r: xóa tất cả nội dung của các thư mục con

  -f: Không thông báo khi thực hiện xóa tập tin

  -i: Nhắc trước khi xóa.

3.6.10. Lệnh cat

- cat > <filename>
- cat <filename>...<filename>
- cat <filename1> ...<filenameN> > <filename>