

LỜI NÓI ĐẦU

Linux là tên gọi của một hệ điều hành máy tính và cũng là tên hạt nhân của hệ điều hành, nó có lẽ là một ví dụ nổi tiếng nhất của phần mềm tự do và của việc phát triển mã nguồn mở. Để góp phần giúp giáo viên và học sinh, sinh viên có thêm tư liệu trong việc học tập và giảng dạy. Giáo trình được xây dựng nhằm giúp học sinh, sinh viên hiểu được các khái niệm cơ bản, chức năng, nhiệm vụ của hệ điều hành nói chung, bước đầu làm quen và làm chủ được hệ điều hành Linux, cũng như việc phát triển các ứng dụng trên nền Linux. Từ đó, học sinh, sinh viên có nền tảng cơ bản để sẵn sàng cho các dự án phát triển ứng dụng trên Linux, hay bắt đầu nghề nghiệp quản trị hệ điều hành Linux sau khi ra trường. Đây là giáo trình Hệ điều hành Linux được biên soạn bởi nhóm tác giả nhằm giúp cho việc nghiên cứu, tìm hiểu về một hệ điều hành mới như Linux giúp cho chúng ta có một cái nhìn rộng hơn về tin học. Linux và các phần mềm mã nguồn mở cung cấp cho người sử dụng mã nguồn của chương trình.

Xuất phát từ thực tế nhu cầu học tập của học sinh, sinh viên chuyên ngành CNTT nói chung và học sinh, sinh viên trường Cao đẳng nghề Đắk Lắk nói riêng. Chúng tôi viết cuốn giáo trình này bám sát theo chương trình khung môn học/mô đun của trường, để phục vụ cho học sinh, sinh viên theo học Môn học/ mô đun “Hệ điều hành Linux”.

Do đang trong quá trình xây dựng giáo trình và bài giảng một cách khoa học và chuyên nghiệp. Vì vậy trong quá trình dịch và biên soạn tài liệu không tránh khỏi những thiếu sót. Rất mong được sự đồng góp ý kiến của người đọc để tài liệu ngày càng được hoàn chỉnh hơn. Nhóm biên soạn xin chân thành cảm ơn.

Nhóm biên soạn

MỤC LỤC

LỜI NÓI ĐẦU.....	1
MỤC LỤC.....	2
Chương 1: Giới thiệu hệ điều hành Linux	6
1.1. Lịch sử phát triển của Linux	6
1.2. Ưu nhược điểm của Linux	7
1.2.1. Những ưu điểm của Linux.....	7
1.2.2. Nhược điểm của Linux.....	8
1.3. Kiến trúc của hệ điều hành Linux	9
1.3.1. Hạt nhân	9
1.3.2. Shell	9
1.3.3. Các tiện ích	10
1.3.4. Chương trình ứng dụng	10
1.4. Các đặc tính cơ bản của Linux.....	10
1.4.1. Tốc độ cao.....	10
1.4.2. Bộ nhớ ảo.....	10
1.4.3. Sử dụng chung thư viện.....	10
1.4.4. Sử dụng chung chương trình xử lý văn bản.....	10
1.4.5. Sử dụng chung giao diện cửa sổ.....	10
1.4.6. Các tiện ích sao lưu dữ liệu	11
1.4.7. Hỗ trợ nhiều ngôn ngữ lập trình	11
Câu hỏi ôn tập chương.....	11
Chương 2: Cài đặt hệ điều hành Linux.....	12
2.1. Chuẩn bị cài đặt.....	12
2.1.1. Chọn cấu hình phần cứng.....	12
2.1.2. Dung lượng đĩa và bộ nhớ.....	12
2.1.3. Những cách cài đặt Linux	13
2.1.4. Phân vùng ổ đĩa cứng.....	13
2.1.5. Chọn cấu hình phần cứng.....	13
2.2. Tiến hành cài đặt	14
2.2.1. Các cách cài đặt.....	14
2.2.2. Trình tự cài đặt.....	14
2.2.3. Thiết lập cấu hình mạng	24
2.2.4. Thiết lập trương khoản người dùng.....	24
2.2.5. Thiết lập cấu hình xác thực	25
2.2.6. Chọn các gói phần mềm cài đặt.....	25
2.2.7. Thiết lập cấu hình X Window	26
2.2.8. Cài đặt các gói phần mềm	27
Câu hỏi ôn tập chương.....	28
Chương 3 Các khái niệm cơ bản trong Linux	29
3.1. Màn hình làm việc	29
3.1.1. Đăng nhập.....	29
3.1.2. Sử dụng shell prompt (Terminal)	29
3.1.3. Tạo account mới.....	29
3.1.4. Nautilus- File Manager	29
3.1.5. Start here	30
3.2. Giao diện đồ họa và giao diện dòng lệnh.....	30
3.2.1. Giao diện đồ họa	30

3.2.2. Giao diện dòng lệnh	34
3.3. Hệ thống tập tin	35
3.3.1. Các kiểu file có trong Linux.....	35
3.3.2. Quy ước tên file trong Linux.....	36
3.3.3. Cấu trúc hệ thống file của Linux	37
3.4. Cấu hình phần cứng.....	38
3.4.1. Cấu hình DHCP Server	38
3.4.2. Cấu hình Web Server	39
3.4.3. Network Card.....	39
3.5. Quản lý tiến trình.....	41
3.5.1. Khái niệm.....	41
3.5.2. Các lệnh cơ bản trong quản lý tiến trình.....	43
3.6. Tập tin và thư mục.....	46
3.6.1. Một số khái niệm.....	46
3.6.2. Các lệnh trong hệ thống tập tin	47
Câu hỏi ôn tập chương.....	50
Chương 4: Cài đặt phần mềm trong hệ điều hành Linux.....	51
4.1. RPM là gì?.....	51
4.1.1. Khái niệm.....	51
4.1.2. Quản lý gói.....	51
4.1.3. Đặc tính của RPM	51
4.2. Sử dụng công cụ dòng lệnh RPM	52
4.2.1. Lệnh rpm.....	52
4.2.2. Cài đặt phần mềm bằng rpm	52
4.2.3. Loại bỏ phần mềm đã cài đặt trong hệ thống.....	53
4.2.4. Nâng cấp phần mềm.....	53
4.2.5. Truy vấn các phần mềm	53
4.3. Cài đặt các gói dạng TAR.....	54
4.3.1. Chuẩn bị cài đặt.....	55
4.3.2. Tiến hành cài đặt	55
Câu hỏi ôn tập chương.....	56
Chương 5: Các ứng dụng phần mềm phổ biến trong Linux.....	57
5.1. Bộ phần mềm văn phòng OpenOffice	57
5.1.1. Hỗ trợ unicode	57
5.1.2. Open wrtiter (~Ms Word)	57
5.1.3. Open calc (~Ms Excel).....	57
5.1.4. Open base (~Ms Access).....	57
5.2. Các phần mềm Internet.....	58
5.2.1. Web Hosting	58
5.2.2. Gửi Và Nhận Thư từ Webmail	58
5.3. Các phần mềm Multimedia.....	60
5.3.1. Một số phần mềm multimedia.....	60
5.3.2. Hình ảnh giao diện của một số phần mềm multimedia.....	60
5.3.3. Trình xử lý ảnh The GIMP.....	61
5.4. Các phần mềm hệ thống	61
5.4.1. Shell thiết lập các tập tin	61
5.4.2. Cài đặt các file .rpm	61
Câu hỏi ôn tập chương.....	62

Chương 6: Giao diện dòng lệnh	63
6.1. Khái niệm Shell	63
6.1.1 Khái niệm.....	63
6.1.2. Phân loại	63
6.2. Một số lệnh cơ bản	63
6.2.1. Lệnh liên quan đến hệ thống	63
6.2.2. Lệnh thao tác trên tập tin.....	64
6.2.3. Lệnh khi làm việc trên terminal	64
6.3. Sử dụng phím tắt	65
6.3.1. Trong terminal.....	65
6.3.2. Trong GNOME	65
6.3.3. Trong OpenOffice	65
6.3.4. Trong vi (vim).....	65
6.4. Cú pháp lệnh.....	66
6.4.1. Tạo thư mục	66
6.4.2. Xóa thư mục với lệnh rmdir	66
6.4.3. Xem đường dẫn thư mục hiện thời với lệnh pwd	66
6.4.4. Lệnh đổi tên thư mục với lệnh mv	67
6.4.5. Tạo file với lệnh touch	67
6.4.6. Tạo file với lệnh cat	67
6.5. Cấu hình Shell	67
6.5.1. Thiết lập môi trường terminal	67
6.5.2. Thiết lập môi trường Shell	68
6.6. Lập trình Shell	68
6.6.1. Lệnh echo.....	68
6.6.2. Lệnh read	69
6.6.3. Sử dụng biến	69
Câu hỏi ôn tập chương.....	70
Chương 7: Hệ thống tập tin	71
7.1. Khái niệm hệ thống tập tin.....	71
7.1.1. Khái niệm.....	71
7.1.2. Một số nội dung liên quan đến tên file (bao gồm cả tên thư mục).....	71
7.2. Các công cụ tìm kiếm tập tin	72
7.2.1. Lệnh Find.....	72
7.2.2. Tìm kiếm với các tiêu chí khác	73
7.2.3. Khắc phục lỗi thường gặp	73
7.3. Di chuyển trong hệ thống tập tin	73
7.3.1. Sao chép file với lệnh cp	73
7.3.2. Di chuyển tới thư mục khác	75
7.3.3. Sao chép thư mục	75
7.3.4. Di chuyển thư mục	75
7.4. Quản lý ổ đĩa và phân vùng	75
7.4.1. Các lệnh quản lý ổ đĩa.....	75
7.4.2. Phân vùng	76
7.5. Bảo trì hệ thống tập tin	76
7.5.1. Quyền hạn	76
7.5.2. Lệnh chmod, chown, chgrp	78
Câu hỏi ôn tập chương.....	79

Chương 8: Quản lý người dùng	80
8.1. Khái niệm tài khoản người dùng	80
8.2. Hệ thống quản lý người dùng	80
8.2.1. Quyền truy nhập	80
8.2.2. Tạo tài khoản người dùng	82
8.2.3. Thay đổi thông tin của tài khoản	83
8.2.4. Tạm khóa tài khoản người dùng.....	83
8.2.5. Hủy tài khoản	83
8.3. Các lệnh quản lý người dùng	83
8.3.1. File /etc/passwd.....	83
8.3.2. Thêm người dùng với lệnh useradd.....	84
8.3.3. Thay đổi thuộc tính người dùng	85
8.3.4. Xóa bỏ một người dùng (lệnh userdel).....	86
8.4. Phân quyền trên hệ thống tập tin	86
Câu hỏi ôn tập chương.....	87
TÀI LIỆU THAM KHẢO	88

Chương 1: Giới thiệu hệ điều hành Linux

Mục tiêu:

- Trình bày sơ lược về lịch sử phát triển của Linux;
- Trình bày được kiến trúc của hệ điều hành Linux;
- Trình bày được một số đặc tính của hệ điều hành Linux;
- So sánh ưu nhược điểm của hệ điều hành Linux so với các hệ điều hành khác;
- Có thái độ nghiêm túc.

1.1. Lịch sử phát triển của Linux

Linux bắt nguồn từ một hệ điều hành lớn hơn có tên là Unix. Unix là một trong những hệ điều hành được sử dụng rộng rãi nhất thế giới do tính ổn định và khả năng hỗ trợ của nó. Ban đầu hệ điều hành Unix đã được phát triển như một hệ điều hành đa nhiệm cho các máy mini và các máy lớn (*mainframe*) trong những năm 70. Cho tới nay nó đã được phát triển trở thành một hệ điều hành phổ dụng trên toàn thế giới, mặc dù với giao diện chưa thân thiện và chưa được chuẩn hóa hoàn toàn.

Linux là phiên bản Unix được cung cấp miễn phí, ban đầu được phát triển bởi **Linus Torvald** năm 1991 khi còn là một sinh viên của trường đại học Helsinki Phần Lan. Hiện nay, Linus làm việc tại tập đoàn Transmeta và tiếp tục phát triển nhân hệ điều hành Linux (*Linux kernel*).

Khi Linus tung ra phiên bản miễn phí đầu tiên của Linux trên Internet, vô tình đã tạo ra một làn sóng phát triển phần mềm lớn nhất từ trước đến nay trên phạm vi toàn cầu. Hiện nay, Linux được phát triển và bảo trì bởi một nhóm hàng nghìn lập trình viên cộng tác chặt chẽ với nhau qua Internet. Nhiều công ty đã xuất hiện, cung cấp Linux dưới dạng gói phần mềm dễ cài đặt, hoặc cung cấp các máy tính đã cài đặt sẵn Linux.

Tháng 11 năm 1991, Linus đưa ra bản chính thức đầu tiên của Linux, phiên bản 0.02. Ở phiên bản này, Linus đã có thể chạy *bash* và *gcc* (*trình dịch C GNU*) nhưng mới chỉ dừng lại ở đó. Hệ thống chưa có các hỗ trợ người dùng và tài liệu hướng dẫn. Các số hiệu phiên bản không ngừng gia tăng cùng với việc bổ sung thêm các tính năng mới.

Sau ba năm nhân Linux ra đời, đến ngày 14-3-1994, hệ điều hành *Linux phiên bản 1.0* được phổ biến, đây là phiên bản tương đối ổn định. Thành công lớn nhất của Linux 1.0 là nó đã hỗ trợ giao thức mạng *TCP/IP* chuẩn Unix, sánh với giao thức *socket BSD* – tương thích cho lập trình mạng. Trình điều khiển thiết bị đã được bổ sung để chạy IP trên một mạng Ethernet hoặc trên tuyến đơn hoặc qua modem. Hệ thống file trong *Linux 1.0* đã vượt xa hệ thống file của Minix thông thường, ngoài ra đã hỗ trợ điều khiển SCSI truy nhập đĩa tốc độ cao. Điều khiển bộ nhớ ảo đã được mở rộng để hỗ trợ điều khiển trang cho các *file swap* và ánh xạ bộ nhớ của file đặc quyền (*chỉ có một ánh xạ bộ nhớ chỉ đọc được thi hành trong Linux 1.0*)

Vào tháng 3-1995, nhân 1.2 được phổ biến. Điều đáng kể của Linux 1.2 so với Linux 1.0 ở chỗ nó hỗ trợ một phạm vi rộng và phong phú phần cứng, bao gồm cả kiến trúc tuyến phần cứng PCI mới. Nhân Linux 1.2 là nhân kết thúc dòng nhân Linux chỉ hỗ trợ PC.

Một điều cần lưu ý về cách đánh chỉ số các dòng nhân Linux. Hệ thống chỉ số được chia thành một số mức, chẳng hạn hai mức như 2.4 hoặc ba mức 2.2.5. Trong cách đánh chỉ số như vậy, quy ước rằng với các chỉ số từ mức thứ hai trở đi, nếu là số chẵn thì dòng nhân đó đã khá ổn định và tương đối hoàn thiện, còn nếu là số lẻ thì dòng nhân đó vẫn đang được phát triển tiếp.

Tháng 6-1996, nhân *Linux 2.0* được phổ biến. Có hai đặc trưng nổi bật của *Linux 2.0* là hỗ trợ kiến trúc phức hợp, bao gồm cả cổng *Alpha 64-bit* đầy đủ, và hỗ trợ kiến trúc đa bộ xử lý. Phân phối nhân *Linux 2.0* cũng chỉ thi hành được trên bộ xử lý *Motorola 68000* và kiến trúc *SPARC* của *SUN*. Các thi hành của *Linux* dựa trên vi nhân *GNU Mach* cũng chạy trên *PC* và *PowerMac*.

Tới năm 2000, nhân *Linux 2.4* được phổ biến. Một trong đặc điểm được quan tâm của nhân này là nó hỗ trợ mã ký tự *Unicode 32 bit*, rất thuận lợi cho việc xây dựng các giải pháp toàn diện và triệt để đối với vấn đề ngôn ngữ tự nhiên trên phạm vi toàn thế giới.

Với phiên bản *Linux 2.2.6*, bạn có thể làm việc trên môi trường đồ họa với các ứng dụng cao cấp như: các tiện ích đồ họa và nhiều tiện ích khác. *Linux* khó có thể thành công được như hiện nay nếu không có các công cụ *GNU* của Tổ chức phần mềm miễn phí (*Free Software Foundation*). Trình dịch *gcc* của *GNU* đã giúp cho việc viết mã của *Linux* dễ dàng hơn rất nhiều. Thậm chí tổ chức này đã yêu cầu các bản *Linux* với các tiện ích kèm theo phải gọi là *GNU/Linux*.

Hệ điều hành *Berkley Unix (BSD)* cũng đóng một vai trò quan trọng đối với *Linux* trong việc làm cho hệ điều hành này trở nên phổ biến như hiện nay. Hầu hết các tiện ích đi kèm với *Linux* được chuyển sang từ *BSD*, đặc biệt là các công cụ về mạng và các tiện ích.

Hiện nay, *Linux* là một hệ điều hành *Unix* đầy đủ và độc lập. Nó có thể chạy *X Window*, *TCP/IP*, *Emacs*, *Web*, thư điện tử và các phần mềm khác. Hầu hết các phần mềm miễn phí và thương mại đều được chuyển lên *Linux*. Rất nhiều các nhà phát triển phần mềm đã bắt đầu chuyển sang viết trên *Linux*. Người ta đã thực hiện các phép đo *benchmarks* trên các hệ *Linux* và thấy rằng chúng thực hiện nhanh hơn khi thực hiện trên các trạm làm việc *Sun Microsystems* và *Compaq*, thậm chí nhiều khi còn nhanh hơn cả trên *Windows 98* và *Windows NT*. Thật khó có thể hình dung được hệ điều hành *Unix* “tí hon” này phát triển nhanh như thế nào.

1.2. Ưu nhược điểm của Linux

1.2.1. Những ưu điểm của Linux

Nếu bạn đã có máy tính trong tay, đầu tiên bạn phải có một hệ điều hành cài đặt trên đó bạn mới có thể sử dụng được các chương trình ứng dụng. Hệ điều hành là chương trình điều hành mọi hoạt động trong máy tính của bạn, mọi chương trình ứng dụng khác đều chạy trên nền của hệ điều hành này.

Sau đây là những lý do cho bạn lựa chọn hệ điều hành *Linux* cài đặt trên máy tính của mình:

- *Linux* là hệ điều hành mã nguồn mở, với nhiều tính năng giống các hệ điều hành khác và được cung cấp miễn phí cho người sử dụng.

- *Linux* đầy đủ: Tất cả những gì có ở *IBM*, *SCO*, *Sun*, ... đều có ở *Linux*, như: *C compiler*, *perl interpreter*, *shell*, *TCP/IP*, *Proxy*, *firewall*, tài liệu hướng dẫn, ...

- *Linux* rất mềm dẻo trong cấu hình, thông qua các tiện ích, dễ dàng sửa đổi ngay cả nhân. *Linux* là hệ điều hành linh động, tin cậy, an toàn và được tiếp tục phát triển với hàng ngàn lập trình viên trên toàn thế giới.

- *Linux* được trợ giúp. Tài liệu giới thiệu *Linux* ngày càng nhiều, không thua kém bất cứ một hệ điều hành nào khác. *Linux* được nhiều tổ chức và công ty lớn trên thế giới sử dụng: *IBM*, *HP*, *Cisco*, *Google*, *Amazon.com*, ...

Ngoài ra khi sử dụng hệ điều hành *Linux* các bạn còn có được các tính năng sau:

- *Tính ổn định*: *Linux* có tính ổn định cao, đây là một trong những ưu điểm của *Linux* so với các hệ điều hành khác. Tính ổn định ở đây có nghĩa là nó ít bị lỗi khi sử

dụng so với hầu hết các hệ điều hành khác. Người sử dụng Linux sẽ không phải lo lắng đến chuyện máy tính của mình bị hiện tượng “treo cứng” khi đang sử dụng nữa. Thông thường lý do để bạn bắt buộc phải khởi động lại hệ thống là do mất điện, nâng cấp phần cứng hoặc phần mềm. Ngay cả server Linux phục vụ những mạng lớn (hàng trăm máy trạm) cũng hoạt động rất ổn định.

- *Tính bảo mật*: Khi làm việc trên Linux người dùng có thể yên tâm hơn về tính bảo mật của hệ điều hành. Linux là hệ điều hành đa nhiệm, đa người dùng, điều này có nghĩa là nhiều người sử dụng có thể vào phiên làm việc của mình trên cùng một máy vào tại cùng một thời điểm. Linux cung cấp các mức bảo mật khác nhau cho người sử dụng. Mỗi người sử dụng chỉ làm việc trên một không gian tài nguyên dành riêng, chỉ có người quản trị hệ thống mới có quyền thay đổi trong máy.

- *Tính hoàn chỉnh*: Bản thân Linux đã được kèm theo các trình tiện ích cần thiết. Tất cả các trình tiện ích mà bạn mong đợi đều có sẵn ở một dạng tương đương rất giống. Trên Linux, các trình biên dịch như C, C++, ... các hạt nhân hay TCP/IP đều được chuẩn hoá.

- *Tính tương thích*: Linux tương thích hầu như hoàn toàn với một số chuẩn UNIX như *IEEE POSIX.1*, *UNIX System V* và *BSD UNIX*. Trên Linux bạn cũng có thể tìm thấy các trình giả lập của DOS và Windows cho phép bạn có thể chạy các ứng dụng quen thuộc trên DOS và Windows. Linux cũng hỗ trợ hầu hết các phần cứng máy PC.

- *Hệ điều hành 32 bit đầy đủ*: Ngay từ đầu Linux đã là hệ điều hành 32 bit đầy đủ. Điều đó có nghĩa là bạn không còn phải lo về các giới hạn bộ nhớ, các trình điều khiển EMM hay các bộ nhớ mở rộng, ... khi sử dụng Linux. Hiện nay đã có những phiên bản Linux 64 bits chạy trên máy *Alpha Digital* hay *Ultra Sparc*.

- *Dễ cấu hình*: Bạn không còn phải bận tâm về các giới hạn 640K và tiến hành tối ưu hoá bộ nhớ mỗi lần cài đặt một trình điều khiển mới. Linux cho bạn hầu như toàn quyền điều khiển về cách làm việc của hệ thống.

- *Khả năng làm việc trên nhiều loại máy*: Cấu hình phần cứng tối thiểu mà Linux cần chỉ là chip 80386, 2MB bộ nhớ, 10-20 MB không gian đĩa để bắt đầu. Khi bạn càng bổ sung phần cứng thì Linux chạy càng nhanh. Linux có khả năng chạy trên nhiều dòng máy khác nhau như *Apple Macintosh*, *Sun*, *Dec Alpha* và *Power PC*.

1.2.2. Nhược điểm của Linux

- Người dùng phải thành thạo:

Trình tự cài đặt tự động, giao diện thân thiện với người dùng chỉ giảm nhẹ phần nào sự phức tạp trong quá trình cài đặt phần mềm, tinh chỉnh màn hình, card âm thanh, card mạng, ... Đôi khi những công việc này bắt buộc bạn phải thao tác từ những dòng lệnh cực kỳ "bí hiểm", nhầm chán và rất dễ nhầm lẫn (Trong khi HĐH Windows thì chỉ cần theo thông báo rồi Next...next... I agree...next... Finish là xong). Để cài đặt thành công, đôi khi bạn phải bỏ ra rất nhiều thời gian để tham khảo và nghiên cứu tài liệu. Vấn đề chính là nằm ở chỗ chúng ta dùng máy vi tính, hầu hết ta đều dùng hệ điều hành Windows, vì vậy, ta mang trong mình nặng tư tưởng của windows.

- Phần cứng ít được hỗ trợ.

Tuy đã cố gắng rất nhiều, nhưng bộ Linux của họ vẫn chưa có đủ trình điều khiển cho tất cả các thiết bị phần cứng có trên thị trường.

- Phần mềm ứng dụng chưa tinh xảo.

Các phần mềm ứng dụng trên Linux rất sẵn và không mất tiền, nhưng đa số không tiện dụng, không phong phú. Nhiều phần mềm ứng dụng thiếu các chức năng thông dụng, đặc trưng mà người dùng đã quen với Microsoft Office.

- **Thiếu chuẩn hóa.**

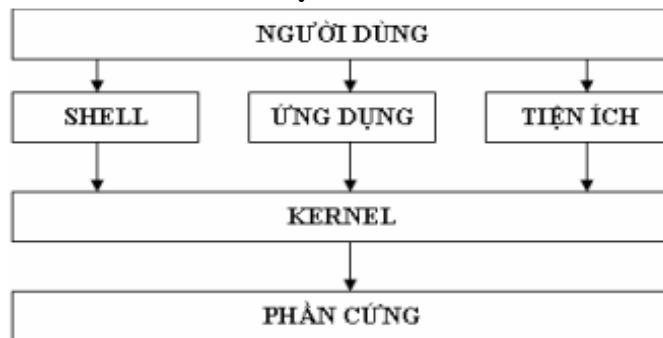
Do Linux toàn miễn phí nên bất cứ ai thích đều có thể tự mình đóng gói, phân phối theo các cửa mình. Có hàng chục nhà phân phối khác nhau trên thị trường, người dùng trước khi cài đặt thường phải tự mình so sánh để tìm ra sản phẩm thích hợp.

Vì linux thuộc dự án phần mềm mã nguồn mở nên bất cứ ai cũng có thể tự mình đóng góp, chỉnh sửa và thậm chí là phân phối lại.

- **Chính sách hỗ trợ khách hàng thiếu nhất quán và tốn kém.**

Mặc dù không phải trả bản quyền, nhưng người dùng vẫn phải trả phí cho mỗi thắc mắc cần được giải đáp từ nhà phân phối, cho dù đó chỉ là gọi qua điện thoại.

1.3. Kiến trúc của hệ điều hành Linux



1.3.1. Hạt nhân

Là trung tâm điều khiển của hệ điều hành Linux, chứa các mã nguồn điều khiển hoạt động của toàn bộ hệ thống. Hạt nhân được phát triển không ngừng, thường có 2 phiên bản mới nhất, một bản đang phát triển mới nhất và một bản ổn định mới nhất. Kernel được thiết kế theo dạng modul, do vậy kích thước thật sự của Kernel rất nhỏ. Chúng chỉ tải những bộ phận cần thiết lên bộ nhớ, các bộ phận khác sẽ được tải nếu có yêu cầu sử dụng. Nhờ vậy so với các hệ điều hành khác Linux không sử dụng lãng phí bộ nhớ nhờ không tải mọi thứ lên mà không cần quan tâm nó có sử dụng không.

Kernel được xem là trái tim của hệ điều hành Linux, ban đầu nhân được phát triển cho các CPU Intel 80386. Điểm mạnh của loại CPU này là khả năng quản lý bộ nhớ. Kernel của Linux có thể truy xuất tới toàn bộ tính năng phần cứng của máy. Yêu cầu của các chương trình cần rất nhiều bộ nhớ, trong khi hệ thống có ít bộ nhớ, hệ điều hành sử dụng không gian đĩa hoán đổi (*swap space*) để lưu trữ các dữ liệu xử lý của chương trình. *Swap space* cho phép ghi các trang của bộ nhớ xuất các vị trí dành sẵn trong đĩa và xem nó như phần mở rộng của vùng nhớ chính. Bên cạnh sử dụng *swap space*, Linux còn hỗ trợ các đặc tính sau:

- Bảo vệ vùng nhớ giữa các tiến trình, điều này không cho phép một tiến trình làm tắt toàn bộ hệ thống.

- Chỉ tải các chương trình khi có yêu cầu.

1.3.2. Shell

Shell cung cấp tập lệnh cho người dùng thao tác với kernel để thực hiện công việc. Shell đọc các lệnh từ người dùng và xử lý. Ngoài ra shell còn cung cấp một số đặc tính khác như: chuyển hướng xuất nhập, ngôn ngữ lệnh để tạo các tập tin lệnh tương tự tập tin *bat* trong DOS.

Có nhiều loại shell được dùng trong Linux. Điểm quan trọng để phân biệt các shell với nhau là bộ lệnh của mỗi shell. Ví dụ, *C shell* thì sử dụng các lệnh tương tự ngôn ngữ C, *Bourne Shell* thì dùng ngôn ngữ lệnh khác.

Shell sử dụng chính trong Linux là *GNU Bourne Again Shell (bash)*. Shell này là shell phát triển từ *Bourne Shell*, là shell sử dụng chính trong các hệ thống Unix, với nhiều tính năng mới như: điều khiển các tiến trình, các lệnh history, tên tập tin dài,

1.3.3. Các tiện ích

Các tiện ích được người dùng thường xuyên sử dụng. Nó dùng cho nhiều thứ như thao tác tập tin, đĩa, nén, sao lưu tập tin, ... Tiện ích trong Linux có thể là các lệnh thao tác hay các chương trình giao diện đồ họa. Hầu hết các tiện ích dùng trong Linux là sản phẩm của chương trình GNU. Linux có sẵn rất nhiều tiện ích như trình biên dịch, trình gỡ lỗi, soạn văn bản, ... Tiện ích có thể được sử dụng bởi người dùng hoặc hệ thống. Một số tiện ích được xem là chuẩn trong hệ thống Linux như *passwd*, *ls*, *pa*, *vi* ...

1.3.4. Chương trình ứng dụng

Khác với các tiện ích, các ứng dụng như chương trình word, hệ quản trị cơ sở dữ liệu, ... là các chương trình có độ phức tạp lớn và được các nhà sản xuất viết ra như:

- Văn phòng (open office)
- Giải trí (movie player, xmms, totem player kaffeine, ...)
- Xử lý ảnh (GIMP)
- Dịch vụ mạng (Telnet, SSH, FTP, Postfix, Apache, Bind, CUPS,
- OpenLDAP, Iptable, Squid, Mozilla-Firefox, SAMBA, NFS)
- Cơ sở dữ liệu (MySQL, PostgreSQL)
- Lập trình (Emacs, C/C++, QT Trolltech, Fortran, Java, R, octave, Lapack, Blas, Python, Perl, AWK, TCL/TK, PHP, ...)
- Quản trị hệ thống (Webmin, VNC, ...), ...

1.4. Các đặc tính cơ bản của Linux

Linux hỗ trợ các tính năng cơ bản thường thấy trong các hệ điều hành Unix và nhiều tính năng khác mà không hệ điều hành nào có được. Linux cung cấp môi trường phát triển một cách đầy đủ bao gồm các thư viện chuẩn, các công cụ lập trình, trình biên dịch, debug, ... như bạn mong đợi ở các hệ điều hành Unix khác. Hệ thống Linux trội hơn các hệ thống khác trên nhiều mặt mà người dùng quan tâm như sự phát triển tốc độ, dễ sử dụng và đặc biệt là sự phát triển và hỗ trợ mạng. Một số đặc điểm của Linux chúng ta cần quan tâm:

1.4.1. Tốc độ cao

Hệ điều hành Linux được biết đến như một hệ điều hành có tốc độ xử lý cao, bởi vì nó thao tác rất hiệu quả đến tài nguyên như: bộ nhớ, đĩa, ...

1.4.2. Bộ nhớ ảo

Khi hệ thống sử dụng quá nhiều chương trình lớn dẫn đến không đủ bộ nhớ chính (RAM) để hoạt động. Trong trường hợp đó, Linux dùng bộ nhớ từ đĩa vào *partition swap*. Hệ thống sẽ đưa các chương trình hoặc dữ liệu nào chưa có yêu cầu truy xuất xuống vùng swap này, khi có nhu cầu thì hệ thống chuyển lên lại bộ nhớ chính.

1.4.3. Sử dụng chung thư viện

Hệ thống Linux có rất nhiều thư viện dùng chung cho nhiều ứng dụng. Điều này sẽ giúp hệ thống tiết kiệm được tài nguyên cũng như thời gian xử lý.

1.4.4. Sử dụng chung chương trình xử lý văn bản

Chương trình xử lý văn bản là một trong những chương trình rất cần thiết đối với người sử dụng. Linux cung cấp nhiều chương trình cho phép người dùng thao tác với văn bản như *vi*, *emacs*, *nroff*.

1.4.5. Sử dụng chung giao diện cửa sổ

Giao diện cửa sổ dùng hệ thống *X Window*, có giao diện như hệ điều hành Windows. Với hệ thống này người dùng rất thuận tiện khi làm việc trên hệ thống. *X Window System* hay còn gọi tắt là X được phát triển tại viện *Massachusetts Institute of*

Technology. Nó được phát triển để tạo ra môi trường làm việc không phụ thuộc phần cứng. X chạy dưới dạng *client – server*. Hệ thống X Window hoạt động qua hai bộ phận:

- Phần server còn gọi là *X server*.
- Phần client được gọi là *X Window manager* hay *desktop environment*.

X Server sử dụng trong hầu hết các bản phân phối của Linux là *Xfree86*. Client sử dụng thường là KDE (*K Desktop Environment*) và GNOME (*GNU Network Object Model Environment*).

Dịch vụ Samba sử dụng tài nguyên đĩa, máy in với Windows. Tên Samba xuất phát từ giao thức *Server Message Block* (SMB) mà Window sử dụng để chia sẻ tập tin và máy in. Samba là chương trình sử dụng giao thức SMB chạy trên Linux. Sử dụng Samba bạn có thể chia sẻ tập tin và máy in với các máy Windows

1.4.6. Các tiện ích sao lưu dữ liệu

Linux cung cấp các tiện ích như *tar*, *cpio* và *dd* để sao lưu và backup dữ liệu. Red Hat Linux còn cung cấp tiện ích *Backup and Restore System Unix* (BRU) cho phép tự động backup dữ liệu theo lịch.

1.4.7. Hỗ trợ nhiều ngôn ngữ lập trình

Linux cung cấp một môi trường lập trình Unix đầy đủ bao gồm các thư viện chuẩn, các công cụ lập trình, trình biên dịch, chương trình debug mà bạn có thể tìm thấy trong các hệ điều hành Unix khác. Ngôn ngữ chủ yếu sử dụng trong các hệ điều hành Unix là C và C++. Linux dùng trình biên dịch cho C và C++ là *gcc*, chương trình biên dịch này rất mạnh, hỗ trợ nhiều tính năng. Ngoài C, Linux cũng cung cấp các trình biên dịch, thông dịch cho các ngôn ngữ khác như *Pascal*, *Fortan*, *Java*, ...

Câu hỏi ôn tập chương

1. Hãy nêu quá trình phát triển của hệ điều hành Linux? Lấy ví dụ minh họa về các nước trên thế giới đang sử dụng hệ điều hành này?
2. Hãy cho biết ưu điểm của hệ điều hành Linux? Nhược điểm khi sử dụng hệ điều hành này?
3. Vẽ sơ đồ kiến trúc chung và nêu kiến trúc của Linux Redhat?
4. Cho biết các đặc tính của hệ điều hành Linux? Liên hệ thực tế?

Chương 2: Cài đặt hệ điều hành Linux

Mục tiêu:

- Trình bày được yêu cầu phần cứng của máy tính cài hệ điều hành Linux;
- Biết qui trình cài đặt hệ điều hành Linux;
- Biết cách phân vùng ổ đĩa;
- Chọn lựa được phương pháp cài đặt nhanh và chính xác nhất;
- Thiết lập được chế độ về cấu hình mạng, tài khoản và xác thực người dùng;
- Xác định gói phần mềm Linux nào cần sử dụng trong quá trình cài đặt;
- Có thái độ cẩn thận, tỉ mỉ, chính xác.

2.1. Chuẩn bị cài đặt

2.1.1. Chọn cấu hình phần cứng

Linux không đòi hỏi máy có cấu hình mạnh. Tuy nhiên nếu phần cứng có cấu hình thấp quá thì có thể không chạy được X-Window hay các ứng dụng có sẵn. Cấu hình tối thiểu nên dùng:

- CPU: Pentium MMX trở lên.
- RAM: 64MB trở lên cho Text mode, 192MB cho Graphics mode.
- Đĩa cứng: Dung lượng đĩa còn phụ thuộc vào loại cài đặt.
 - Custum Installation (mininum): 520MB
 - Server (minimum): 870MB
 - Personal Desktop: 1.9GB
 - Workstation: 2.4GB
 - Custum Installation (everything): 5.3GB
- 2MB cho card màn hình nếu muốn sử dụng chế độ đồ họa.

2.1.2. Dung lượng đĩa và bộ nhớ

- *Dung lượng đĩa:*

Sau khi có bìa điều khiển thích hợp cho ổ đĩa rồi, bạn phải quan tâm đến các yêu cầu về dung lượng ổ đĩa. Linux chấp nhận một lúc nhiều ổ đĩa cứng và có thể cài đặt nó không cùng trên một ổ duy nhất.

Muốn sử dụng Linux cho có hiệu quả, bạn phải phân vùng lại ổ đĩa cứng và cấp phát đủ dung lượng đĩa cho các tệp hệ thống Linux và cho các tệp dữ liệu của bạn. Phân vùng (Partition) là chia những vùng theo ý của người sử dụng khi bắt đầu thiết lập thông số cho ổ đĩa cứng và trước khi định dạng ổ đĩa cứng.

Dung lượng đĩa cần thiết tùy thuộc vào phần mềm bạn sẽ cài đặt và số lượng dữ liệu mà phần mềm ấy sinh ra. So với hầu hết các hệ điều hành kiểu UNIX, Linux đòi hỏi dung lượng đĩa ít hơn. Bạn có thể chạy toàn bộ hệ Linux (không có phần X Window-tức là chỉ ở Text mode) với chỉ 80 MB (bản kernel 2.2.4-10). Nếu cài đặt không sót một thứ gì trong bản phát hành, bạn sẽ cần từ 1.8 GB đến 3.5 GB tùy theo phiên bản và nhà sản xuất.

Thông thường lệnh DOS fdisk hoặc một vài thương phẩm khác cho phép bạn phân chia lại ổ đĩa cứng và Linux cũng có tiện ích tương tự gọi là FIPS. Chú ý: Nếu bạn cài đặt Linux vào một ổ đĩa cứng mới nguyên thì không sao, còn đối với ổ đang dùng thì phải phân vùng và định dạng lại. Việc này sẽ xóa sạch toàn bộ thông tin trên ổ đĩa cứng, do đó bạn phải sao lưu cẩn thận trước khi cài đặt Linux. Nếu ổ đĩa cứng có dung lượng lớn, bạn có thể phân thành nhiều vùng và sao chép thông tin trở lại vào các vùng đã khai báo.

- *Bộ nhớ*

Linux không đòi hỏi nhiều RAM, nhất là khi so sánh với các hệ điều hành khác như Windows 2000, XP hoặc Windows NT.

Theo kernel và HOW-TO phiên bản ngày 11-7-2001, Linux chỉ cần 2 MB RAM, nhưng trong thực tế sử dụng thì Linux cần ít nhất 4 MB RAM. Thật sự, cấu hình thấp với 4 MB RAM chỉ có thể chạy ở chế độ văn bản, không có giao diện đồ họa. Từ phiên bản RedHat v.7.2, bạn cần ít nhất là 64 MB RAM và hiện nay 128 MB là yêu cầu trung bình. Nếu có ít hơn 4 MB RAM, bạn phải chạy với tệp hoán chuyển (swap file) ở trên đĩa cứng, được dùng như bộ nhớ ảo và do đó làm chậm hệ thống. Lượng RAM cần thiết còn phụ thuộc vào việc bạn sử dụng máy để làm gì. Càng muốn có nhiều chức năng bạn càng cần thêm RAM. Khi bạn dùng máy để quản trị một cơ sở dữ liệu thì lượng RAM cần thiết sẽ tăng lên rất nhiều.

Việc sử dụng giao diện đồ họa X Window (bằng phần mềm XFree86) làm tăng nhu cầu về bộ nhớ. Bạn cần ít nhất 8 MB RAM vật lý và 8 MB tệp hoán chuyển, tức là 16 MB RAM ảo để có một hệ thống hoạt động hiệu quả.

2.1.3. Những cách cài đặt Linux

Có thể cài đặt Linux bằng một trong các cách sau:

- Từ ổ đĩa CD-ROM.
- Từ bản sao chép Linux trên ổ đĩa cứng hoặc USB.
- Cài đặt từ server mạng, sử dụng HTTP, FTP, hoặc NFS, ...
- Phương pháp cài đặt từ Windows bằng **Wubi**

2.1.4. Phân vùng ổ đĩa cứng

- Điều này cho phép bạn tạo các phân vùng mới trên không gian trống của đĩa, hoặc để sử dụng các phân vùng Linux đã tồn tại.
- Cấp phát không gian trao đổi thích hợp.
- Xác định hệ thống file nào để sử dụng.

• **Lưu ý:**

- Cần chuẩn bị phân vùng đĩa còn trống để cài Linux.
- Linux cần tối thiểu hai phân vùng là *Linux Native* (ext3) và *Linux swap*. Đơn giản, bạn có thể dùng *Partition Magic* để phân chia đĩa.
- Một partition là *Linux native* ext3. Cần khoảng 2GB trở lên để cài Linux, bao gồm cả KDE và GNOME, các tiện ích đồ họa, multimedia, và lập trình. Tối thiểu bạn cần 400MB và cài toàn bộ là 4,5GB.
- Một partition là *Linux swap*, là phân vùng trao đổi của Linux dành cho việc sử dụng bộ nhớ ảo làm không gian trao đổi. Thông thường, dung lượng bộ nhớ ảo tối ưu sẽ gấp đôi dung lượng bộ nhớ RAM của hệ thống.

2.1.5. Chọn cấu hình phần cứng

Linux không đòi hỏi máy có cấu hình mạnh. Tuy nhiên nếu phần cứng có cấu hình thấp quá thì có thể không chạy được X-Window hay các ứng dụng có sẵn. Cấu hình tối thiểu nên dùng:

- CPU: Pentium MMX trở lên.
- RAM: 64MB trở lên cho Text mode, 192MB cho Graphics mode.
- Đĩa cứng: Dung lượng đĩa còn phụ thuộc vào loại cài đặt.
 - Custom Installation (minimum): 520MB
 - Server (minimum): 870MB
 - Personal Desktop: 1.9GB
 - Workstation: 2.4GB
 - Custom Installation (everything): 5.3GB
- 2MB cho card màn hình nếu muốn sử dụng chế độ đồ họa.

2.2. Tiến hành cài đặt

2.2.1. Các cách cài đặt

Có thể cài đặt Linux bằng một trong các cách sau:

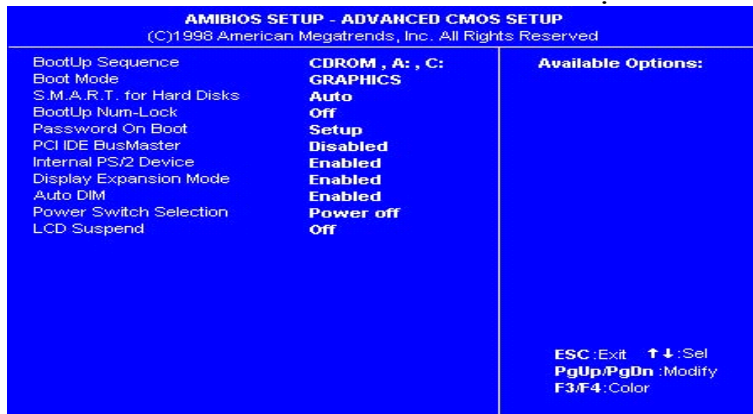
- Từ ổ đĩa CD-ROM.
- Từ bản sao chép Linux trên ổ đĩa cứng hoặc USB.
- Cài đặt từ server mạng, sử dụng HTTP, FTP, hoặc NFS, ...

2.2.2. Trình tự cài đặt

Quá trình cài đặt hệ điều hành Red Hat Linux 9.0

- Boot từ CD-ROM (*Hình 2.1*)

Nếu máy bạn có CD-ROM, bạn hãy khởi động máy tính, chỉnh lại BIOS thứ tự boot đầu tiên là CD-ROM và đưa đĩa cài đặt vào ổ CD.

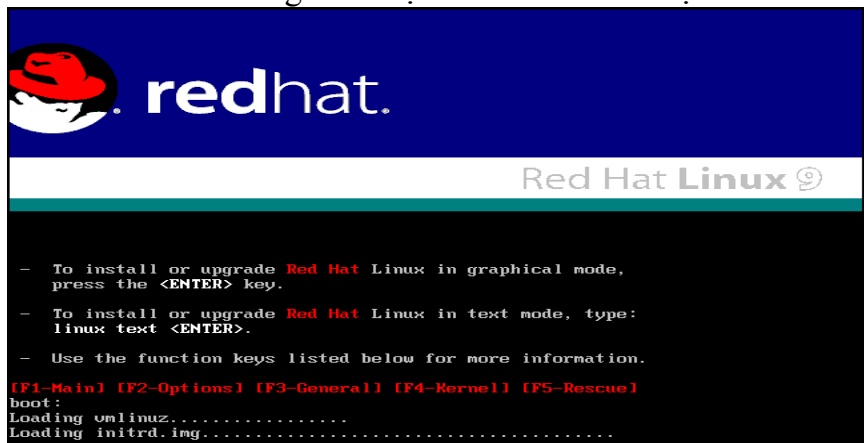


Hình 2.2.2.1 Boot CD

- Chọn chế độ cài đặt (*Hình 2.2*)

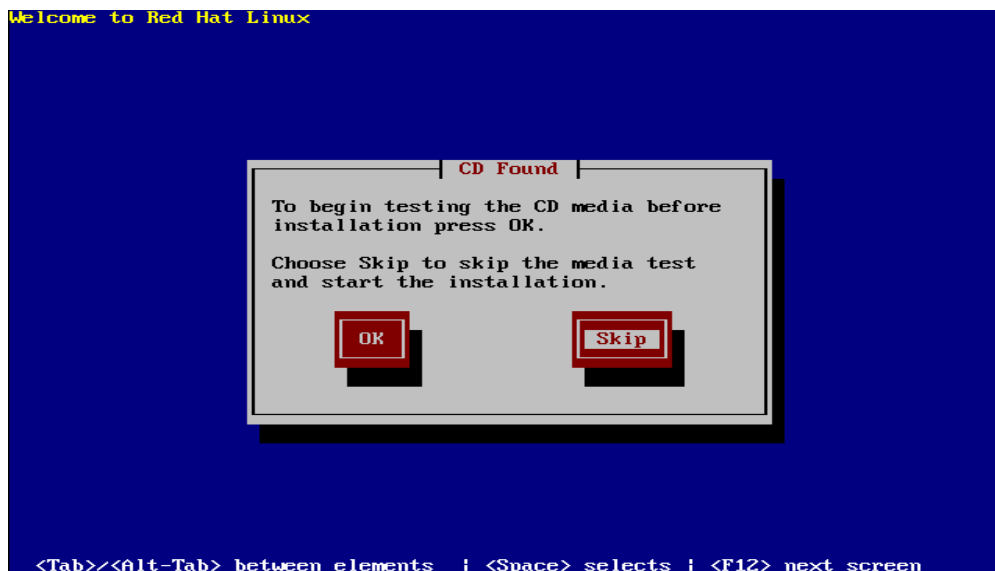
Chúng ta có thể chọn các chế độ:

- Chương trình hệ điều hành Linux đặt dưới chế độ đồ họa (*Graphical mode*) -> [Enter].
- Linux text: Chương trình hệ điều hành Linux đặt dưới chế độ text (*Text mode*).



Hình 2.2.2.2 Chọn chế độ cài đặt

- Kiểm tra đĩa trước khi cài đặt (*Hình 2.3*)

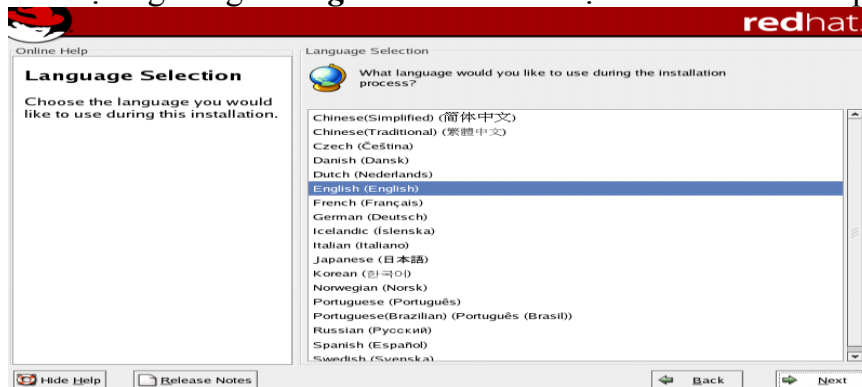


Hình 2.2.2.3 Thông báo kiểm tra đĩa

Chọn **OK** để test đĩa, hoặc chọn **Skip** để bỏ qua quá trình test.

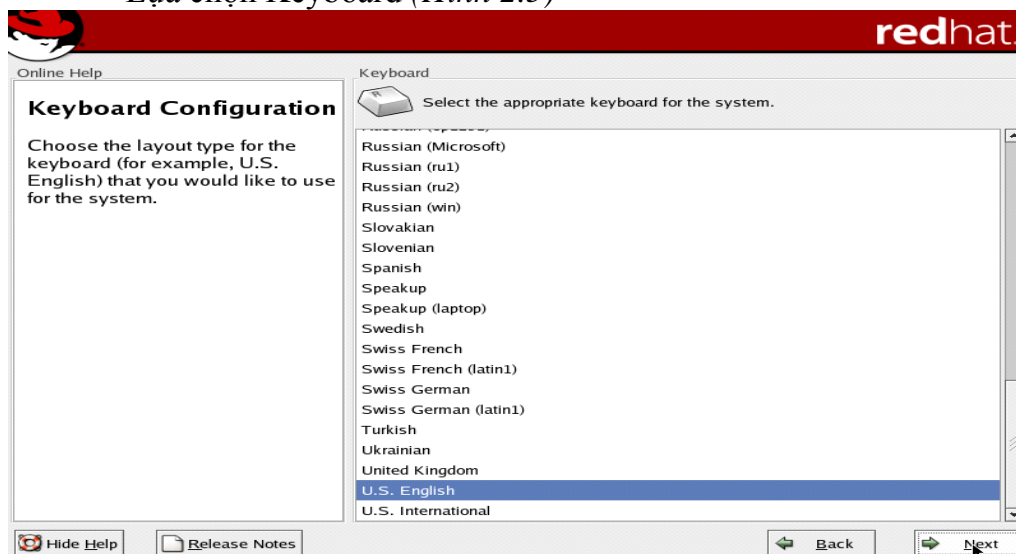
Chọn ngôn ngữ hiển thị trong quá trình cài đặt (Hình 2.4)

Ta chọn ngôn ngữ **“English”**. Sau khi chọn nhấn **Next** để tiếp tục.



Hình 2.2.2.4 Chọn ngôn ngữ

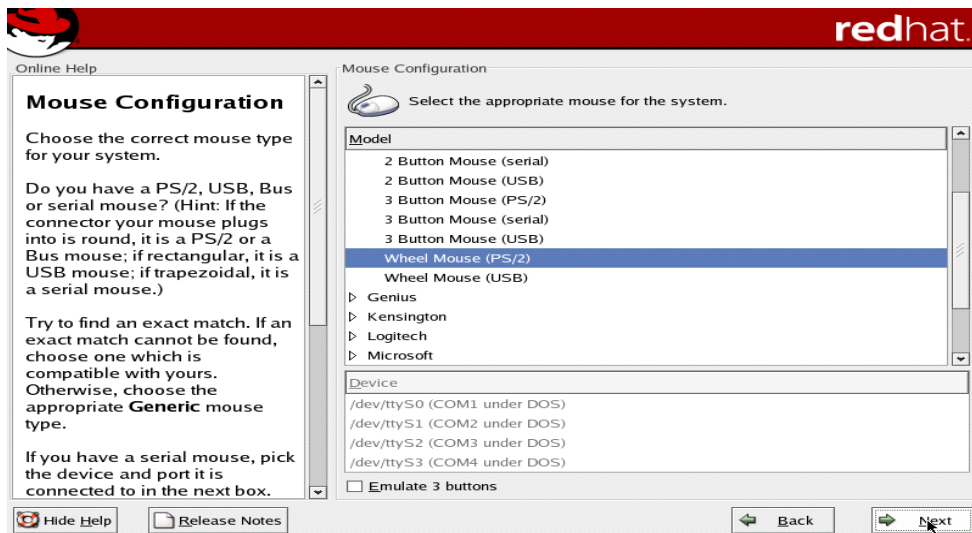
- Lựa chọn Keyboard (Hình 2.5)



Hình 2.2.2.5 Chọn phím

Chọn kiểu bàn phím thích hợp với hệ thống, chọn **Next**.

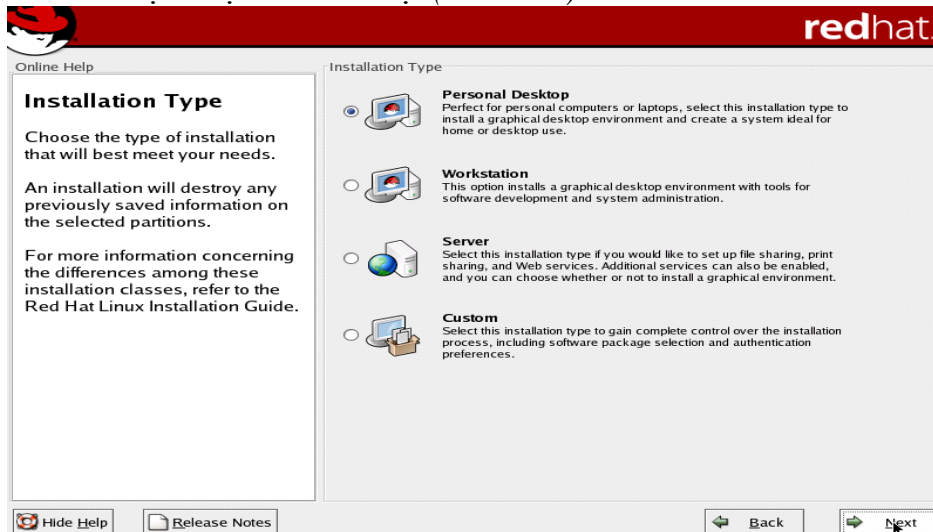
- Lựa chọn chuột (Hình 2.6)



Hình 2.2.2.6 Chọn chuột

Chọn loại mouse phù hợp với mouse của mình. Khi chọn lưu ý cổng gắn mouse là serial hay PS/2, chọn *Next*.

- Lựa chọn kiểu cài đặt (Hình 2.7)



Hình 2.2.2.7 Chọn kiểu cài đặt

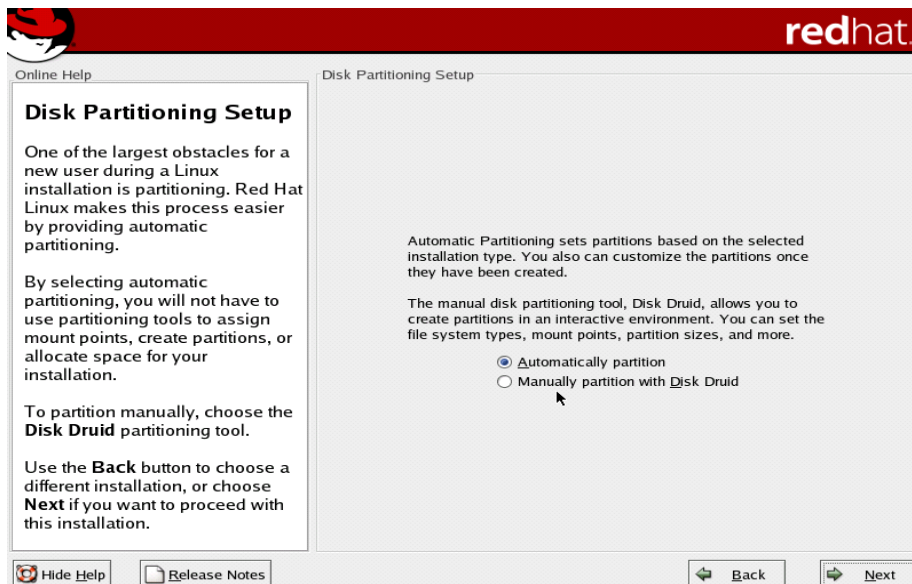
Personal Desktop: dành cho người mới bắt đầu với Linux hoặc cho những hệ thống desktop cá nhân. Chương trình cài đặt sẽ chọn lựa những gói phần mềm cần thiết nhất cho cấu hình này. Dung lượng đĩa cần cho kiểu cài đặt này chiếm khoảng 1.5GB, bao gồm cả môi trường đồ họa.

WorkStation: dành cho những trạm làm việc với chức năng đồ họa cao cấp và các công cụ phát triển.

Server: cài đặt hệ thống đóng vai trò máy chủ như *web server*, *ftp sever*, *SQL server*,...

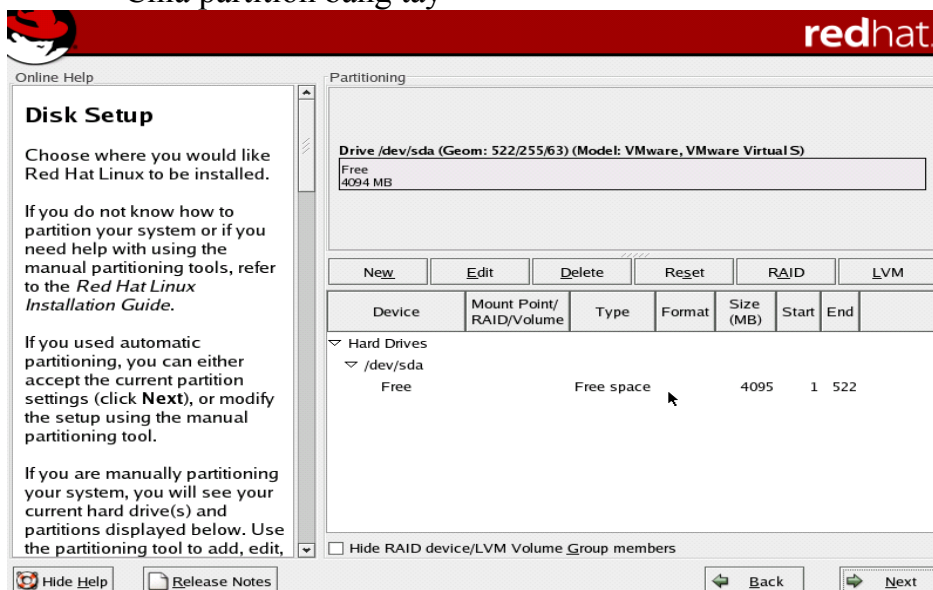
Custom: đây là lựa chọn linh hoạt cho bạn trong quá trình cài đặt. Bạn có thể chọn các gói phần mềm, các môi trường làm việc, *boot loader* tùy theo ý bạn.

- Chọn cách chia partition (Hình 2.8)



Hình 2.2.2.8 Lựa chọn chia partition

- **Automatically partition**: cho phép hệ thống tự động phân vùng ổ đĩa hợp lý để cài hệ điều hành (thông thường theo cách này thì hệ thống sẽ tạo ra hai phân vùng: */boot*, */swap*).
- **Manually partition with Disk Druid**: chia partition bằng tiện ích Disk Druid. Đây là cách chia partition dưới dạng đồ họa dễ dùng. Nếu ta là người mới học cách cài đặt thì nên lựa chọn **Automatically partition**.
- Chia partition bằng tay



Hình 2.2.2.9 Cách chia bằng tay

- **Remove all Linux partitions on this system**: khi ta muốn loại bỏ tất cả các Linux partition có sẵn trong hệ thống.
- **Keep all partitions and use existing free space**: khi ta muốn giữ lại tất cả các partition có sẵn và chỉ sử dụng không gian trống còn lại để phân vùng.
- Tùy theo từng yêu cầu riêng mà ta có thể lựa chọn các yêu cầu trên cho phù hợp, sau đó chọn **Next**.

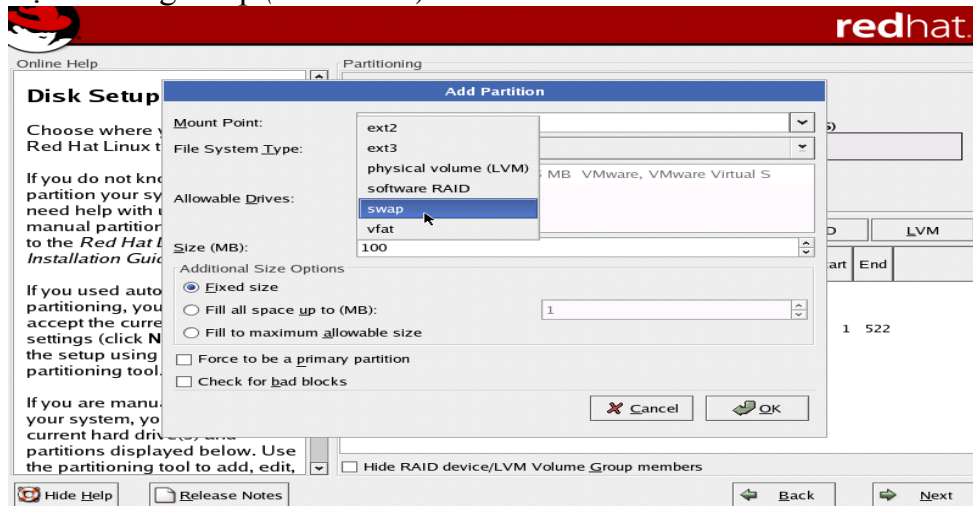
Khi ta chọn cách chia partition bằng tay ta sẽ sử dụng tiện ích **Disk Druid**.

Disk Druid hiển thị các partition của đĩa dưới chế độ đồ họa ở phía trên. Bạn có thể chọn từng partition để thao tác.

Chi tiết các partition gồm kích thước, loại hệ thống tập tin, thư mục được mount vào.

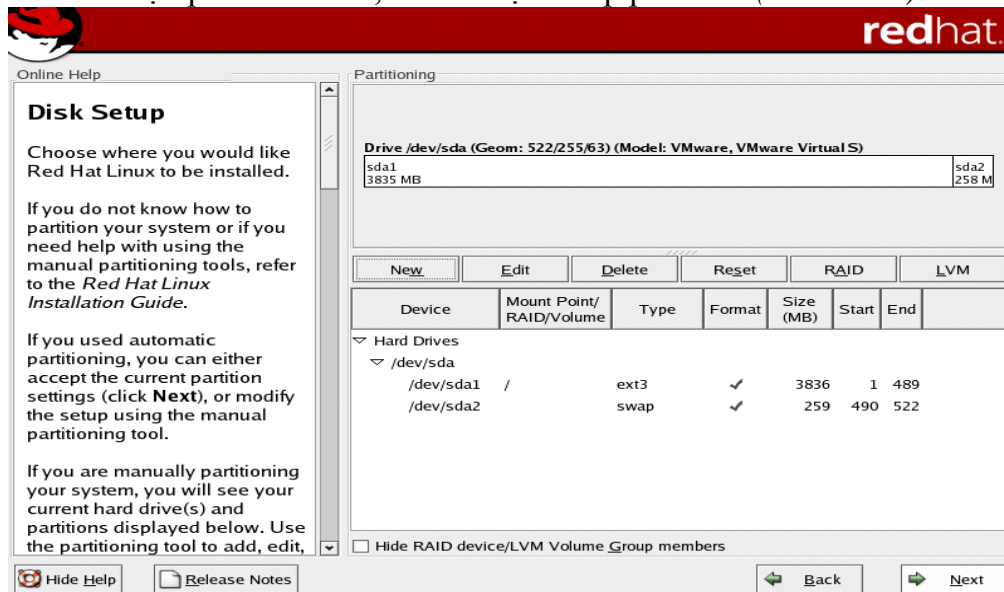
- **New**: tạo một partition mới, chỉ định tên phân vùng (*mount point*), loại filesystem (*ext3*) và kích thước (*size*) tính bằng MByte (tùy chọn).
- **Edit**: thay đổi lại các tham số của phân vùng được chọn.
- **Delete**: xóa phân vùng được chọn.
- **Reset**: phục hồi lại trạng thái đĩa như trước khi thao tác.
- **Make RAID**: sử dụng với RAID (*Redundant Array of Independent Disks*) khi ta có ít nhất 3 đĩa cứng.

Tạo ra vùng swap (Hình 2.10)



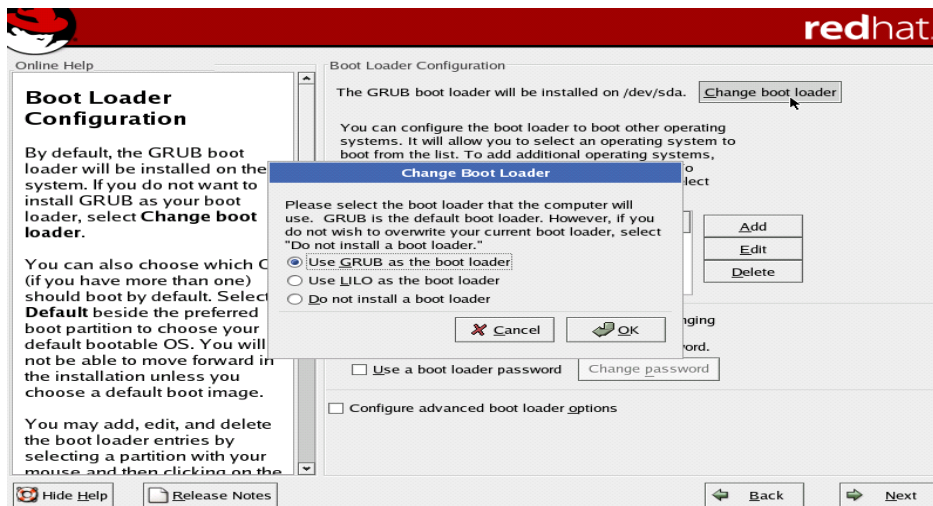
Hình 2.2.2.10 Tạo Swap

- Tạo partition ext3, sau khi tạo swap partition (Hình 2.11)



Hình 2.2.2.11 Tạo ext3

- Cấu hình Boot Loader



Hình 2.2.2.12 Cấu hình boot loader

Đây là chương trình dùng để khởi động Linux khi bạn có hơn một hệ điều hành được cài đặt trên hệ thống. Boot Loader cho phép bạn chọn các hệ điều hành để khởi động qua menu. Khi chúng ta chọn, chúng sẽ xác định các tập tin cần thiết để khởi động hệ điều hành và giao quyền điều khiển lại cho hệ điều hành. Boot Loader có thể được cài vào Master Boot Record hoặc vào sector đầu tiên của partition.

Linux cho phép bạn sử dụng chương trình Boot Loader là **GRUB** hoặc **LILO**. Cả 2 Boot Loader đều có thể hỗ trợ quản lý nhiều hệ điều hành trên một hệ thống.

- Bạn chọn cài Boot Loader vào *Master Boot Record (MBR)* khi chưa có chương trình Boot Loader nào (ví dụ như của Windows) được cài, hoặc bạn chắc chắn chương trình Boot Loader của bạn có thể khởi động được các hệ điều hành khác trong máy của mình. Khi cài lên MBR thì các chương trình Boot Loader trước đó sẽ bị thay thế bằng Boot Loader mới.

- Chọn cài Boot Loader vào sector đầu tiên của partition cài đặt khi bạn đã có chương trình Boot Loader tại MBR và không muốn thay thế nó. Trong trường hợp này, chương trình Boot Loader kia nắm quyền điều khiển trước và trở đến chương trình Boot Loader của Linux khi có yêu cầu khởi động hệ điều hành này.

- Bạn không cài chương trình Boot Loader, khi đó bạn phải sử dụng đĩa mềm boot để khởi động hệ điều hành.

- Ta có thể đặt mật khẩu cho Boot Loader thông qua nút **Change password**.

GRUB: Là boot loader mặc định, đây là chương trình rất mạnh và uyển chuyển.

GRUB tự động dò các hệ điều hành hiện có trên hệ thống và thêm vào danh sách khi khởi động.

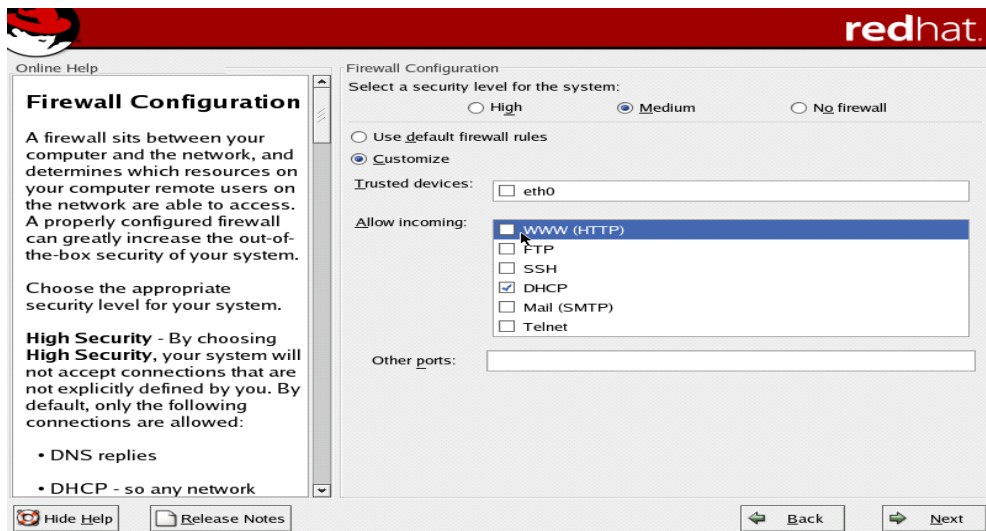
Tuỳ chọn "**configure advance boot loader option**" cho phép bạn chọn việc cài GRUB lên đâu trong ổ cứng.

Nếu chọn GRUB để khởi động hệ thống, GRUB sẽ được cài lên Master Boot Record (*/dev/hda*).

Nếu chọn một chương trình khác để khởi động như *system commander* chẳng hạn, bạn hãy chọn cài GRUB lên "**first sector of boot partition**". Như vậy, system commander sẽ tự động nhận ra Linux và thêm vào mục nhập khởi động cho Linux.

- Cấu hình Firewall

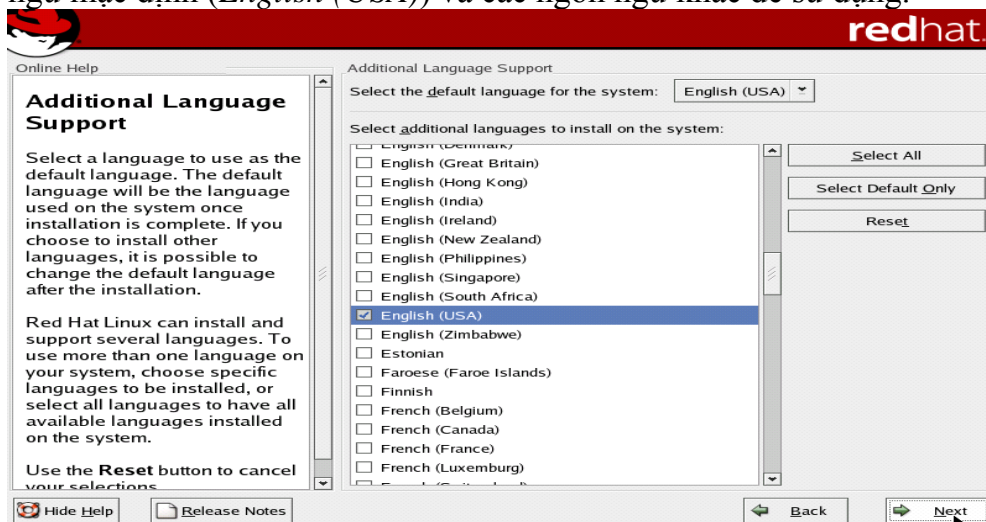
Trong Linux có tích hợp Firewall để bảo vệ hệ thống chống lại một số truy xuất bất hợp pháp từ bên ngoài. Ta chọn **Enable Firewall**, sau đó chọn loại dịch vụ cần cho phép bên ngoài truy cập vào Firewall.



Hình 2.2.2.13 Cấu hình Firewall

- Lựa chọn ngôn ngữ cho hệ thống (Hình 2.14)

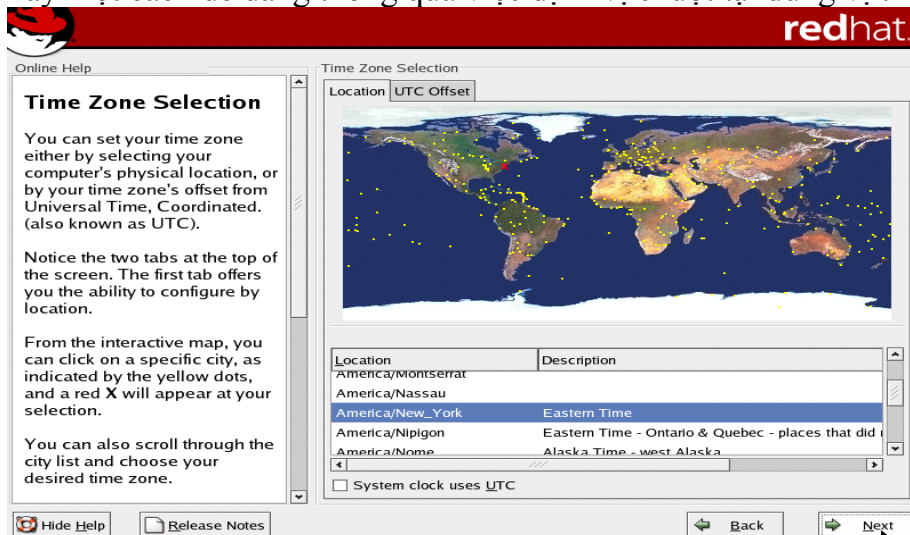
Bạn có thể cài đặt và sử dụng nhiều ngôn ngữ trong Linux. Có thể chọn ngôn ngữ mặc định (*English (USA)*) và các ngôn ngữ khác để sử dụng.



Hình 2.2.2.14 Chọn ngôn ngữ

- Lựa chọn vùng định thời gian (Hình 2.15)

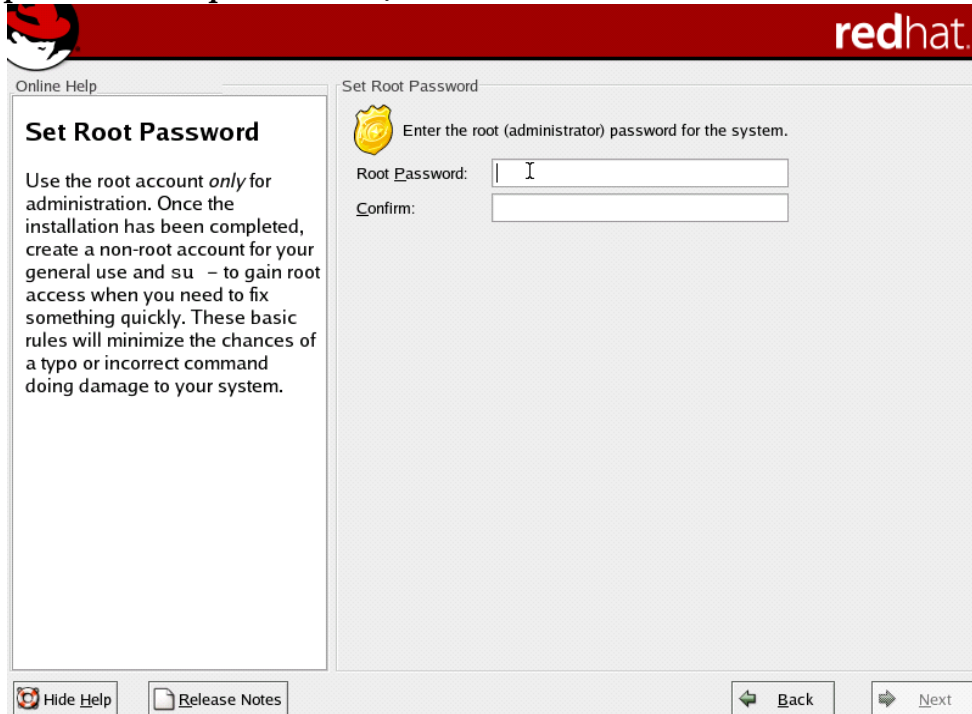
Các vị trí chia theo châu lục. Ở Việt Nam là *Asia/Saigon*, ta có thể chọn mục này một cách dễ dàng thông qua việc định vị chuột tại đúng vị trí trên bản đồ.



Hình 2.2.2.15 Định thời gian

- Thiết lập password cho user *root* (Hình 2.16)

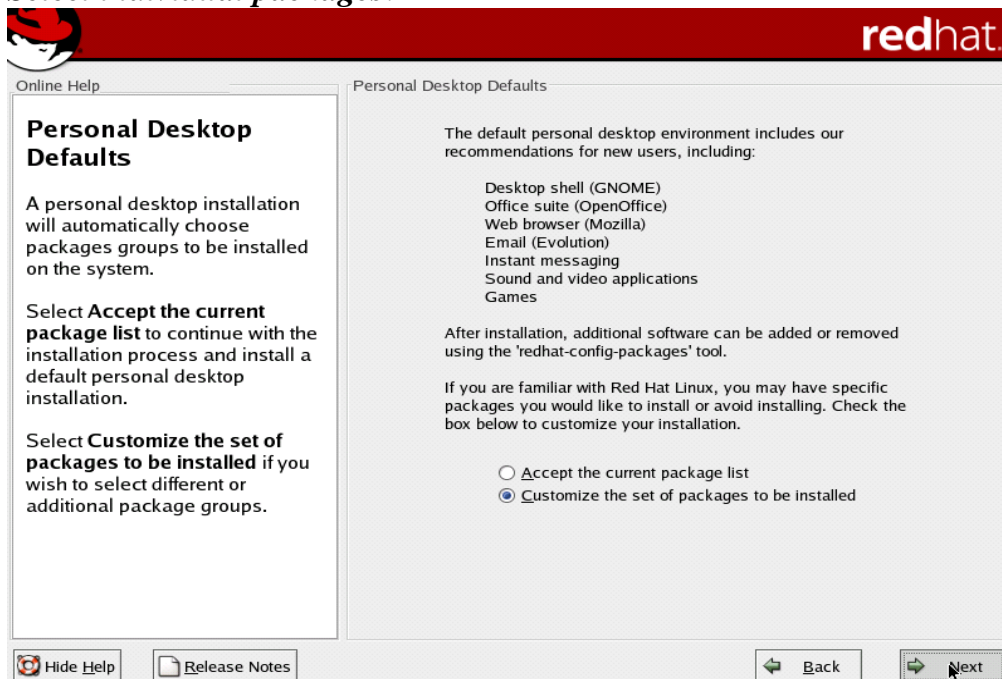
Trên Linux người quản trị thường được gọi là người dùng **root**. Mật khẩu của user *root* bắt buộc có chiều dài tối thiểu của password là 6 ký tự. Bạn nên đặt password gồm có ký tự, số và các ký tự đặc biệt để đảm bảo an toàn. Lưu ý password phân biệt chữ hoa và thường. Bạn phải đánh vào 2 lần, khi dòng chữ bên dưới xuất hiện “**Root password accepted**” thì được.



Hình 2.2.2.16 Thiết lập password

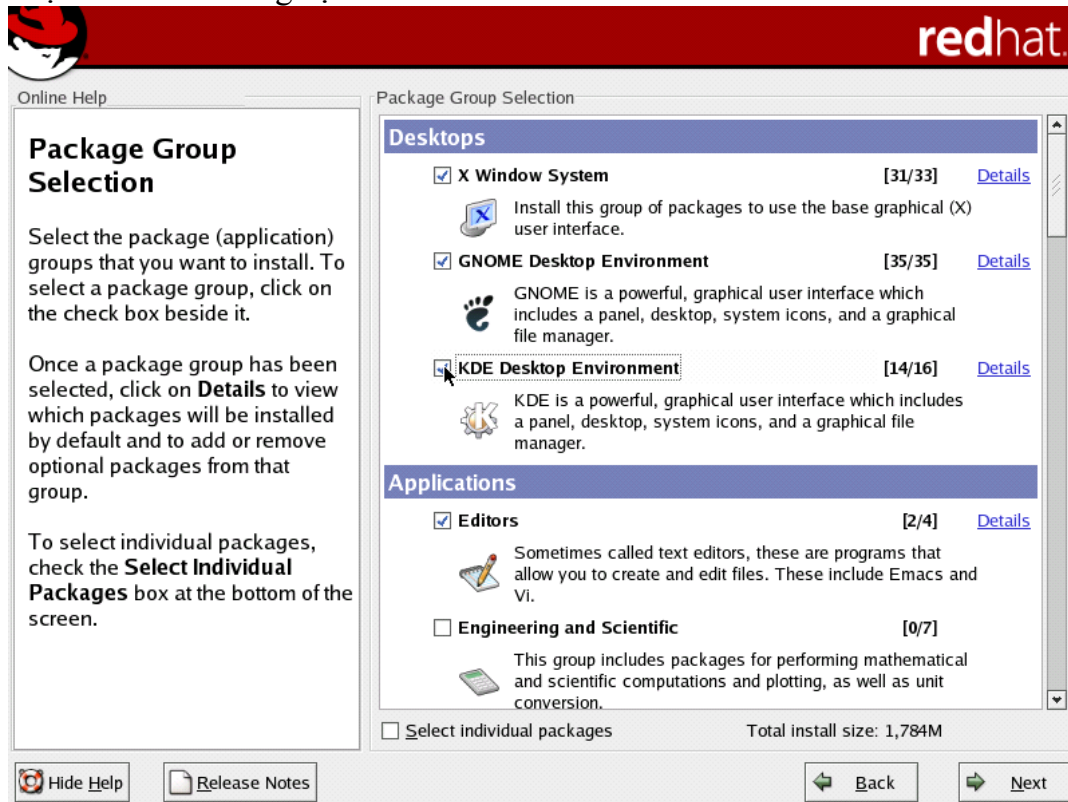
- Lựa chọn các gói cài đặt

Bạn chọn các chương trình cần cài đặt, nếu chọn **everything** là cài tất cả các chương trình, chọn **Minimal** là chỉ cài một số chương trình hoặc phần mềm thông dụng. Nếu bạn nắm rõ các package cần thiết cho các chương trình mình mong muốn thì chọn **Select individual packages**.



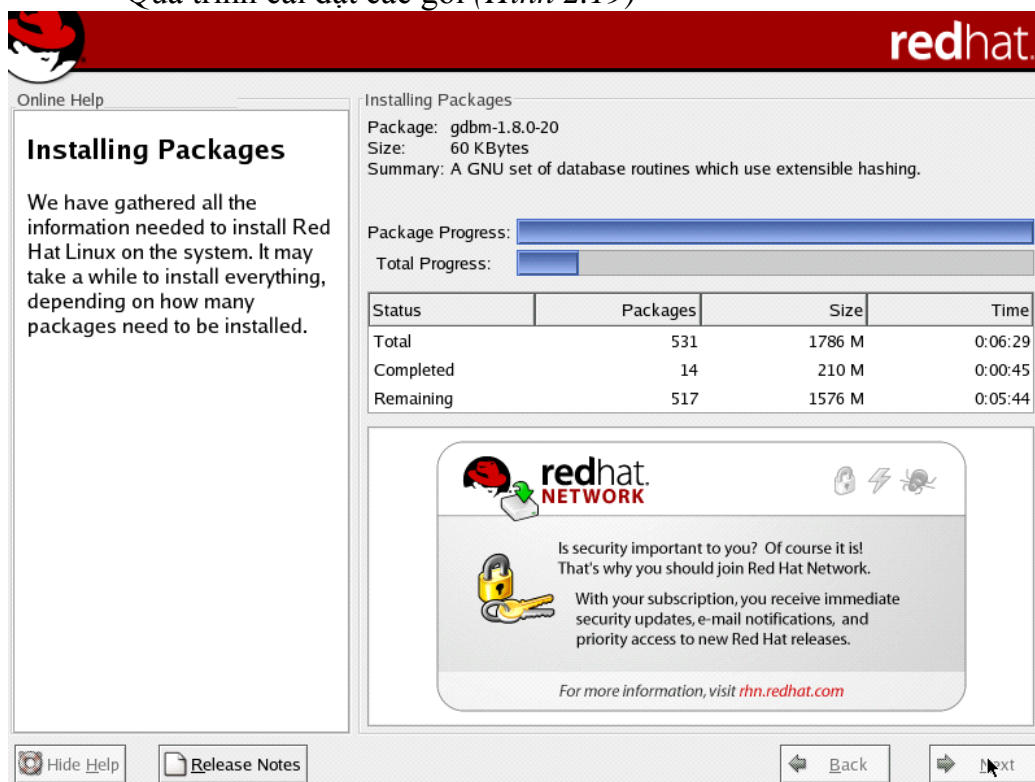
Hình 2.2.2.17 Chọn chương trình cài đặt

Ta có thể chọn **Details** để chọn chi tiết các thành phần trong từng phần mềm hoặc nhóm các công cụ.



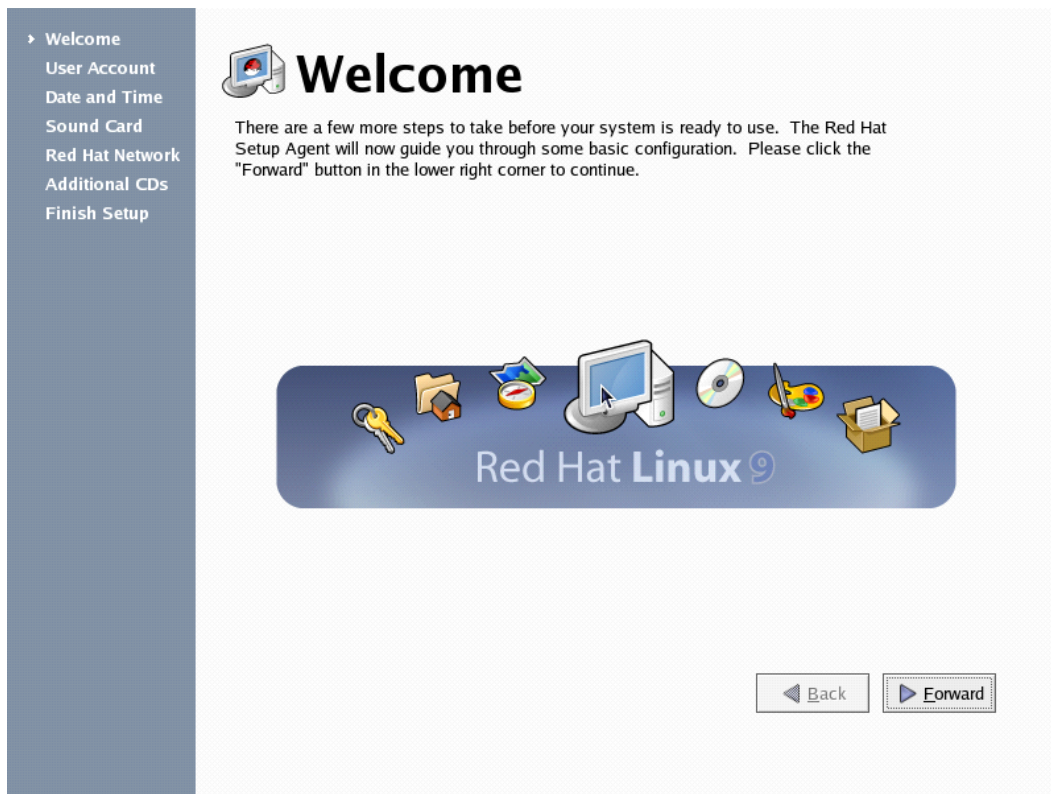
Hình 2.2.2.18 Chọn chi tiết thành phần

- Quá trình cài đặt các gói (Hình 2.19)

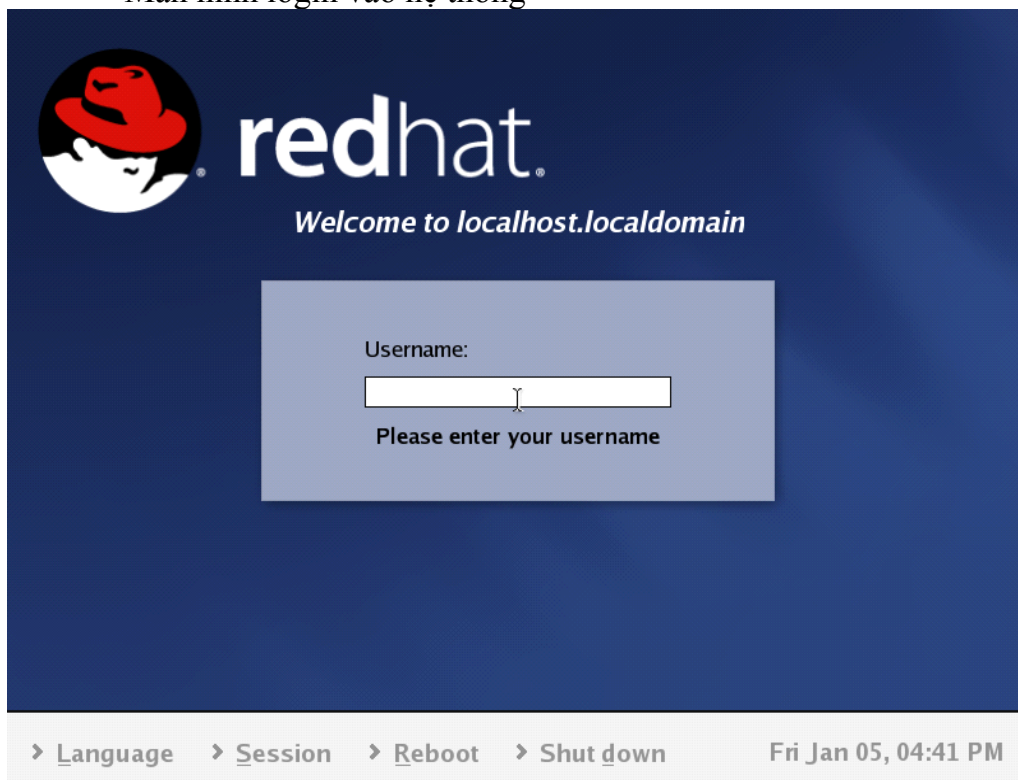


Hình 2.2.2.19 Cài đặt gói

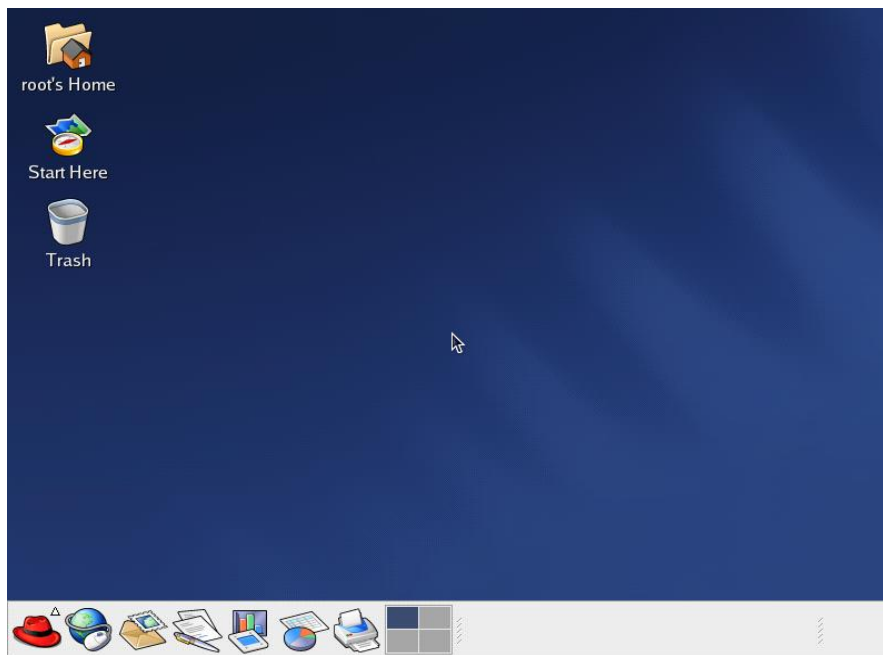
- Màn hình sau khi cài đặt xong và khởi động lần đầu tiên



Hình 2.2.2.20 Màn hình khởi động
- Màn hình login vào hệ thống



Hình 2.2.2.21 Màn hình login
- Màn hình của hệ điều hành Red hat linux 9.0



Hình 2.2.2.22 Màn hình Linux

2.2.3. Thiết lập cấu hình mạng

Trong Windows, để xem cấu hình mạng của máy ta dùng ipconfig. Còn trong Linux thì ta dùng ifconfig. Nhưng ifconfig còn mạnh hơn thế, nó có thể dùng để cấu hình IP cho máy bằng câu lệnh `[root@nhuhoang.net /]# ifconfig eth0 192.168.1.10 netmask 255.255.255.0`

Trong đó eth0, eth1 lần lượt là tên các card mạng của bạn, nếu máy chỉ có 1 NIC thì mặc định sẽ là eth0. Ngoài ra ta còn có thể gán nhiều IP cho Card mạng bằng cách dùng tham số : eth0:1 , eth0:2 , ... Khi đó từ 1 máy khác ta ping đến các IP này thì đều là ping đến 1 NIC duy nhất. Các bạn có thể dùng ifconfig để kiểm tra.

```
[root@nhuhoang.net /]# ifconfig eth0:1 192.168.1.31 netmask 255.255.255.0
```

```
[root@nhuhoang.net /]# ifconfig eth0:2 192.168.1.32 netmask 255.255.255.0
```

```
[root@nhuhoang.net /]# ifconfig
```

Chú ý: các thiết lập IP này chỉ có tác dụng đến khi restart. Muốn thiết lập vẫn có tác dụng cho lần khởi động sau thì bạn phải sửa các file sau:

- Đối với RHEL, Fedora: thêm vào cuối file `/etc/sysconfig/network/ifcfg-eth0` các dòng sau:

```
DEVICE=eth0
```

```
BOOTPROTO=none
```

```
HWADDR=00:48:54:1C:AD:86
```

```
ONBOOT=yes
```

```
TYPE=Ethernet
```

```
IPADDR=192.168.1.254
```

```
NETMASK=255.255.255.0
```

```
GATEWAY=192.168.1.1
```

(trong CentOS, file chịu trách nhiệm cấu hình mạng là `/etc/sysconfig/network-scripts/ifcfg-eth0` Bạn điền thông tin giống hệt như trong Fedora)

Để chỉ định DNS server, vi `/etc/resolv.conf` và thêm vào dòng: `nameserver 192.168.1.1;`

2.2.4. Thiết lập trương khoản người dùng

Việc cấu hình account dùng để thiết lập mật khẩu root và có thể tạo thêm các account khác để log in vào hệ thống khi việc cài đặt hoàn tất. Tài khoản root là tài khoản có

quyền cao nhất trong hệ thống. Bạn có thể cài đặt, cấu hình hệ thống hay làm mọi chuyện một khi đăng nhập vào hệ thống với tài khoản này.

2.2.5. Thiết lập cấu hình xác thực

* Cấu hình dịch vụ SSH trên Server

-Cấu hình lại file sshd_config

```
#vim /etc/ssh/sshd_config
```

Chú ý sshd với ssh nhé. Thêm hai dòng

```
PublicKeyAuthentication yes
```

```
AuthorizedKeysFile .ssh/authorized_keys
```

Hai dòng này đã bị [disable](#) bằng kí tự #

Chú ý:

```
PasswordAuthentication no # yes
```

Không cho phép (Cho phép) user login bằng [password](#), thường khi cấu hình xong, chúng ta sẽ không cho phép login bằng [password](#) thường để đảm bảo tối đa việc bảo mật.

-Tạo thư mục .ssh và file authorized_keys trong thư mục này. Thư mục .ssh được tạo trong home folder của user truy cập sử dụng ssh. Đối với các user thường thì là /home/username/, đối với user [root](#) thì là [/root/](#) (đường dẫn này có thể thay đổi trong cấu hình)

Ở đây thực hiện với root:

```
# cd /root
```

```
# mkdir .ssh
```

```
# cd /root/.ssh/
```

```
# vi authorized_keys
```

Copy nội dung của public key và paste vào đây rồi save lại (Làm như thế nào mình xin đề đến phần Client)

-Restart lại dịch vụ ssh

```
# /etc/init.d/sshd restart
```

* Cấu hình trên Client

- Đầu tiên tạo key pair để khởi tạo các key public lẫn private: `ssh-keygen -t rsa`
- Chấp nhận tất cả các mặc định, chỉ cần nhấn **Enter** khi được hỏi một

passphrase:

Generating public/private rsa key pair.

Enter file in which to save the key (/root/.ssh/id_rsa)

Enter passphrase (empty for no passphrase)

Enter same passphrase again

Your identification has been saved in /root/.ssh/id_rsa.

Your public key has been saved in /root/.ssh/id_rsa.pub.

- Bây giờ ta sẽ có một tập hợp các key trong [/root/.ssh/](#). Copy chúng vào máy muốn đăng nhập từ xa, trong trường hợp này là máy [192.168.1.100](#) :

```
ssh-copy-id -i ~/.ssh/id_rsa.pub root@192.168.1.100
```

- Ta sẽ được nhắc nhở về mật khẩu cho tài khoản trên máy chủ từ xa, nhập vào và **ssh-copy-id** sẽ copy key này vào thư mục chính xác và sắp xếp quyền truy cập hệ thống tập tin.

Tiến hành một thử nghiệm để chắc chắn rằng nó hoạt động: `ssh 192.168.1.100`

2.2.6. Chọn các gói phần mềm cài đặt

Việc chọn các gói phần mềm để cài đặt được thực hiện rất thuận tiện khi các gói phần mềm được gom lại thành nhóm. Có thể chọn cài các gói phần mềm ngay lúc

này các gói cần thiết hoặc có thể cài thêm sau khi hoàn tất cài đặt. Bạn chọn mục “select individual package” để cài thêm các gói mà mặc định sẽ không cài cho bạn. Ví dụ như mc (Midnight Commander, tương tự NC trong DOS). Sau khi lựa chọn xong, chương trình cài đặt sẽ duyệt các gói phụ thuộc để bạn cài thêm. Trong suốt quá trình chọn gói phần mềm cài đặt, bạn được thông báo dung lượng cần để cài đặt. Nên chú ý để không vượt quá dung lượng phân vùng mà bạn đã dành cho Linux trong quá trình chọn lựa. Một điều chú ý là bạn nên cài các programming develop và kernel source, các thư viện lập trình để thuận tiện cho việc sau này cần biên dịch lại nhân hệ điều hành hoặc cài đặt và biên dịch phần mềm và driver cho hệ thống.

2.2.7. Thiết lập cấu hình X Window

- Khởi động Xwindows từ text mode:

Bước 1: Bây giờ thử cho máy chạy ở runlevel 3 như sau: `#init 3`

Bước 2: Nếu đang khởi động trong text mode thì muốn vào Xwindows thì phải làm sao đây. Rất đơn giản ta sử dụng lệnh sau: `#startx`

- Lựa chọn chương trình quản lý của sổ:

Trong các hệ thống RedHat, FEDORA, CENTOS thì thường sử dụng hai chương trình quản lý của sổ là Gnome và KDE. Để có thể chuyển qua sử dụng chương trình nào thì ta có thể sử dụng lệnh `switchdesk` để chuyển qua lại giữa các chương trình.

Ví dụ 1: Muốn chuyển qua trình quản lý KDE thì ta thực hiện như sau: `#switchdesk KDE`

Red Hat Linux switchdesk 4.0 Copyright (C) 1999-2004 Red Hat, Inc. Redistributable under the terms of the GNU General Public License Desktop now set up to run KDE. For system defaults, remove /home/andrew/.Xclients

Ví dụ 2: Muốn chuyển qua trình quản lý GNOME thì ta thực hiện như sau:

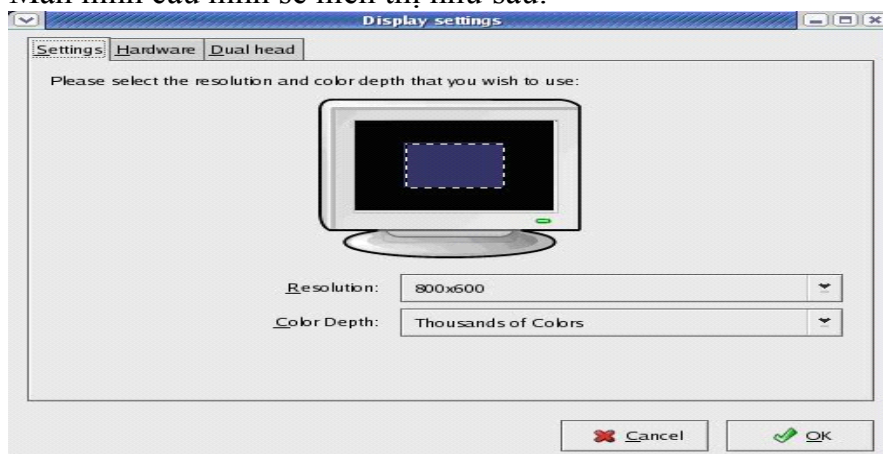
`#switchdesk GNOME` Red Hat Linux switchdesk 4.0

Copyright (C) 1999-2004 Red Hat, Inc. Redistributable under the terms of the GNU General Public License Desktop now set up to run GNOME.

- Thay đổi các thông số display của màn hình:

Bước 1: Từ dấu nhắc lệnh gõ lệnh sau để vào cửa sổ cấu hình `#system-config-display`

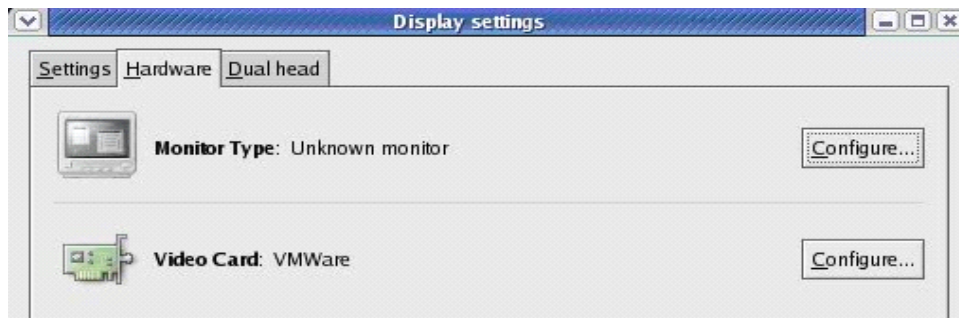
Màn hình cấu hình sẽ hiển thị như sau:



Hình 2.2.7.1 Màn hình Xwindow

Trong cửa sổ trên ta có thể cấu hình các thông số như: resolution, Color depth bằng cách chọn giá trị tương ứng.

Bước 2: Muốn cấu hình về loại monitor và card màn hình đang sử dụng ta chuyển qua phần hardware rồi chọn mục Configure để cấu hình như hình sau:



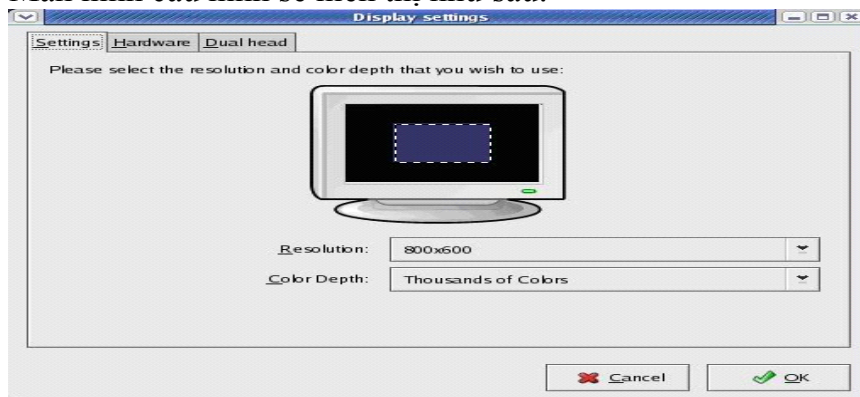
Hình 2.2.7.2 Màn hình thiết lập

- Thay đổi các thông số của chuột:

Từ dấu nhắc lệnh gõ lệnh sau để vào cửa sổ cấu hình

#system-config-mouse

Màn hình cấu hình sẽ hiển thị như sau:



Hình 2.2.7.3 Thay đổi chuột

Muốn cấu hình về loại mouse nào thì ta chọn tương ứng với loại mouse ta đang sử dụng.

2.2.8. Cài đặt các gói phần mềm

* Cài đặt các file.deb

File này rất dễ cài ta chỉ cần Click đúp vào file và trình cài đặt phần mềm trên hệ thống tự mở, click Install Package và chờ quá trình cài đặt hoàn tất.



Chú ý: Một số phần mềm yêu cầu máy phải cài sẵn một số Dependency, nếu không đủ các Dependency lúc cài đặt sẽ báo lỗi. Để giải quyết vấn đề Dependency với file.deb nhanh gọn có thể sử dụng gói Gdebi.

Đây là một ứng dụng có giao diện người dùng, nó cho phép cài đặt các gói deb đã download sẵn và nằm trên HDD, còn các Dependency phải kết nối Internet để Gdebi tự tìm và download. Gdebi cũng có thể chạy trong chế độ GUI bằng cách chuyển tới folder chứa file .deb và gõ `sudo gdebi package_name.deb` tại nhắc lệnh và vẫn có khả năng giải quyết các Dependency.

* Cài đặt các file.rpm

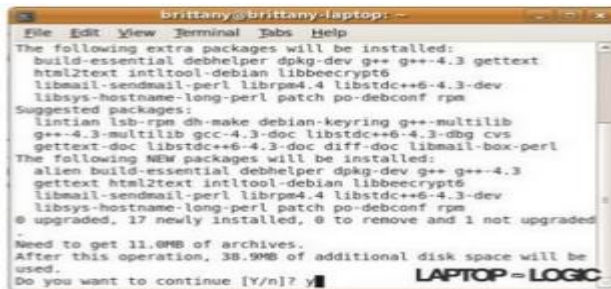
Ta nên sử dụng gói Alien để chuyển từ .rpm sang .deb cho dễ cài.

Mở Terminal, gõ vào `sudo apt - get install alien` để dowload và cài đặt gói Alien thông qua tiện ích quản lý gói APT.

Gõ vào Password ứng với User ta đang Logon, gõ y để đồng ý cài đặt gói Alien



```
brittany@brittany-laptop: ~$ sudo apt-get install alien
```



```
brittany@brittany-laptop: ~$ sudo apt-get install alien
The following extra packages will be installed:
  build-essential debhelper dpkg-dev g++ g++-4.3 gettext
  html2text intltool-debian libbeecrypt6
  libmail-sendmail-perl librpm4.4 libstdc++6-4.3-dev
  libsys-hostname-long-perl patch po-debconf rpm
Suggested packages:
  lintian lsb-rpm dh-make debian-keyring g++-multilib
  g++-4.3-multilib gcc-4.3-doc libstdc++6-4.3-dbg cvs
  gettext-doc libstdc++6-4.3-doc diff-doc libmail-box-perl
The following NEW packages will be installed:
  alien build-essential debhelper dpkg-dev g++ g++-4.3
  gettext html2text intltool-debian libbeecrypt6
  libmail-sendmail-perl librpm4.4 libstdc++6-4.3-dev
  libsys-hostname-long-perl patch po-debconf rpm
0 upgraded, 17 newly installed, 0 to remove and 1 not upgraded
Need to get 11.0MB of archives.
After this operation, 38.9MB of additional disk space will be
used.
Do you want to continue [Y/n]? y
```

Sau khi cài xong Alien ta move file .rpm tới Desktop rồi mở Terminal gõ `cd Desktop`



```
brittany@brittany-laptop: ~/Desktop$ cd Desktop
```

Bây giờ gõ `sudo alien -k filename.rpm` để convert từ file .rpm -> .deb. Sau đó cài file .deb như trên.



```
brittany@brittany-laptop: ~/Desktop$ sudo alien -k flash.rpm
File "flash.rpm" not found.
brittany@brittany-laptop: ~/Desktop$ sudo alien -k flash.rpm
Warning: Skipping conversion of scripts in package flash-plugin: postinst prepm
Warning: Use the --scripts parameter to include the scripts.
flash-plugin 10.0.15.3-release-1.1386.deb generated
brittany@brittany-laptop: ~/Desktop$
```

* Cài đặt file .bin

Download và save file .bin tới Desktop, sau đó mở Terminal và gõ `cd Desktop`

Gõ tiếp `sudo chmod +x filename .bin`

Gõ `./filename.bin` sau đó chương trình sẽ cài trong Terminal.

Câu hỏi ôn tập chương

1. Hãy nêu các bước chọn cấu hình cho hệ thống máy tính trước khi cài đặt hệ điều hành linux?
2. Hãy cho biết các bước cài đặt Redhat Linux?
3. Hãy nêu các bước thiết lập cấu hình cho hệ điều hành Linux?
4. Lựa chọn các gói phần mềm phù hợp để cài cho Linux?

Chương 3 Các khái niệm cơ bản trong Linux

Mục tiêu:

- Hiểu một số khái niệm cơ bản của hệ điều hành Linux;
- Biết các giao diện thao tác lệnh;
- Cài đặt được các cấu hình phần cứng;
- Quản lý được các tiến trình;
- Quản lý được người dùng trong hệ điều hành Linux;
- Có thái độ nghiêm túc, cẩn thận, chính xác.

3.1. Màn hình làm việc

3.1.1. Đăng nhập

Sau khi khởi động, muốn sử dụng Linux, phải đăng nhập vào tài khoản người dùng. Việc đăng nhập có thể ở **chế độ đồ họa** hoặc **console** (dòng lệnh). Ở chế độ đồ họa, một hộp hội thoại xuất hiện để bạn nhập username và password. Ở chế độ console (text), một dấu nhắc login chờ bạn nhập user name và password.. Nếu đã đăng nhập ở chế độ console (giao diện dòng lệnh), khi muốn vào chế độ đồ họa, ta dùng lệnh sau: **\$ startx**.

Một thói quen tốt là hãy đăng nhập vào tài khoản user bình thường để làm việc với Linux. **Tài khoản root chỉ được đăng nhập khi thật cần thiết** như cài đặt phần mềm, sửa chữa, phục hồi hệ thống hoặc cấu hình hệ thống.

3.1.2. Sử dụng shell prompt (Terminal)

Đây là một trình ứng dụng cho phép nhập lệnh với giao diện dòng lệnh thay vì tương tác với hệ thống qua giao diện đồ họa. Chạy ứng dụng này bằng cách: Chọn **Main menu -> system Tools -> Terminal** Thoát khỏi terminal bằng cách nhập lệnh **exit** hoặc nhấn **Ctrl_D** từ dấu nhắc.

3.1.3. Tạo account mới

Để tạo một account mới, ta sử dụng **User Manager**, có thể làm theo các cách sau: Nhấn chuột chọn icon Start Here (trên màn hình desktop) -> **System Settings -> User & Groups**

Chọn **Main Menu -> System Setting -> Users & Groups**

Gõ lệnh **redhat-config-users** tại dấu nhắc lệnh của shell

Với giao diện đồ họa và các icon chức năng đơn giản, dễ hiểu, ta có thể tự thao tác công việc quản lý user của mình.

Lưu ý là với người dùng **root**, ta có thể thay đổi password của bất kỳ user nào, điều này cần thiết khi ta quên password của user đó.

Khi quên mật khẩu root, ta phải khởi động máy ở chế độ single user để thay đổi password cho root.

Groups là thuộc tính nhóm của các user có chung những thuộc tính và quyền truy xuất hạn chế nào đó do người quản trị hệ thống cấp.

Mọi thay đổi về phân quyền, nhóm đều được thực hiện thông qua icon chức năng **Propertie**

Logout và Shutdown

Để logout, chọn **Main Menu ->Log Out**

Trước khi tắt máy, *nhất thiết phải shutdown chứ không được tắt nguồn điện trước*. Nếu không, hệ thống Linux có thể bị hỏng.

Nếu đang ở chế độ **console**, dùng lệnh **halt** để shutdown hệ thống. Lệnh **reboot** được dùng để khởi động lại hệ thống.

3.1.4. Nautilus- File Manager

Natilus là một file manager giống như *Window explore* trong MS Window. Đây là một môi trường tích hợp hầu hết các chức năng mà một hệ thống có thể làm được: duyệt hệ thống file, cấu hình desktop, cấu hình hệ thống, truy nhập tài nguyên mạng, v.v.

Để chạy Natilus như một file manager: *double click* vào biểu tượng **Home directory**.

Với Natilus, ta có thể tạo thư mục, copy hay xóa, sửa đổi file và thư mục một cách dễ dàng giống y như đối với Window explore. Tuy nhiên Natilus còn mạnh hơn với khả năng thao tác với nhiều kiểu tài liệu khác nhau được tích hợp trong Natilus.

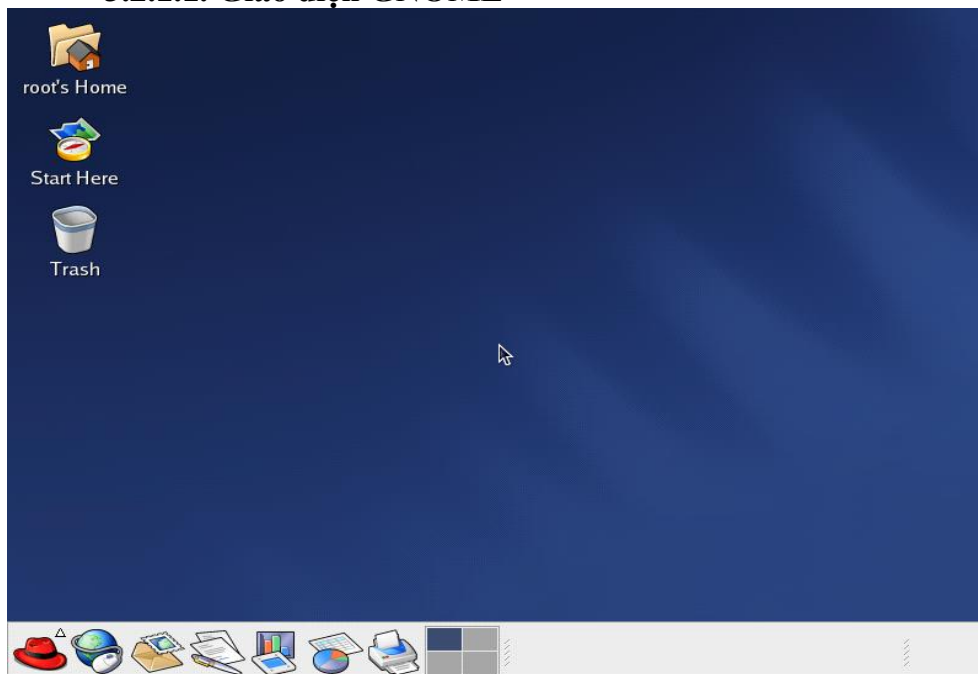
3.1.5. Start here

Start here được thiết kế để lưu trữ những công cụ và ứng dụng khi sử dụng hệ thống. Start here là một trung tâm cung cấp cho bạn các công cụ để sử dụng và cấu hình hệ thống của mỗi người.

3.2. Giao diện đồ họa và giao diện dòng lệnh

3.2.1. Giao diện đồ họa

3.2.1.1. Giao diện GNOME



Hình 3.2.1.1 Giao diện Gnome

GNOME là môi trường đồ họa mặc định của hệ điều hành Linux. GNOME có giao diện người dùng thân thiện và dễ sử dụng, mang đến khả năng xử lý mạnh mẽ và tiện lợi.

+ Màn hình nền (*desktop*)

Nền màn hình là không gian làm việc mà trong đó chúng ta chạy các ứng dụng và thực hiện các ứng dụng cho công việc của mình. Trên nền màn hình máy tính có chứa các biểu tượng cho phép liên kết tới các ứng dụng khác.

Biểu tượng chỉ đến thư mục riêng của người dùng. Mỗi người dùng có một thư mục riêng khác nhau. Thường các thư mục này để trong thư mục /home.



Trash

Biểu tượng thùng rác



root's Home



+ Thanh công cụ (Panel)



Trong môi trường GNOME thanh công cụ là phần có hình đánh nổi bật nhất. Từ thanh công cụ ta có thể chạy các ứng dụng và các ứng dụng nhỏ, các tiện ích và các ứng dụng nhỏ được đặt trên thanh công cụ. Trong số các ứng dụng đang chạy trên thanh công cụ là ứng dụng trang GNOME và đồng hồ chạy ở góc màn hình.

- GNOME Panel có thể xem như là trái tim của giao diện GNOME, có chức năng như một thanh chứa menu chính, menu người dùng, đường dẫn kích hoạt ứng dụng và nhiều đối tượng đặc biệt khác.

- Panel được thiết kế rất tiện dụng và linh hoạt. Người dùng có thể dễ dàng tùy biến nó như: thay đổi vị trí, thêm, bớt các đối tượng cho phù hợp với mục đích sử dụng.

*MainSystemMenu:

* Menu hệ thống chính trên Linux hoạt động tương tự như start menu ở MS-Windows. Nhấp trên biểu tượng chiếc nón màu đỏ sẽ hiển thị nhiều tùy chọn như sau:

- Accessories: chứa một số chương trình tiện ích như Calculator, Text Editor,...
- Games: Sub-menu là một danh sách các game để bạn thư giãn.
- Graphic: các chương trình xử lý đồ họa như: Photo tool, chương trình đọc file PDF: PDF viewer, PostScript viewer: xem các file postscript.
- Office: đây là bộ công cụ văn phòng, tương tự như MS-Office ở MS-Windows.
- Preferences: gồm nhiều công cụ đồ họa để cấu hình giao diện của desktop.
- Programming: cung cấp một số công cụ liên quan đến lập trình.
- Sound & Video: danh sách các chương trình xử lý multimedia như: CD Player, Sound recorder, Volume control,...

- System settings: tab này cung cấp các tùy chọn để cấu hình hệ thống: Add/Remove Application, Network, User and Group,...

- System tool: quản lý đĩa, phần cứng,...

- Nút Action: menu chứa một số chức năng thường dùng.

- Run Application: cho phép người dùng chọn chương trình và thực thi chúng.

- Search for file: một công cụ tìm kiếm file hiệu quả.

- Take screenshot: lưu lại màn hình thành một file ảnh.

- Lock screen: khoá lại màn hình khi người sử dụng muốn rời máy tính một lát.

- Log out: cho người dùng đăng xuất và thoát khỏi GNOME. Nếu người dùng đăng nhập

GNOME bằng màn hình đồ họa, thao tác này thoát khỏi GNOME và trở về màn hình đồ họa lúc người dùng đăng nhập. Còn nếu đăng nhập bằng dòng lệnh, sẽ trở lại giao diện dòng lệnh.

- Ngoài ra còn có một panel khi người dùng nhấp chuột phải lên màn hình desktop, nội dung panel này như sau:

- Open terminal: tạo một cửa sổ với dấu nhắc hệ thống, cho phép người sử dụng tương tác với hệ thống thông qua giao diện dòng lệnh.

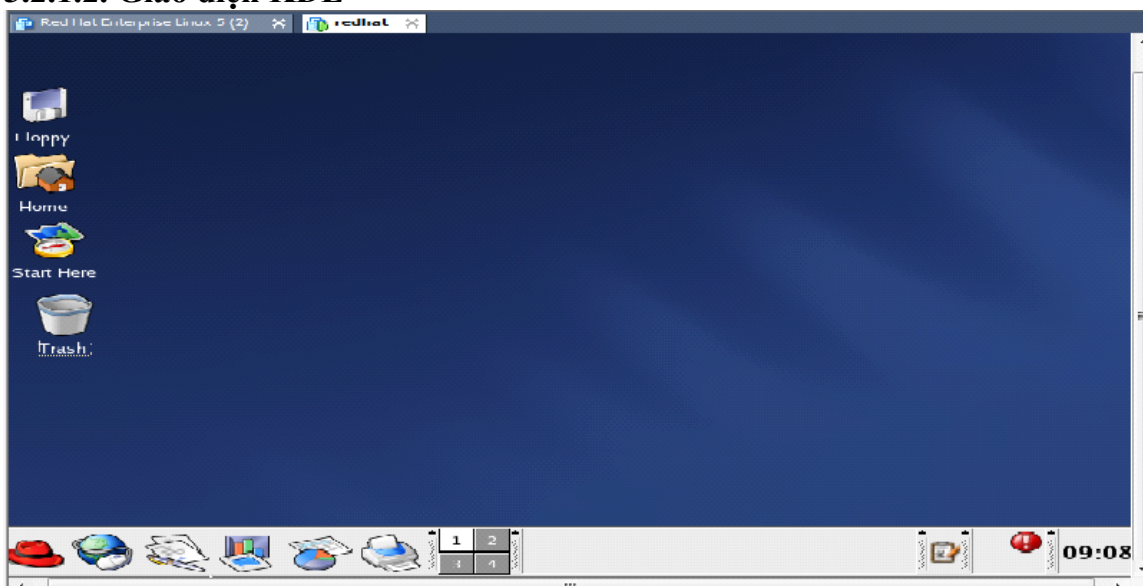
- Create folder: tạo mới một thư mục.

- Create launcher: tương tự như chức năng tạo shortcut của MS-Windows. Chức năng này cho phép tạo một biểu tượng lối tắt để nhanh chóng kích hoạt một ứng dụng.
 - Create document: tạo mới một file rỗng.
 - Clean up by name: sắp xếp lại item trên desktop theo tên.
- User default background: chọn hình nền cho desktop.

* Thoát khỏi giao diện GNOME:

- Có thể thoát khỏi giao diện GNOME bằng cách chọn Action “Log out”.
- Tại đây cho phép chọn nhiều tùy chọn như: Log out, Shut down, Restart the computer.

3.2.1.2. Giao diện KDE



Hình 3.2.1.2 Giao diện KDE

- KDE là môi trường giao diện đồ họa được sử dụng trên Linux tương tự như GNOME. Nếu đăng nhập thành công ta sẽ thấy màn hình chào của KDE.
- Sử dụng một các thành phần cơ bản trên KDE:

***KDEDesktop:**

- + Một màn hình KDE cơ bản bao gồm 3 vùng chính:
- + Panel ở đáy màn hình dùng để kích hoạt các ứng dụng, hoặc di chuyển qua lại giữa các chương trình. Bên trái cùng là 1 nút lớn mang biểu tượng chiếc nón, dùng để kích hoạt menu chứa các ứng dụng.
- + Thanh công cụ, mặc định được tích hợp chung với Panel, dùng để chuyển đổi qua lại hoặc quản lý các ứng dụng đang chạy. Nhấp vào một ứng dụng trên thanh taskbar này sẽ thu nhỏ ứng dụng lại.
- + Phần còn lại, cũng là phần hiển thị chính của màn hình nền, có chức năng hiển thị thông tin về file và thư mục. KDE hỗ trợ rất nhiều dạng màn hình nền, mỗi dạng có một cửa sổ hiển thị khác nhau. Nhấp lên những nút số trên Panel để di chuyển giữa các màn hình.

*** Thanh công cụ (Panel)**



Panel là một thanh dài nằm phía dưới màn hình. Mặc định Panel gồm biểu tượng menu chính, biểu tượng các chương trình ứng dụng, tiện ích, màn hình đang hoạt động và các chương trình đang chạy.

- Bạn có thể thêm hay xóa bớt các chương trình trên Panel bằng cách nhấp chuột phải trên Panel, chọn Panel/Configuration để mở cửa sổ Panel Setting.

- K Menu là menu chính của KDE. Khi nhấp chọn vào menu chính, bạn có thể thực hiện các tác vụ tương tự như nút Start trên Windows.

- Trên KDE bạn có thể làm việc với nhiều màn hình khác nhau, mặc định có thể chọn 4 màn hình, mỗi màn hình đều chứa các icon, các chương trình ứng dụng(application) và các hình nền (background) khác nhau.

Ta có thể thay đổi số lượng và trên các màn hình nền trên KDE, thực hiện theo cách sau:

- Nhấp chuột phải trên màn hình chọn Configure Desktop, công cụ cấu hình KDE sẽ xuất hiện.

- Nhấp chọn thẻ Multiple Desktops.

- Có thể thay đổi tên cho màn hình bằng cách xóa các tên mặc định và nhấp vào tên mới trong mỗi ô tương ứng.

- Bằng cách thay đổi vị trí con chạy tại góc trên màn hình, bạn có thể thay đổi số lượng màn hình muốn chạy đồng thời trên KDE.

- Chọn Apply để hoàn tất công việc.

* Trợ giúp:

- Bạn có thể tìm thấy toàn bộ tài liệu về cách sử dụng, cấu hình, làm việc với các chương trình ứng dụng... trên KDE, tại trung tâm trợ giúp KDE(KDE help Center).

* Có nhiều cách để mở phần trợ giúp:

- Từ menu chính: chọn Help.

- Nhấp chuột phải lên Panel và chọn help-> KDE Panel Handbook.

* Quản lý tập tin với Konquerer:

- Konquerer là trình quản lý tập tin của KDE, được tích hợp thêm chức năng của FTP – chương trình ứng dụng và kết nối mạng. Có thể kích hoạt Konquerer bằng cách chọn K Menu->Home.

- Để mở một file hoặc thư mục, đơn giản chỉ cần nhấp chuột trái lên nó một lần. Nhấp vào dấu hai chấm(..) trong nội dung thư mục sẽ trở về thư mục cha.

* Cấu hình cho KDE:

- Có thể dễ dàng thay đổi phong cách hay cách thức làm việc của giao diện KDE. Để làm điều này bạn có thể sử dụng chương trình KDE Control Center.

- Kích hoạt KDE Control Center: Chọn K Menu->Control Center.

* Appearance & Themes:

- Background: thay đổi tùy hình nền.

- Colors: thay đổi màu sắc hiển thị.

- Fonts: cấu hình font cho hệ thống.

* Desktop: cấu hình cho desktop.

- Multiple Desktops: cấu hình số lượng desktop ảo.

- Panels: chọn lựa vị trí các Panel.

- Taskbar: tùy chọn cho taskbar.

* Internet & Network: cấu hình để giao tiếp trên mạng cục bộ và internet.

- File sharing: cho/không cho phép chia sẻ file.

- Local network browsing: cấu hình lisa, reslisa và isoslaves.

- Preferences: cấu hình cho các tham số mạng thông dụng như: timeout,...

* KDE Compoments:

- Compoment Chooser: chọn các gói mặc định cho service.

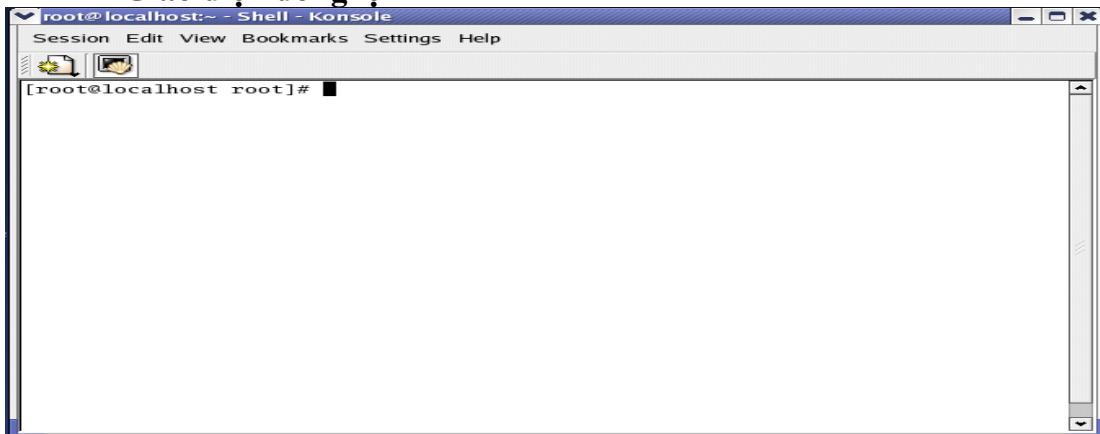
- File manager: cấu hình cho trình quản lý Konquerer.

* Peripherals: cấu hình cho các thiết bị ngoại vi.

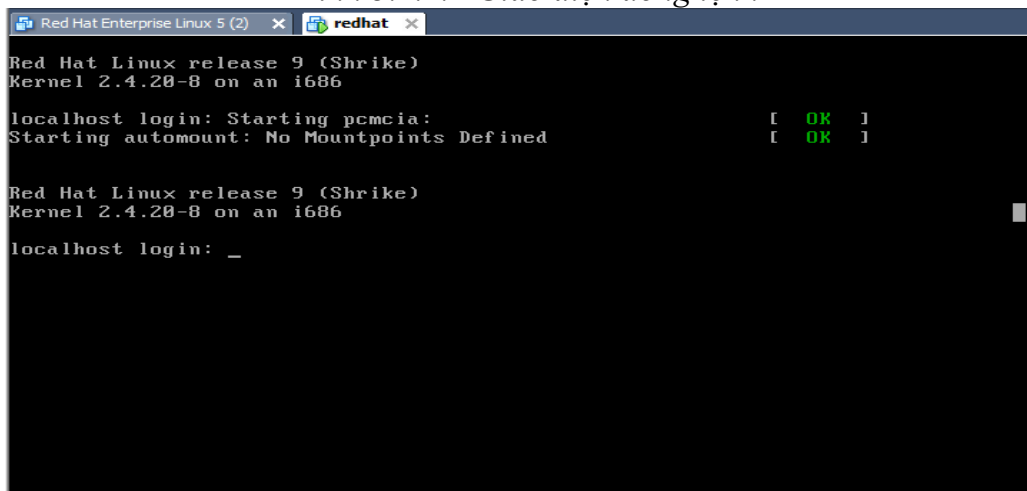
- Digital Camera.
 - Mouse.
 - Printers.
 - * Power Control: các tùy chọn về điện năng.
 - Laptop Battery: sử dụng cho pin trên laptop.
 - * Regional & Accessibility:
 - Country/Region & Language: tùy chọn về ngôn ngữ, tiền tệ, thời gian cho vùng địa lý bạn đang sống.
 - Keyboard Shortcuts: tạo các phím tắt.
 - * Security & Privacy:
 - Password & User Account: thông tin người dùng như: tên, mật mã, email,...
 - * Sound & Multimedia:
 - Audio CDs: cấu hình cho chương trình audio.
 - Sound system: điều chỉnh hệ thống âm thanh.
 - * System Administrations:
 - Date & Time: thay đổi ngày và giờ hệ thống.
 - Font Installer: cài đặt và quản lý font chữ của hệ thống.
 - * Thoát khỏi giao diện KDE:
 - Từ K Menu: chọn Logout.
 - Từ màn hình: nhấp nút phải chuột chọn Logout.
- Có 2 cách để thoát khỏi KDE:

3.2.2. Giao diện dòng lệnh

- Giao diện dòng lệnh



Hình 3.2.2.1 Giao diện dòng lệnh



Hình 3.2.2.2 Giao diện dòng lệnh

- Một số câu lệnh cơ bản

- + exit: thoát khỏi cửa sổ dòng lệnh.
- + logout: tương tự exit.
- + reboot: khởi động lại hệ thống.
- + halt: tắt máy.
- + startx: khởi động chế độ xwindows từ cửa sổ terminal.
- + mount: gắn hệ thống tập tin từ một thiết bị lưu trữ vào cây thư mục chính.
- ví dụ: mount /mnt/floppy: lắp thêm ổ mềm. ; mount /mnt/cdrom: lắp ổ CD-ROM
- + unmount: ngược với lệnh mount.
- + clear: xoá trắng cửa sổ dòng lệnh.
- + date: xem ngày, giờ hệ thống.
- + cal: xem lịch hệ thống.
- + ifconfig : để xem thông tin chi tiết về các giao diện mạng; thông thường giao diện mạng ethernet có tên là eth(). Có thể cài đặt các thiết lập mạng như địa chỉ IP hoặc bằng cách dùng lệnh này (xem man ifconfig). Nếu có điều gì đó chưa chính xác, ta có thể stop hoặc start (tức ngừng hoặc khởi động) giao diện bằng cách dùng lệnh ifconfig up/down.
- + passwd: cho phép thay đổi mật khẩu (passwd người_dùng_sở_hữu_mật_khẩu hoặc tên người dùng khác nếu ta đăng nhập hệ thống với vai trò root).

3.3. Hệ thống tập tin

3.3.1. Các kiểu file có trong Linux

Có rất nhiều file khác nhau trong Linux, nhưng bao giờ cũng tồn tại một số kiểu file cần thiết cho hệ điều hành và người dùng, dưới đây giới thiệu lại một số các kiểu file cơ bản.

-File người dùng (user data file): là các file tạo ra do hoạt động của người dùng khi kích hoạt các chương trình ứng dụng tương ứng. Ví dụ như các file thuần văn bản, các file cơ sở dữ liệu hay các file bảng tính.

-File hệ thống (system data file): là các file lưu trữ thông tin của hệ thống như: cấu hình cho khởi động, tài khoản của người dùng, thông tin thiết bị ... thường được cất trong các tệp dạng văn bản để người dùng có thể can thiệp, sửa đổi theo ý mình.

-File thực hiện hay thực thi (executable file): là các file chứa mã lệnh hay chỉ thị cho máy tính thực hiện. File thực hiện lưu trữ dưới dạng mã máy mà ta khó có thể tìm hiểu được ý nghĩa của nó, nhưng tồn tại một số công cụ để "hiểu" được các file đó. Khi dùng trình ứng dụng mc, file thực hiện được bắt đầu bởi dấu (*) và thường có màu xanh lục.

- Thư mục hay còn gọi là file bao hàm (directory): là file bao hàm các file khác và có cấu tạo hoàn toàn tương tự như file thông thường khác nên có thể gọi là file. Trong mc, file bao hàm thường có màu trắng và bắt đầu bằng dấu ngã (~) hoặc dấu chia (/). Ví dụ: /, /home, /bin, /usr, /usr/man, /dev ...

-File thiết bị (device file): là file mô tả thiết bị, dùng như là định danh để chỉ ra thiết bị cần thao tác. Theo quy ước, file thiết bị được lưu trữ trong thư mục /dev. Các file thiết bị hay gặp trong thư mục này là tty (teletype - thiết bị truyền thông), ttyS (teletype serial - thiết bị truyền thông nối tiếp), fd0, fd1, ... (floppy disk- thiết bị ổ đĩa mềm), hda1, hda2, ... hdb1, hdb2, ... (hardisk thiết bị ổ cứng theo chuẩn IDE; a, b,... đánh số ổ đĩa vật lý; 1, 2, 3... đánh số ổ logic). Trong mc, file thiết bị có màu tím và bắt đầu bằng dấu cộng (+).

-File liên kết (linked file): là những file chứa tham chiếu đến các file khác trong hệ thống tệp tin của Linux. Tham chiếu này cho phép người dùng tìm nhanh tới file thay vì tới vị trí nguyên thủy của nó. Hơn nữa, người ta có thể gắn vào đó các thông tin phụ trợ làm cho file này có tính năng trội hơn so với tính năng nguyên thủy của nó. Ta thấy loại file này giống như khái niệm shortcut trong MS-Windows98. Không giống một số hệ điều hành khác (như MS-DOS chẳng hạn), Linux quản lý thời gian của tệp tin qua các thông số thời gian truy nhập (accessed time), thời gian kiến tạo (created time) và thời gian sửa đổi (modified time).

3.3.2. Quy ước tên file trong Linux

Một đối tượng điển hình trong các hệ điều hành đó là file. File là một tập hợp dữ liệu có tổ chức được hệ điều hành quản lý theo yêu cầu của người dùng. Cách tổ chức dữ liệu trong file thuộc về chủ của nó là người đã tạo ra file. File có thể là một văn bản (trường hợp đặc biệt là chương trình nguồn trên C, PASCAL, shell script ...), một chương trình ngôn ngữ máy, một tập hợp dữ liệu.

Hệ điều hành quản lý file theo tên gọi của file (tên file) và một số thuộc tính liên quan đến file. Trước khi giới thiệu một số nội dung liên quan đến tên file và tên thư mục, chúng ta giới thiệu sơ bộ về khái niệm thư mục.

Để làm việc được với các file, hệ điều hành không chỉ quản lý nội dung file mà còn phải quản lý các thông tin liên quan đến các file. Thư mục (directory) là đối tượng được dùng để chứa thông tin về các file, hay nói theo một cách khác, thư mục chứa các file. Các thư mục cũng được hệ điều hành quản lý vì vậy, thư mục cũng được coi là file song trong một số trường hợp để phân biệt với "file" thư mục, chúng ta dùng thuật ngữ file thông thường. Khác với file thông thường, hệ điều hành lại quan tâm đến nội dung của thư mục.

Tên file trong Linux có thể dài tới 256 ký tự, bao gồm các chữ cái, chữ số, dấu gạch nối, gạch chân, dấu chấm. Tên thư mục/file trong Linux có thể có nhiều hơn một dấu chấm, ví dụ: This_is.a.VERY_long.filename. Nếu trong tên file có dấu chấm "." thì xâu con của tên file từ dấu chấm cuối cùng được gọi là phần mở rộng của tên file (hoặc file). Ví dụ, tên file trên đây có phần mở rộng là *.filename*.

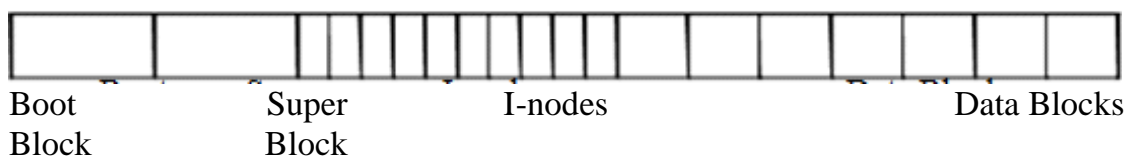
Chúng ta nên lưu ý rằng, không phải ký tự nào cũng có nghĩa. Nếu có hai file chỉ khác nhau ở ký tự cuối cùng, thì đối với Linux, đó là hai file có thể trùng tên. Bởi lẽ, Linux chỉ lấy 32 hay 64 ký tự đầu tiên trong tên file mà thôi (tùy theo phiên bản Linux), phần tên file còn lại dành cho chủ của file, Linux theo dõi thông tin, nhưng thường không xem các ký tự đứng sau ký tự thứ 33 hay 65 là quan trọng đối với nó.

Xin nhắc lại lưu ý về phân biệt chữ hoa và chữ thường đối với tên thư mục/file, ví dụ hai file FILENAME.tar.gz và filename.tar.gz là hai file khác nhau. Nếu trong tên thư mục/file có chứa khoảng trống, sẽ phải đặt tên thư mục/file vào trong cặp dấu nháy kép để sử dụng thư mục/file đó. Ví dụ, để tạo thư mục có tên là "My document" chẳng hạn, hãy đánh dòng lệnh sau: # mkdir "My document"
Một số ký tự sau không được sử dụng trong tên thư mục/file: !, *, \$, &, # ... Khi sử dụng chương trình mc, việc hiển thị tên file sẽ bổ sung một ký tự theo nghĩa: dấu "*" cho file khả thi trong Linux, dấu "~" cho file sao lưu, dấu "." cho file ẩn, dấu "@" cho file liên kết...

Tập hợp tất cả các file có trong hệ điều hành được gọi là hệ thống file là một hệ thống thống nhất. Bởi chính từ cách thức sử dụng thư mục, hệ thống file được tổ chức logic theo dạng hình cây: Hệ thống file được xuất phát từ một thư mục gốc (được kí hiệu là "/") và cho phép tạo ra thư mục con trong một thư mục bất kỳ. Thông thường, khi khởi tạo Linux đã có ngay hệ thống file của nó.

3.3.3. Cấu trúc hệ thống file của Linux

Hệ thống file của linux gồm bốn thành phần chính là Boot block (dùng để khởi động hệ thống), Siêu khối (Super block), Danh sách inode và Vùng dữ liệu.



Block 0

Thường không được sử dụng và thường chứa mã để nạp HĐH (boot the computer). Nó chứa một đoạn chương trình sẽ được đọc vào máy khi khởi động hệ điều hành Mặc dù Boot block chỉ cần thiết khi khởi động máy nhưng tương tự với Boot record của DOS, tất cả các hệ thống file UNIX đều có một Boot block (block này có thể để trống).

Block 1:

Là Super Block (siêu khối), trình bày trạng thái của hệ thống File (số lượng I-node, số Disk Block, điểm bắt đầu của danh sách của khối đĩa trống (free disk blocks)). Là một dạng bản ghi mô tả tình trạng của hệ thống file. Nó gồm các thông tin sau:

- Kích thước hệ thống file.
- Số khối còn trống trong hệ thống file.
- Danh sách khối trống trong hệ thống file.
- Chỉ số của khối tiếp theo trong danh sách khối trống.
- Kích thước của danh sách inode.
- Số inode còn trống trong hệ thống file.
- Danh sách inode còn trống trong hệ thống file.
- Chỉ số inode tiếp theo trong danh sách inode trống trong hệ thống file.
- Trường khoá của danh sách khối và inode trống.
- Cờ báo hiệu super block đã bị thay đổi.

I-nodes

Tương ứng bảng FAT trong MS-DOS, trình bày bên trong của một File được cho bởi một I-node, chứa đựng các thông tin mô tả về lưu trữ file trên đĩa và một số thông tin khác như: người chủ sở hữu, quyền truy nhập, thời gian truy nhập file. Mỗi I-node dài 64 byte và miêu tả chính xác một file. Inode là một bảng chứa các thông tin chi tiết về một file. Mỗi file đều được gắn với một inode qua số hiệu inode. Khi file được sử dụng bởi một tiến trình nào đó thì inode sẽ được đọc vào bộ nhớ và quản lý bởi kernel. Mỗi inode bao gồm các thông tin sau:

- Quyền sở hữu file: Quyền sở hữu được chia làm hai phần là người sở hữu file và nhóm người sở hữu. Người sở hữu thường là người tạo ra file đó. Nhóm người sở hữu file, trong UNIX System V thì thường thuộc về nhóm của người tạo ra file đó, còn trong BSD UNIX thì file thuộc về nhóm sở hữu thư mục mà file được tạo ra. Quyền sở hữu của người sử dụng và của nhóm đối với mỗi file có thể thay đổi được (ví dụ lệnh `chown`, `chgrp` của shell). Quyền sở hữu này cùng với quyền truy nhập của file sẽ quyết định xem ai có thể truy nhập tới tập tin và có thể truy nhập như thế nào.

- Loại file: Khái niệm file của UNIX có khác so với file trong DOS, ta có thể kể tới một số loại file sau.

Kiểu file thường: Đó là các file văn bản, các file nhị phân, file dữ liệu hay là các file chương trình...

Thư mục con: Là những file tạo ra cấu trúc phân cấp cho hệ thống file gồm danh sách các file trong nó và có thể chứa cả các thư mục khác. Nó có một vai trò quan trọng trong việc biến đổi tên file thành số hiệu inode. Thư mục là một file mà toàn bộ dữ liệu là chuỗi các phần tử (entry), mỗi phần tử chứa một số hiệu inode và tên file tương ứng trong thư mục. Đối với hệ UNIX System V chỉ cho phép tên file tối đa dài 14 ký tự còn đối với các hệ khác chiều dài này có thể lớn hơn. Do thư mục là các file đặc biệt nên tuy các file có thể đọc dữ liệu trong thư mục như đối với các file thường nhưng kernel giành quyền ghi thư mục để đảm bảo tính chính xác của cấu trúc.

+ Kiểu file đặc biệt: Đây là cơ chế mà UNIX sử dụng để truy nhập tới các thiết bị vào ra. Mỗi thiết bị vào ra trong UNIX đều được coi như là một file trong hệ thống file. Ta có thể truy nhập tới thiết bị vật lý thông qua việc truy nhập các file này. Người ta chia kiểu này làm hai loại dựa trên cách truy nhập tới chúng, đó là kiểu ký tự (character, ví dụ như file ứng với cổng nối tiếp) và kiểu khối (block, ví dụ như file ứng với ổ đĩa).

+ Kiểu file móc nối(symbolic link): Đây là file chứa đường dẫn tới một file khác. Cơ chế này cho phép ta truy nhập tới một tập tin bằng nhiều tên khác nhau. Thực chất của nó là định nghĩa một file với một tên khác.

+ Kiểu FIFO: là một hàng đợi (queue) theo kiểu first-in-first-out hay còn được gọi là named pipe. FIFO được dùng để trao đổi dữ liệu giữa các tiến trình. Loại file này chỉ có trong hệ UNIX System V mà không có trong BSD UNIX.

+ Kiểu socket: là một cơ chế tạo ra các đầu cuối (endpoint) cho phép các tiến trình liên hệ với nhau. Khái niệm socket sẽ được đề cập tới trong phần sau.

-Quyền truy nhập file: Hệ thống bảo vệ file theo 3 lớp người sử dụng là chủ sở hữu, nhóm sở hữu và các người sử dụng khác. Mỗi lớp người sử dụng đều có 3 quyền đọc, ghi, và thực hiện. Các quyền này được thiết lập tách biệt nhau. Do thư mục là một kiểu file đặc biệt nên quyền truy nhập tới thư mục có thay đổi. Quyền đọc cho phép tiến trình được đọc thư mục, quyền ghi cho phép tạo ra hoặc xóa bỏ các phần tử của thư mục (thông qua lệnh creat, mknod, link hay unlink), quyền thực hiện cho phép tiến trình tìm kiếm tên file trong thư mục.

- Thời gian: Lưu trữ thời gian mà file bị thay đổi gần nhất, thời gian file được truy cập gần nhất và thời gian inode bị thay đổi gần nhất.

- Số file liên kết: Thể hiện số file có trong cấu trúc cây thư mục.

Bảng địa chỉ các khối dữ liệu: Mặc dù người sử dụng xử lý file như một chuỗi liên tiếp các byte nhưng trong kernel lưu trữ dữ liệu trên những khối không liên tiếp. inode phải xác định các khối chứa dữ liệu của file. Bảng này được mã hoá khá phức tạp để có thể chứa một số lượng địa chỉ thay đổi nhưng kích thước bảng lại không thay đổi.

-Kích thước file: Lưu giữ chính xác kích thước thực của file.

Chú ý : Inode hoàn toàn không lưu giữ tên file và không thể xác định đường dẫn tới file thông qua inode. Khi inode được đọc vào bộ nhớ, một số trường được thêm vào làm cho inode trong bộ nhớ (in-core inode) khác với inode trên đĩa.

3.4. Cấu hình phần cứng

3.4.1. Cấu hình DHCP Server

Để cấu hình DHCP server bạn cần phải cài package dhcpd.*.rpm này trong đĩa CD Linux. Cài đặt DHCP bằng lệnh: `#rpm -ivh dhcpd.*.rpm`

Để hoàn thành việc cấu hình DHCP bạn cần phải tạo ra tập tin cấu hình /etc/dhcpd.conf và chỉnh sửa tập tin này. Ví dụ về nội dung cấu hình chính của tập tin dhcpd.conf

```

ddns-update-style interim;
default-lease-time 600;
max-lease-time 7200;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255; option routers 192.168.1.254;
option domain-name-servers 192.168.1.1, 192.168.1.2;
option domain-name "example.com";
subnet 192.168.1.0 netmask 255.255.255.0 {
range 192.168.1.10 192.168.1.100;
}

```

Tập tin /var/lib/dhcp/dhcpd.leases. Tập tin này được sử dụng bởi daemon dhcpd để lưu những thông tin về các địa chỉ IP đã được cấp phát

3.4.2. Cấu hình Web Server

Các tập tin và thư mục cấu hình của Apache :

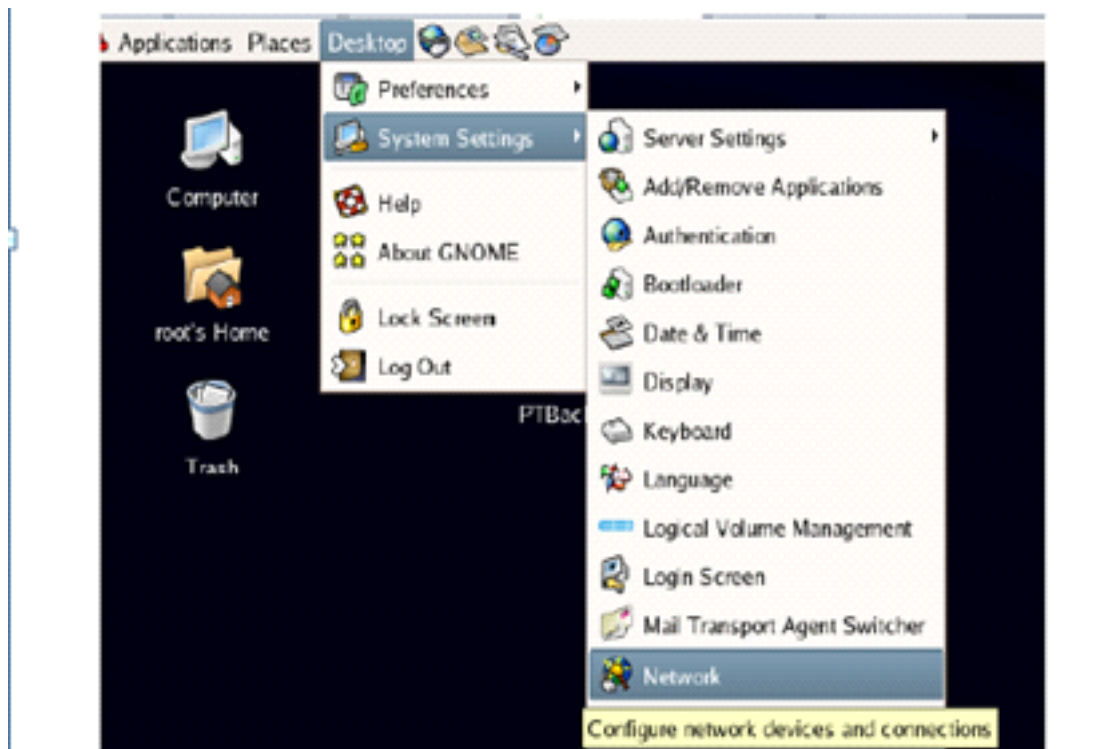
- /etc/httpd/conf: thư mục lưu giữ các tập tin cấu hình như httpd.conf.
- /etc/httpd/modules : lưu các module của Web Server.
- /etc/httpd/logs : lưu các tập tin log của Apache.
- /var/www/html : lưu các trang Web.
- /var/www/cgi-bin : lưu các script sử dụng cho các trang Web.

Tập tin cấu hình Apache được tạo thành từ nhiều chỉ dẫn (directive) khác nhau. Mỗi dòng/một đoạn là một directive và phục vụ cho một cấu hình riêng biệt. Có những directive có ảnh hưởng với nhau. Những dòng bắt đầu bằng dấu # là những dòng chú thích.

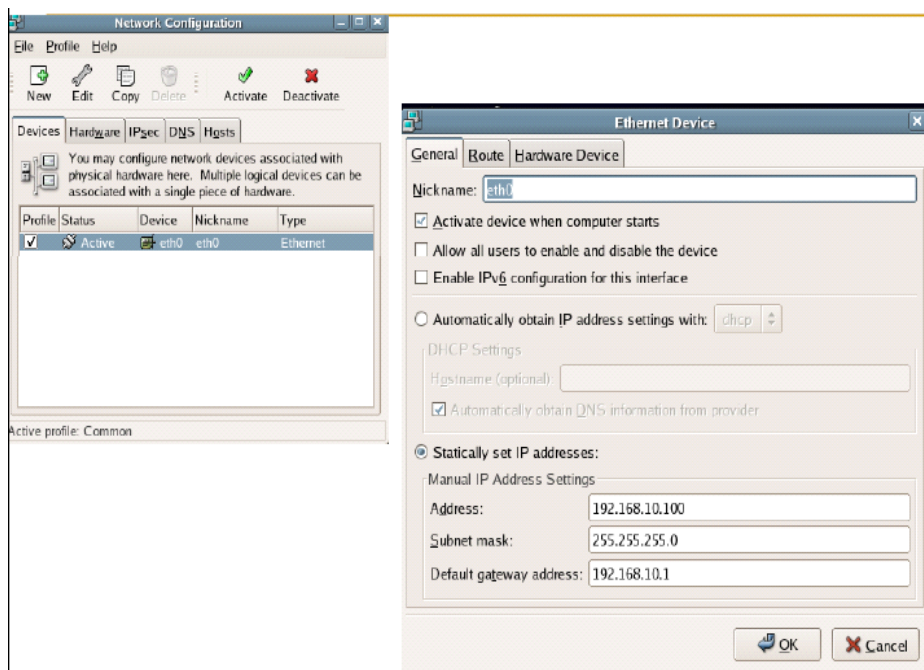
3.4.3. Network Card

- Đặt địa chỉ cho card mạng:

Để đặt địa chỉ cho Card mạng ta có thể sử dụng lệnh GUI

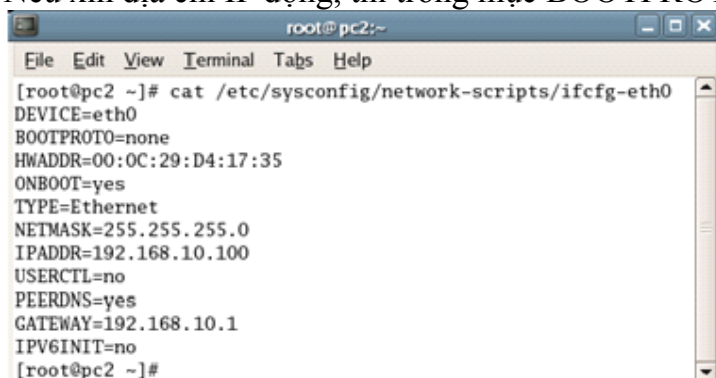


Hình 3.4.3.1 Cửa sổ thiết lập



Hình 3.4.3.2 Thiết lập thông số

Nếu xin địa chỉ IP động, thì trong mục BOOTPROTO sẽ có giá trị cho DHCP



Hình 3.4.3.3 Cửa sổ chạy lệnh

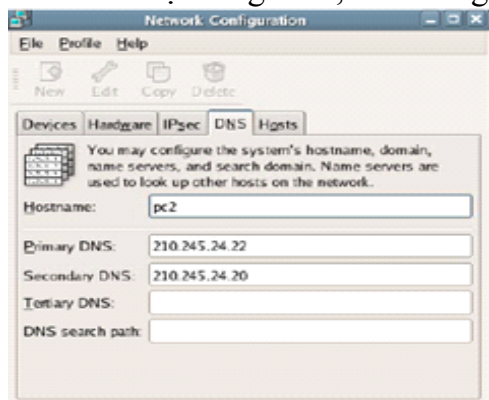
- Sửa thông số Card mạng qua file

+ Có thể thay đổi thông số Card mạng thông qua việc edit file /sysconfig/network-scripts-eth<n>, thường địa chỉ này với máy chỉ có Card mạng sẽ là ifcfg-eth0.

+ sau khi đổi thông số cho Card mạng ta nên thực hiện câu lệnh như sau:
#service network restart để các thay đổi được ghi nhận.

- Đặt địa chỉ name server:

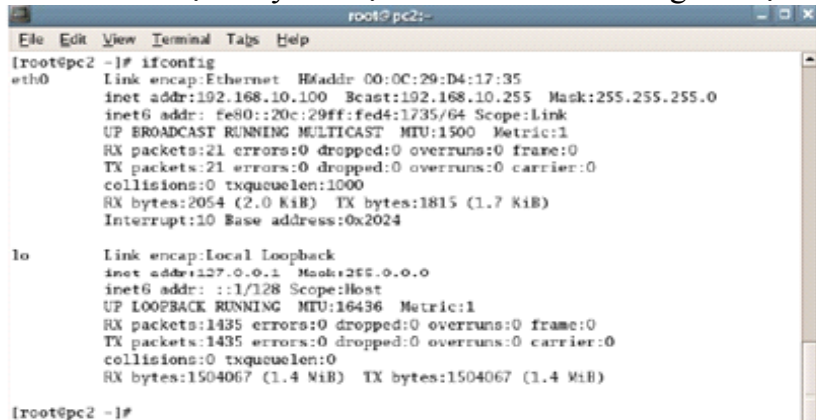
Ta có thể đặt bằng GUI, nằm cùng mục với địa chỉ IP



Hình 3.4.3.4 Đặt name server

- Xem, thay đổi địa chỉ IP

Muốn xem hoặc thay đổi địa chỉ IP ta có thể dùng câu lệnh như sau:



```
root@pc2:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:D4:17:35
          inet addr:192.168.10.100  Bcast:192.168.10.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fed4:1735/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:21 errors:0 dropped:0 overruns:0 frame:0
          TX packets:21 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2054 (2.0 KiB)  TX bytes:1815 (1.7 KiB)
          Interrupt:10 Base address:0x2024

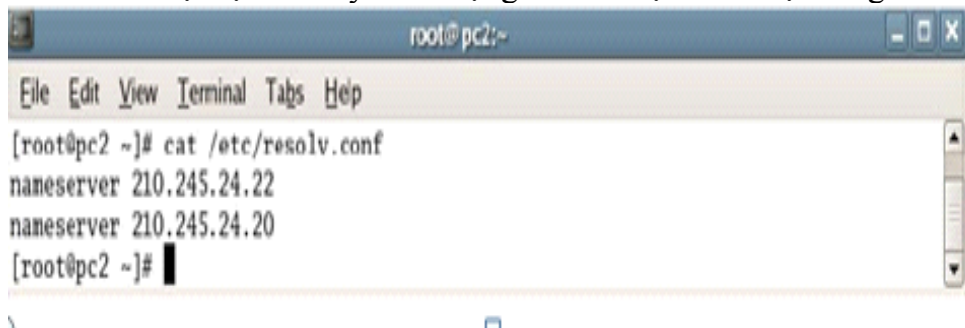
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1435 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1435 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1504067 (1.4 MiB)  TX bytes:1504067 (1.4 MiB)

root@pc2:~#
```

Hình 3.4.3.5 Thay đổi IP

- Đặt địa chỉ name server qua file

Nếu muốn đặt địa chỉ này ta sử dụng các câu lệnh sửa nội dung file /etc/resolv.conf



```
root@pc2:~# cat /etc/resolv.conf
nameserver 210.245.24.22
nameserver 210.245.24.20
root@pc2:~#
```

Hình 3.4.3.6 Đặt name server

- Một số câu lệnh kiểm tra kết nối mạng

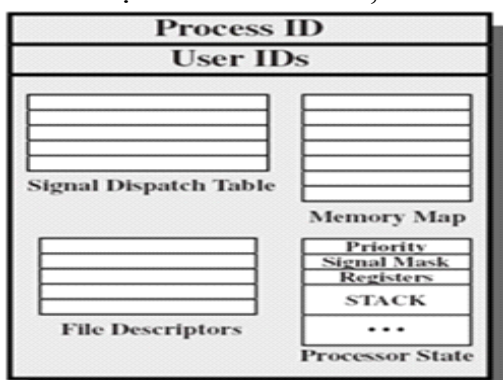
- Ping
- Traceroute
- Nslookup
- dig

3.5. Quản lý tiến trình

3.5.1. Khái niệm

Linux là hệ điều hành đa người dùng, đa tiến trình, bất cứ chương trình nào đang chạy đều được coi là một tiến trình. Tiến trình (process) là một chương trình đơn chạy trên không gian địa chỉ ảo của nó.

Cần phân biệt tiến trình với lệnh, một dòng lệnh shell có thể sinh ra nhiều tiến trình. Có thể có nhiều tiến trình cùng chạy một lúc. Ví dụ dòng lệnh `ls -l | sort | more` sẽ khởi tạo ba tiến trình: `ls`, `sort` và `more`



Hình 3.5.1.1. Process ID

Có 3 loại tiến trình chính trên Linux :

-Tiến trình với đối thoại (Interactive processes): là tiến trình khởi động và quản lý bởi shell, kể cả tiến trình forthground hoặc background.

-Tiến trình batch (Batch processes): Tiến trình không gắn liền đến bàn điều khiển (terminal) và được nằm trong hàng đợi để lần lượt thực hiện.

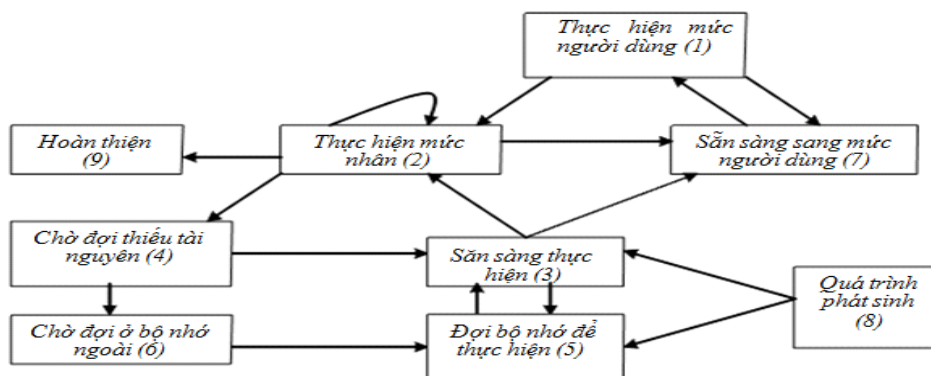
-Tiến trình ẩn trên bộ nhớ (Daemon processes): Là các tiến trình chạy dưới nền (background). Các tiến trình này thường được khởi động từ đầu. Đa số các chương trình server cho các dịch vụ chạy theo phương thức này. Đây là các chương trình sau khi được gọi lên bộ nhớ, đợi thụ động các yêu cầu chương trình khách (client) để trả lời sau các cổng xác định (cổng là khái niệm gắn liền với giao thức TCP/IP BSD socket. Chúng ta sẽ giải thích rõ trong phần TCP/IP). Hầu hết các dịch vụ trên Internet như mail, Web, Domain Name Service ... chạy theo nguyên tắc này. Các chương trình được gọi là các chương trình daemon và tên của nó thường kết thúc bằng ký tự “d” như named, inetd ... Ký tự “d” cuối được phát âm rời ra như “ê” trong tiếng việt. Ví dụ named được phát âm là “nêm ê”.

Tiến trình có thể trải qua nhiều trạng thái khác nhau và tại một thời điểm một tiến trình rơi vào một trong các trạng thái đó. Bảng dưới đây giới thiệu các trạng thái cơ bản của tiến trình trong Linux.

Bảng 3.5.1.1 Bảng trạng thái tiến trình

Kí hiệu	Ý nghĩa
D	(uninterruptible sleep) ở trạng thái này tiến trình bị treo và không thể chạy lại nó bằng một tín hiệu
R	(runnable) trạng thái sẵn sàng thực hiện, tức là tiến trình có thể thực hiện được nhưng chờ đến lượt thực hiện vì một tiến trình khác đang có CPU
S	(sleeping) trạng thái tạm dừng, tức là tiến trình tạm dừng không hoạt động (20 giây hoặc ít hơn)
T	(traced or stopped) trạng thái dừng, tiến trình có thể bị treo bởi một tiến trình ngoài
Z	(zombie process) tiến trình đã kết thúc thực hiện, nhưng nó vẫn được tham chiếu trong hệ thống
W	Không có các trạng thường trú
<	Tiến trình có mức ưu tiên cao hơn
N	Tiến trình có mức ưu tiên thấp hơn
L	Có các trang khóa bên trong bộ nhớ

Sơ đồ biểu diễn các trạng thái và việc chuyển trạng thái trong UNIX được trình bày trong hình dưới đây (Số hiệu trạng thái quá trình xem trong hình vẽ).



Hình 3.5.1.2 Số hiệu trạng thái quá trình

Khi quá trình được phát sinh nó ở trạng thái (8), tùy thuộc vào tình trạng bộ nhớ quá trình được phân phối bộ nhớ trong (3) hay bộ nhớ ngoài (5). Trạng thái (3) thể hiện quá trình đã sẵn sàng thực hiện, các thành phần của nó đã ở bộ nhớ trong chờ đợi CPU để thực hiện.

Việc thực hiện tiếp theo tùy thuộc vào trạng thái trước đó của nó. Nếu lần đầu phát sinh, nó cần đi tới thực hiện mức nhân để hoàn thiện công việc lời gọi fork sẽ từ trạng thái (3) sang trạng thái (1), trong trường hợp khác, từ trạng thái (3) nó đi tới trạng thái chờ đợi CPU ở mức người dùng (7).

Trong trạng thái thực hiện ở mức người dùng (1), quá trình đi tới trạng thái (2) khi gặp lời gọi hệ thống hoặc hiện tượng ngắt xảy ra. Từ trạng thái (1) tới trạng thái (7) khi hết lượng tử thời gian.

Trạng thái (4) là trạng thái chờ đợi trong bộ nhớ còn trạng thái (6) thể hiện việc chờ đợi trong bộ nhớ ngoài.

Cung chuyển từ trạng thái (2) vào ngay trạng thái (2) xảy ra khi ở quá trình ở trạng thái thực hiện mức nhân, nhân hệ thống gọi các hàm xử lý ngắt tương ứng.

3.5.2. Các lệnh cơ bản trong quản lý tiến trình

- Sử dụng lệnh ps

Linux cung cấp cho người dùng hai cách thức nhận biết có những chương trình nào đang chạy trong hệ thống. Cách dễ hơn, đó là lệnh jobs sẽ cho biết các quá trình nào đã dừng hoặc là được chạy trong chế độ nền.

Cách phức tạp hơn là sử dụng lệnh ps. Lệnh này cho biết thông tin đầy đủ nhất về các quá trình đang chạy trên hệ thống.

Ví dụ:

```
# ps
PID  TTY  TIME  CMD
7813 pts/0      00:00:00 bash
7908 pts/0      00:00:00 ps
#
```

(PID - chỉ số của tiến trình, TTY - tên thiết bị đầu cuối trên đó tiến trình được thực hiện, TIME - thời gian để chạy tiến trình, CMD - lệnh khởi tạo tiến trình).
Cú pháp lệnh ps: ps [tùy-chọn]

Lệnh ps có một lượng quá phong phú các tùy chọn được chia ra làm nhiều loại. Dưới đây là một số các tùy chọn hay dùng.

Các tùy chọn đơn giản:

- A, -e : chọn để hiển thị tất cả các tiến trình.
- T : chọn để hiển thị các tiến trình trên trạm cuối đang chạy.
- a : chọn để hiển thị tất cả các tiến trình trên một trạm cuối, bao gồm cả các tiến trình của những người dùng khác.
- r : chỉ hiển thị tiến trình đang được chạy. Chọn theo danh sách:
- C : chọn hiển thị các tiến trình theo tên lệnh.
- G : hiển thị các tiến trình theo chỉ số nhóm người dùng.
- U : hiển thị các tiến trình theo tên hoặc chỉ số của người dùng thực sự (người dùng khởi động tiến trình).
- p : hiển thị các tiến trình theo chỉ số của tiến trình.
- s : hiển thị các tiến trình thuộc về một phiên làm việc.
- t : hiển thị các tiến trình thuộc một trạm cuối.
- u : hiển thị các tiến trình theo tên và chỉ số của người dùng Thiết đặt định dạng được đưa ra của các tiến trình:

– f : hiển thị thông tin về tiến trình với các trường sau UID - chỉ số người dùng, PID - chỉ số tiến trình, PPID - chỉ số tiến trình khởi tạo ra tiến trình, C - , STIME - thời gian khởi tạo tiến trình, TTY - tên thiết bị đầu cuối trên đó tiến trình được chạy, TIME , thời gian để thực hiện tiến trình, CMD - lệnh khởi tạo tiến trình– l : hiển thị đầy đủ các thông tin về tiến trình với các trường F, S, UID, PID, PPID, C, PRI, NI, ADDR, SZ, WCHAN, TTY, TIME, CMD

– o xâu-chọn : hiển thị các thông tin về tiến trình theo dạng do người dùng tự chọn thông qua xâu-chọn các kí hiệu điều khiển hiển thị có các dạng như sau:

%C, %cpu % CPU được sử dụng cho tiến trình
 %mem % bộ nhớ được sử dụng để chạy tiến trình
 %G tên nhóm người dùng
 %P chỉ số của tiến trình cha khởi động ra tiến trình con
 %U định danh người dùng
 %c lệnh tạo ra tiến trình
 %p chỉ số của tiến trình
 %x thời gian để chạy tiến trình
 %y thiết bị đầu cuối trên đó tiến trình được thực hiện

Ví dụ: muốn xem các thông tin như tên người dùng, tên nhóm, chỉ số tiến trình, chỉ số tiến trình khởi tạo ra tiến trình, tên thiết bị đầu cuối, thời gian chạy tiến trình, lệnh khởi tạo tiến trình, hãy gõ lệnh:

```
# ps -o '%U %G %p %P %y %x %c'
USER GROUP PID PPID TTY TIME COMMAND
root root 1929 1927 pts/1 00:00:00 bash
root root 2279 1929 pts/1 00:00:00 ps
```

Nếu muốn xem người dùng “X” nào đó đang sử dụng những tiến trình nào ta có thể sử dụng lệnh sau: ps -u X

- Lệnh hủy tiến trình kill

Trong một số trường hợp, sử dụng lệnh kill để hủy bỏ một tiến trình. Điều quan trọng nhất khi sử dụng lệnh kill là phải xác định được chỉ số của tiến trình mà chúng ta muốn hủy.

Cú pháp lệnh: kill [tùy-chọn] <chỉ-số-của-tiến-trình>kill -l [tín hiệu]

Lệnh kill sẽ gửi một tín hiệu đến tiến trình được chỉ ra. Nếu không chỉ ra một tín hiệu nào thì ngầm định là tín hiệu TERM sẽ được gửi.

Một số tùy chọn:

– s xác định tín hiệu được gửi. Tín hiệu có thể là số hoặc tên của tín hiệu.

Dưới đây là một số tín hiệu hay dùng:

2	SIGINT	(interrupt) đây là tín hiệu được gửi khi nhấn CTRL+c
9	SIGKILL	(kill) tín hiệu này sẽ dừng tiến trình ngay lập tức
15	SIGTERM	tín hiệu này yêu cầu dừng tiến trình ngay lập tức, nhưng cho phép chương trình xóa các file tạm.
Số	Tên	Ý nghĩa
1	SIGHUP	(hang up) đây là tín hiệu được gửi đến tất cả các tiến trình đang chạy trước khi logout khỏi hệ thống

- p lệnh kill sẽ chỉ đưa ra chỉ số của tiến trình mà không gửi một tín hiệu nào.
- l hiển thị danh sách các tín hiệu mà lệnh kill có thể gửi đến các tiến trình (các tín hiệu này có trong file /usr/include/Linux/signal.h)

Ví dụ:

```
# ps
PID TTY TIME CMD
2240 pts/2 00:00:00 bash
2276 pts/2 00:00:00 man
2277 pts/2 00:00:00 more
2280 pts/2 00:00:00 sh
2281 pts/2 00:00:00 sh
```

kill 2277

```
PID TTY TIME CMD
2240 pts/2 00:00:00 bash
2276 pts/2 00:00:00 man
2280 pts/2 00:00:00 sh
2281 pts/2 00:00:00 sh
```

- **Lệnh sleep ngừng hoạt động một thời gian**

Nếu muốn cho máy nghỉ một thời gian mà không muốn tắt vì ngại khởi động lại thì cần dùng lệnh sleep.

Cú pháp: sleep [tùy-chọn]...

NUMBER[SUFFIX]

- NUMBER: số giây(s) ngừng hoạt động.
- SUFFIX : có thể là giây(s) hoặc phút(m) hoặc giờ hoặc ngày(d)

Các tùy chọn:

- -help hiển thị trợ giúp và thoát
- - version hiển thị thông tin về phiên bản và thoát

- **Xem cây tiến trình với lệnh pstree**

Đã biết lệnh để xem các tiến trình đang chạy trên hệ thống, tuy nhiên trong Linux còn có một lệnh cho phép có thể nhìn thấy mức độ phân cấp của các tiến trình, đó là lệnh pstree.

Cú pháp lệnh: pstree [tùy-chọn] [pid | người-dùng]

Lệnh pstree sẽ hiển thị các tiến trình đang chạy dưới dạng cây tiến trình. Gốc của cây tiến trình thường là init. Nếu đưa ra tên của một người dùng thì cây của các tiến trình do người dùng đó sở hữu sẽ được đưa ra. pstree thường gộp các nhánh tiến trình trùng nhau vào trong dấu ngoặc vuông, ví dụ:

```
nit +-getty
```

```
|_getty
   |_getty
```

thành

```
init ---4*[getty]
```

- a chỉ ra tham số dòng lệnh. Nếu dòng lệnh của một tiến trình được tráo đổi ra bên ngoài, nó được đưa vào trong dấu ngoặc đơn.
- c không thể thu gọn các cây con đồng nhất. Mặc định, các cây con sẽ được thu gọn khi có thể
- h hiển thị tiến trình hiện thời và "tổ tiên" của nó với màu sáng trắng

- H giống như tùy chọn -h, nhưng tiến trình con của tiến trình hiện thời không có màu sáng trắng
- l hiển thị dòng dài.
- n sắp xếp các tiến trình cùng một tổ tiên theo chỉ số tiến trình thay cho sắp xếp theo tên

Ví dụ:

```
# pstree
init--+-apmd
|-atd
|  |-automount
|  |-crond
|  |-enlightenment
|  |-gdm--X
|      `--gdm---gnome-session
|-gen_util_applet
|-gmc
```

- *Lệnh thiết lập độ ưu tiên của tiến trình*

Lệnh nice

Ngoài các lệnh xem và hủy bỏ tiến trình, trong Linux còn có hai lệnh liên quan đến độ ưu tiên của tiến trình, đó là lệnh nice và lệnh renice.

Để chạy một chương trình với độ ưu tiên định trước, hãy sử dụng lệnh nice. Cú pháp lệnh: nice [tùy-chọn] [lệnh [tham-số] ...]

Lệnh nice sẽ chạy một chương trình (lệnh) theo độ ưu tiên đã sắp xếp. Nếu không có lệnh, mức độ ưu tiên hiện tại sẽ hiển thị. Độ ưu tiên được sắp xếp từ -20 (mức ưu tiên cao nhất) đến 19 (mức ưu tiên thấp nhất).

- ADJUST : tăng độ ưu tiên theo ADJUST đầu tiên
- - help : hiển thị trang trợ giúp và thoát

Lệnh renice

Để thay đổi độ ưu tiên của một tiến trình đang chạy, hãy sử dụng lệnh renice.

Cú pháp lệnh: renice <độ-ưu-tiên> [tùy-chọn]

Lệnh renice sẽ thay đổi mức độ ưu tiên của một hoặc nhiều tiến trình đang chạy.

- g : thay đổi quyền ưu tiên theo nhóm người dùng
- p : thay đổi quyền ưu tiên theo chỉ số của tiến trình
- u : thay đổi quyền ưu tiên theo tên người dùng

Ví dụ:

```
# renice +1 987 -u daemon root -p 32
```

lệnh trên sẽ thay đổi mức độ ưu tiên của tiến trình có chỉ số là 987 và 32, và tất cả các tiến trình do người dùng daemon và root sở hữu.

- *Lệnh lsof liệt kê các files được mở bởi các tiến trình khác*

Liệt kê các files, sockets, pipes đang mở, đang được sử dụng bởi các tiến trình khác, nếu muốn biết tất cả các tiến trình đang sử dụng shell bash sử dụng lệnh như sau:

```
# lsof /bin/bash
```

```
COMMAND PID  USERFD  TYPE  DEVICE  SIZENODENAME
bash1838 khanhduong txt   REG 252,0729040170156/bin/bash
```

3.6. Tập tin và thư mục

3.6.1. Một số khái niệm

Người dùng đã từng làm việc với hệ điều hành DOS/Windows thì rất quen biết với các khái niệm: tập tin (file), thư mục, thư mục hiện thời ... Để đảm bảo tính hệ

thông và thuận tiện cho người dùng chưa từng làm việc thành thạo với một hệ điều hành nào khác, chương này vẫn giới thiệu về các khái niệm này một cách sơ bộ.

Một đối tượng điển hình trong các hệ điều hành đó là file. File là một tập hợp dữ liệu có tổ chức được hệ điều hành quản lý theo yêu cầu của người dùng. Cách tổ chức dữ liệu trong file thuộc về chủ của nó là người đã tạo ra file. File có thể là một văn bản (trường hợp đặc biệt là chương trình nguồn trên C, PASCAL, shell script ...), một chương trình ngôn ngữ máy, một tập hợp dữ liệu ... Hệ điều hành tổ chức việc lưu trữ nội dung file trên các thiết bị nhớ lâu dài (chẳng hạn đĩa từ) và đảm bảo các thao tác lên file. Chính vì có hệ điều hành đảm bảo các chức năng liên quan đến file nên người dùng không cần biết file của mình lưu ở vùng nào trên đĩa từ, bằng cách nào đọc/ghi lên các vùng của đĩa từ mà vẫn thực hiện được yêu cầu tìm kiếm, xử lý lên các file.

Hệ điều hành quản lý file theo tên gọi của file (tên file) và một số thuộc tính liên quan đến file. Trước khi giới thiệu một số nội dung liên quan đến tên file và tên thư mục, chúng ta giới thiệu sơ bộ về khái niệm thư mục.

Để làm việc được với các file, hệ điều hành không chỉ quản lý nội dung file mà còn phải quản lý các thông tin liên quan đến các file. Thư mục (directory) là đối tượng được dùng để chứa thông tin về các file, hay nói theo một cách khác, thư mục chứa các file. Các thư mục cũng được hệ điều hành quản lý trên vật dẫn ngoài và vì vậy, theo nghĩa này, thư mục cũng được coi là file song trong một số trường hợp để phân biệt với "file" thư mục, chúng ta dùng thuật ngữ file thông thường. Khác với file thông thường, hệ điều hành lại quan tâm đến nội dung của thư mục.

Một số nội dung sau đây liên quan đến tên file (bao gồm cả tên thư mục):

Tên file trong Linux có thể dài tới 256 ký tự, bao gồm các chữ cái, chữ số, dấu gạch nối, gạch chân, dấu chấm. Tên thư mục/file trong Linux có thể có nhiều hơn một dấu chấm, ví dụ: This_is.a.VERY_long.filename. Nếu trong tên file có dấu chấm "." thì xâu con của tên file từ dấu chấm cuối cùng được gọi là phần mở rộng của tên file (hoặc file). Ví dụ, tên file trên đây có phần mở rộng là .filename. Chú ý rằng khái niệm phần mở rộng ở đây không mang ý nghĩa như một số hệ điều hành khác (chẳng hạn như MS-DOS).

*** Lưu ý:**

Chúng ta nên lưu ý rằng, không phải ký tự nào cũng có nghĩa. Nếu có hai file chỉ khác nhau ở ký tự cuối cùng, thì đối với Linux, đó là hai file có thể trùng tên. Bởi lẽ, Linux chỉ lấy 32 hay 64 ký tự đầu tiên trong tên file mà thôi (tùy theo phiên bản Linux), phần tên file còn lại dành cho chủ của file, Linux theo dõi thông tin, nhưng thường không xem các ký tự đứng sau ký tự thứ 33 hay 65 là quan trọng đối với nó.

Xin nhắc lại lưu ý về phân biệt chữ hoa và chữ thường đối với tên thư mục/file, ví dụ hai file FILENAME.tar.gz và filename.tar.gz là hai file khác nhau.

+ Nếu trong tên thư mục/file có chứa khoảng trống, sẽ phải đặt tên thư mục/file vào trong cặp dấu nháy kép để sử dụng thư mục/file đó. Ví dụ, để tạo thư mục có tên là "My document" chẳng hạn, hãy đánh dòng lệnh sau: `# mkdir "My document"`

+ Một số ký tự sau không được sử dụng trong tên thư mục/file: `!, *, $, &, # ...`

Tập hợp tất cả các file có trong hệ điều hành được gọi là hệ thống file là một hệ thống thống nhất. Bởi chính từ cách thức sử dụng thư mục, hệ thống file được tổ chức logic theo dạng hình cây: Hệ thống file được xuất phát từ một thư mục gốc (được kí hiệu là "/") và cho phép tạo ra thư mục con trong một thư mục bất kỳ. Thông thường, khi khởi tạo Linux đã có ngay hệ thống file của nó.

3.6.2. Các lệnh trong hệ thống tập tin

- Tạo thư mục với lệnh mkdir

Lệnh *mkdir* tạo một thư mục, cú pháp: **mkdir [tùy-chọn] <thư-mục>**

Lệnh này cho phép tạo một thư mục mới nếu thư mục đó chưa thực sự tồn tại. Để tạo một thư mục, cần đặc tả tên và vị trí của nó trên hệ thống file (vị trí mặc định là thư mục hiện thời).

Ví dụ: nếu muốn tạo thư mục test trong thư mục home, hãy gõ lệnh sau: *mkdir /home/test*

- Thay đổi thư mục làm việc hiện thời với lệnh cd

Cú pháp lệnh: *cd*

Chuyển đến thư mục /usr/include : *\$cd /usr/include*

Chuyển trở lại thư mục "home": *\$cd*

Chuyển đến thư mục cha: *\$cd..*

- Xem nội dung thư mục với lệnh ls

Sử dụng lệnh *ls* và một số các tùy chọn của nó là có thể biết được mọi thông tin về một thư mục.

Cú pháp lệnh: *# ls [tùy-chọn] [file]*

Lệnh này đưa ra danh sách các file liên quan đến tham số file trong lệnh. Trường hợp phổ biến tham số file là một thư mục, tuy nhiên trong một số trường hợp khác, tham số file xác định nhóm (khi sử dụng các mô tả nhóm *, ? và cặp [và]); nếu không có tham số file, mặc định danh sách các file có trong thư mục hiện thời sẽ được hiển thị.

Các tùy chọn của lệnh:

- a : liệt kê tất cả các file, bao gồm cả file ẩn.
- l : đưa ra thông tin đầy đủ nhất về các file và thư mục.
- s : chỉ ra kích thước của file, tính theo khối (1 khối = 1024 byte).
- F :

- Xóa thư mục với lệnh rmdir

Lệnh *rmdir* được dùng để xóa bỏ một thư mục.

Cú pháp lệnh: *rmdir [tùy-chọn] <thư-mục>*

Có thể xóa bỏ bất kỳ thư mục nào nếu có quyền đó. Lưu ý rằng, thư mục chỉ bị xóa khi nó "rỗng", tức là không tồn tại file hay thư mục con nào trong đó.

Không có cách gì khôi phục lại các thư mục đã bị xóa, vì thế hãy suy nghĩ cẩn thận trước khi quyết định xóa một thư mục.

Các tùy chọn của lệnh:

- - ignore-fail-on-non-empty : bỏ qua các lỗi nếu xóa một thư mục không rỗng.
- p, --parents : xóa bỏ một thư mục, sau đó lần lượt xóa bỏ tiếp các thư mục có trên đường dẫn chứa thư mục vừa xóa. .

Ví dụ, dòng lệnh *rmdir -p /a/b/c* sẽ tương đương với ba dòng lệnh *rmdir /a/b/c*, *rmdir /a/b*, *rmdir /a* (với điều kiện các thư mục là rỗng).

- - verbose : đưa ra thông báo khi xóa một thư mục.
- - help : hiển thị trang trợ giúp và thoát.

Ví dụ:

```
# rmdir -p /test/test1/test2
```

```
rmdir: /: No such file or directory
```

Dòng lệnh trên sẽ lần lượt xóa ba thư mục test2, test1, test và hiển thị thông báo trên màn hình kết quả của lệnh.

- Xem đường dẫn thư mục hiện thời với lệnh pwd

Cú pháp lệnh: **pwd**

Lệnh này cho biết hiện người dùng đang ở trong thư mục nào và hiện ra theo dạng một đường dẫn tuyệt đối.

Ví dụ: gõ lệnh pwd tại dấu nhắc lệnh sau khi người dùng duonglk vừa đăng nhập thì màn hình hiển thị như sau:

```
# pwd
/home/duonglk
```

- Lệnh đổi tên thư mục với lệnh mv

Cú pháp lệnh: mv <tên-cũ> <tên-mới>

Lệnh này cho phép đổi tên một thư mục từ tên-cũ thành tên-mới.

Ví dụ: # mv Tongket thongke sẽ đổi tên thư mục Tongket thành thongke .

Nếu sử dụng lệnh mv để đổi tên một thư mục với một cái tên đã được đặt cho một file thì lệnh sẽ gặp lỗi. Nếu tên mới trùng với tên một thư mục đang tồn tại thì nội dung của thư mục được đổi tên sẽ ghi đè lên nội dung của thư mục trùng tên.

- Tạo file (tập tin) với lệnh touch

Lệnh touch có nhiều chức năng, trong đó một chức năng là giúp tạo file mới trên hệ thống: touch rất hữu ích cho việc tổ chức một tập hợp các file mới.

Cú pháp lệnh: touch <file>

Thực chất lệnh này có tác dụng dùng để cập nhật thời gian truy nhập và sửa chữa lần cuối của một file. Vì lý do này, các file được tạo bằng lệnh touch đều được sắp xếp theo thời gian sửa đổi. Nếu sử dụng lệnh touch đối với một file chưa tồn tại, chương trình sẽ tạo ra file đó. Sử dụng bất kỳ trình soạn thảo nào để soạn thảo file mới.

Ví dụ: dùng lệnh touch để tạo file newfile: # touch newfile

- Tạo file với lệnh cat

Lệnh cat tuy đơn giản nhưng rất hữu dụng trong Linux. Chúng ta có thể sử dụng lệnh này để lấy thông tin từ đầu vào (bàn phím...) rồi kết xuất ra file hoặc các nguồn khác, hay để xem nội dung của một file..Phần này trình bày tác dụng của lệnh cat đối với việc tạo file.

Cú pháp lệnh: cat > filename

Theo ngầm định, lệnh này cho phép lấy thông tin đầu vào từ bàn phím rồi xuất ra màn hình. Soạn thảo nội dung của một file bằng lệnh cat tức là đã đổi hướng đầu ra của lệnh từ màn hình vào một file. Người dùng gõ nội dung của file ngay tại dấu nhắc màn hình và gõ CTRL+d để kết thúc việc soạn thảo.

Nhược điểm của cách tạo file này là nó không cho phép sửa lỗi, ví dụ nếu muốn sửa một lỗi chính tả trên một dòng, chỉ có cách là xóa đến vị trí của lỗi và gõ lại nội dung vừa bị xóa.

Ví dụ: tạo file newfile trong thư mục /home/vd bằng lệnh cat.

```
# cat > /home/vd/newfile
This is a
example of cat
command
^D
```

Sau khi soạn thảo xong, gõ Enter và CTRL+d để trở về dấu nhắc lệnh, nếu không gõ Enter thì phải gõ CTRL+d hai lần. Khi sử dụng lệnh này, nếu file chưa tồn tại thì sẽ tạo file mới, nếu file đó đã tồn tại thì sẽ xóa file cũ và tạo file mới. Có thể sử dụng luôn lệnh cat để xem nội dung của file vừa soạn thảo:

```
# cat /home/vd/newfile
This is a example of cat command
```

Để thêm nội dung vào phần cuối của file có sẵn dùng lệnh: cat >> filename.

Để tổng hợp hai tập tin thành một ta sử dụng cú pháp lệnh sau: \$cat file1 file2 > file3

- Xóa file với lệnh rm

Lệnh rm là lệnh rất "nguy hiểm" vì trong Linux không có lệnh khôi phục lại những gì đã

xóa, vì thế hãy cẩn trọng khi sử dụng lệnh này. Lệnh rm cho phép xóa bỏ một file hoặc nhiều file.

Cú pháp lệnh: rm [tùy-chọn] <file> ...

Lệnh rm cho phép xóa nhiều file cùng một lúc bằng cách chỉ ra tên của các file cần xóa trong dòng lệnh (hoặc dùng kí hiệu mô tả nhóm). Dùng lệnh # rm bak/*.h xóa mọi file với tên có hai kí hiệu cuối cùng là ".h" trong thư mục con bak.

- Sao chép tập tin với lệnh cp

Lệnh cp có hai dạng như sau:

cp [tùy-chọn] <file-nguồn> ... <file-đích>

cp [tùy-chọn] --target-directory=<thư-mục> <file-nguồn>...

Lệnh này cho phép sao file-nguồn thành file-đích hoặc sao chép từ nhiều file-nguồn vào một thư mục đích (tham số <file-đích> hay <thư-mục>). Dạng thứ hai là một cách viết khác đổi thứ tự hai tham số vị trí.

File đích được tạo ra có cùng kích thước và các quyền truy nhập như file nguồn, tuy nhiên file đích có thời gian tạo lập là thời điểm thực hiện lệnh nên các thuộc tính thời gian sẽ khác.

Ví dụ:

```
# cp /home/ftp/vd /home/test/vd1
```

Nếu ở vị trí đích, mô tả đầy đủ tên file đích thì nội dung file nguồn sẽ được sao chép sang file đích. Trong trường hợp chỉ đưa ra vị trí file đích được đặt trong thư mục nào thì tên của file nguồn sẽ là tên của file đích.

```
# cp /home/ftp/vd /home/test/
```

Trong ví dụ này, tên file đích sẽ là vd nghĩa là tạo một file mới /home/test/vd.

Nếu sử dụng lệnh này để sao một thư mục, sẽ có một thông báo được đưa ra cho biết nguồn là một thư mục và vì vậy không thể dùng lệnh cp để sao chép.

```
# cp . newdir
```

```
cp: .: omitting directory
```

Câu hỏi ôn tập chương

1. Trình bày các kiểu file và quy ước tên file trong Linux
2. Trình bày cấu trúc hệ thống file của Linux
3. Trình bày cấu hình DHCP server và Network Card trong linux?
4. Trình bày các lệnh cơ bản trong quản lý tiến trình
5. Trình bày các lệnh cơ bản trong hệ thống tập tin và thư mục

Chương 4: Cài đặt phần mềm trong hệ điều hành Linux

Mục tiêu:

- Trình bày được các kỹ năng cài đặt phần mềm trong hệ điều hành Linux;
- Biết sử dụng công cụ dòng lệnh RPM;
- Cài đặt được các gói dạng TAR;
- Có thái độ cẩn thận, nghiêm túc.

4.1. RPM là gì?

4.1.1. Khái niệm

RPM là viết tắt của chữ ‘Red Hat Package Management’. Đây là một trong những dạng gói phần mềm dễ dùng nhất. Các tập tin RPM thường có kết thúc bằng ‘.rpm’. Trong các bản Red Hat Linux, tên tập tin RPM thường tuân theo dạng sau: <tên gói>-<phiên bản>-<số hiệu patch>.<hệ máy>.rpm

Example 1.1. Tên gói

emacs-21.2-1.i386.rpm

Một số gói dùng để phát triển phần mềm có dạng: <tên gói>-devel-<phiên bản>-<số hiệu patch>.<hệ máy>.rpm.

Các gói dạng này thường chứa các tập tin thư viện và tập tin header, chỉ cần thiết để biên dịch chương trình dùng lệnh rpm.

Cách dùng thông thường để cài đặt gói rpm là (cần quyền root): rpm -Uvh <tập tin rpm>

Để biết thông tin về gói, ta dùng lệnh sau: rpm -qpi <tập tin rpm>

Thông thường khi cài đặt có thể bạn sẽ gặp lỗi ‘dependencies’.

Example 1.2. Lỗi dependency

error: failed dependencies:

libatk.so.12 is needed by libglade2-1.99.7-3

libgdk_pixbuf-1.3.so.14 is needed by libglade2-1.99.7-3

libgdk-x11-1.3.so.14 is needed by libglade2-1.99.7-3

libglib-1.3.so.14 is needed by libglade2-1.99.7-3

Bạn phải biết các tập tin libatk.so.12, libglib-1.3.so.14... nằm trong gói nào và cài đặt các gói đó trước. Để biết danh sách các tập tin trong một gói, ta dùng lệnh: rpm -qpl <tập tin rpm>

Gỡ cài đặt, dùng lệnh (cần quyền root): rpm -e <tên gói>

Lưu ý: tên gói không phải là tên tập tin rpm. Quá trình gỡ cài đặt có thể sẽ thất bại nếu việc gỡ cài đặt ứng dụng này có thể ảnh hưởng đến ứng dụng khác. Trong trường hợp đó, phải gỡ cài đặt tất cả các ứng dụng cần đến ứng dụng bạn định gỡ cài đặt trước khi có thể gỡ cài đặt chính ứng dụng đó.

4.1.2. Quản lý gói

* Liệt kê danh sách tất cả các gói đã cài đặt:

\$rpm -qa

* Kiểm tra gói <tên gói> có cài đặt chưa:

\$rpm -q <tên gói>

* Liệt kê danh sách tập tin của một gói:

\$rpm -ql <tên gói>

* Cho biết gói nào chứa tập tin <tập tin>:

\$rpm -qf <tập tin>

4.1.3. Đặc tính của RPM

Để hiểu rõ hơn đặc tính sử dụng của RPM, chúng ta xem xét các mục đích của việc xây dựng RPM.

-Khả năng nâng cấp phần mềm: Với RPM ta có thể nâng cấp các thành phần riêng biệt của hệ thống mà không cần phải cài lại. Khi có một phiên bản mới của hệ điều hành dựa trên RPM (như RedHat Linux chẳng hạn) thì ta không phải cài lại hệ thống mà chỉ cần nâng cấp thôi. RPM cho phép nâng cấp hệ thống một cách tự động, thông minh. Các tập tin cấu hình được gìn giữ cẩn thận qua các lần nâng cấp, vì thế không sợ thay đổi các tùy chọn sẵn có của hệ thống được nâng cấp.

-Truy vấn thông tin hiệu quả: RPM cũng được thiết kế cho mục đích truy vấn các thông tin về các package trong hệ thống. Ta có thể tìm kiếm thông tin các package hoặc các tập tin cài đặt trong toàn bộ cơ sở dữ liệu. Cũng có thể hỏi tập tin cụ thể thuộc về package nào và nó ở đâu. Package RPM có các tập tin chứa các thông tin rất hữu ích về package này và nội dung của package. Các tập tin này cho phép người dùng tìm kiếm thông tin dễ dàng trong một package riêng lẻ.

-Thẩm tra hệ thống (System Verification): Một đặc tính rất mạnh của RPM là cho phép bạn thẩm tra lại các package. Nếu nghi ngờ một tập tin nào bị xóa hay bị thay thế trong package, có thể kiểm tra lại rất dễ dàng. Ta cần phải chú ý đến các dấu hiệu bất bình thường của hệ thống, nên kiểm tra và cài lại nếu cần thiết.

4.2. Sử dụng công cụ dòng lệnh RPM

RPM làm việc theo sáu chế độ khác nhau, trong đó ta có thể sử dụng năm chế độ từ dòng lệnh hoặc bằng gnoRPM, một công cụ căn cứ trên X Window, ta cũng có thể sử dụng GnomeRPM hay KDERPM. Các chế độ đó là cài đặt, tháo bỏ cài đặt, cập nhật, tìm kiếm, kiểm sát và xây dựng.

4.2.1. Lệnh rpm

Lưu ý rằng ta phải thực hiện rpm với người dùng quản trị (root). RPM có 5 chế độ thực hiện là cài đặt (installing), xóa (uninstalling), nâng cấp (upgrading), truy vấn (querying) và thẩm tra (verifying).

4.2.2. Cài đặt phần mềm bằng rpm

Package RPM thường chứa các tập tin giống như foo-1.0-1.i386.rpm Tên tập tin này bao gồm tên package (foo), phiên bản (1.0), số hiệu phiên bản (1), kiến trúc sử dụng (i386). Lệnh cài đặt :

```
# rpm -ivh tên-tập-tinRPM
```

Ví dụ:

```
#rpm -ivh foo-1.0-1.i386.rpm  
foo
```

Một số trường hợp lỗi khi cài đặt.

+ Package đã cài rồi.

+ Xung đột với tập tin cũ đã tồn tại.

+ Package phụ thuộc vào package khác. Ví dụ: package đã được cài đặt trước

```
# rpm -ivh foo-1.0-1.i386.rpm
```

```
foo package foo-1.0-1 is already installed
```

Nếu muốn cài chồng lên package đã cài rồi dùng lệnh thêm tham số --replacepks

```
#rpm -ivh --replacepks tên-tập-tin-package
```

Ví dụ:

```
# rpm -ivh --replacepks foo-1.0-1.i386.rpm
```

Ví dụ: xung đột với tập tin cũ đã tồn tại

```
# rpm -ivh foo-1.0-1.i386.rpm
```

```
foo /usr/bin/foo conflicts with file from bar-1.0-1
```

Để bỏ qua lỗi này, ta có thể cài đè lên bằng cách sử dụng tùy chọn --replacefiles.

```
# rpm -ivh --replacefiles foo-1.0-1.i386.rpm
```

Ví dụ: Package phụ thuộc vào package khác

```
# rpm -ivh foo-1.0-1.i386.rpm
```

failed dependencies:

bar is needed by foo-1.0-1

Giải quyết trường hợp này ta phải cài các package được yêu cầu. Nếu ta muốn tiếp tục cài mà không cài các package khác thì dùng tùy chọn --nodeps. Tuy nhiên lúc này có thể package của bạn cài có thể chạy không tốt.

4.2.3. Loại bỏ phần mềm đã cài đặt trong hệ thống

Xóa package thì đơn giản hơn cài. Lệnh xóa.

```
# rpm -e tên-package
```

Lưu ý là khi xóa chúng ta dùng tên-package chứ không dùng tên tập tin RPM. Ví dụ:

```
# rpm -e foo
```

removing these packages would break dependencies: foo is needed by bar-1.0-1

Nếu muốn xóa các package bỏ qua các lỗi, bạn dùng thêm tham số --nodeps. Tuy nhiên đây không phải là ý kiến hay, vì nếu chương trình bạn xóa có liên quan đến chương trình khác. Khi đó chương trình này sẽ hoạt động không được.

4.2.4. Nâng cấp phần mềm

Upgrade cũng tương tự như cài đặt mới.

```
# rpm -Uvh tên-tập-tinRPM
```

Ví dụ:

```
# rpm -Uvh foo-2.0-1.i386.rpm
```

foo

Khi upgrade RPM sẽ xóa các phiên bản cũ của package. ta có thể dùng lệnh này để cài đặt, khi đó sẽ không có phiên bản cũ nào bị xóa đi.

Khi RPM tự động nâng cấp với tập tin cấu hình, ta thấy chúng thường xuất hiện một thông báo như sau : saving /etc/foo.conf as /etc/foo.conf.rpmsave. Điều này có nghĩa là khi tập tin cấu hình của phiên bản cũ không tương thích với phiên bản mới thì chúng lưu lại và tạo tập tin cấu hình mới. Nâng cấp thực sự là sự kết hợp giữa Uninstall và Install. Vì thế khi upgrade cũng thường xảy ra các lỗi như khi Install và Uninstall và thêm một lỗi nữa là khi ta upgrade với phiên bản cũ hơn.

```
# rpm -Uvh foo-1.0-1.i386.rpm
```

foo package foo-2.0-1 (which is newer) is already installed

Trong trường hợp này ta thêm tham số --oldpackage

```
# rpm -Uvh --oldpackage foo-1.0-1.i386.rpm
```

foo #####

4.2.5. Truy vấn các phần mềm

Để truy vấn thông tin từ cơ sở dữ liệu của những package đã cài đặt ta dùng.

```
# rpm -q tên-package
```

Ví dụ:

```
# rpm -q foo
```

foo-2.0-1 //kết quả truy vấn

Thay vì xác định tên package, ta có thể sử dụng thêm một số tham số khác kết hợp với -q để xác định package mà ta muốn truy vấn, chúng được gọi là Package Specification Options

+ -a : Truy vấn tất cả các package.

+ -f<tập-tin>: Truy vấn những package chứa tập-tin. Khi xác định tập tin bạn phải chỉ rõ đường dẫn (ví dụ : /usr/bin/ls)

+ -p <tên-tập-tin-package> : Truy vấn package tên-tập-tin-package

Có một số cách xác định những thông tin hiển thị về package. Sau đây là các tùy chọn sử dụng để xác định loại thông tin cần tìm kiếm. Chúng được gọi là Information Selection Options

+ -i: xác định các thông tin về package bao gồm : tên, mô tả, phiên bản, kích thước,

ngày tạo, ngày cài đặt, nhà sản xuất ...

+ -l: Hiển thị những tập tin trong package.

+ -s: Hiển thị trạng thái của các tập tin trong package.

+ -d: hiển thị danh sách tập tin tài liệu cho package (ví dụ man, README, info file ...)

+ --c: hiển thị danh sách tập tin cấu hình.

4.3. Cài đặt các gói dạng TAR

Ngoài các phần mềm được đóng gói dạng file nhị phân(file *.rpm) còn có các phần mềm được cung cấp dạng file source code như: *.tar hoặc *.tgz. Thông thường để cài đặt phần mềm này ta cần phải dựa vào trợ giúp của file giúp đỡ trong từng chương trình hoặc phần mềm, các file(README or INSTALL,) này nằm trong các thư mục con của thư mục sau khi ta dùng lệnh tar để giải nén source. Để thực hiện việc cài đặt này ta thường làm các bước sau:

Bước 1: Giải nén file tar.

Ví dụ:

```
[root@bigboy tmp]# tar -xvzf linux-  
software-1.3.1.tar.gz      linux-software-  
1.3.1/  
linux-software-1.3.1/plugins-scripts/  
linux-software-1.3.1/linux-  
software-plugins.spec  
[root@bigboy tmp]#  
Tạo các thư mục con chứa các  
file cài đặt [root@bigboy  
tmp]# ls  
linux-software-1.3.1      linux-  
software-1.3.1.tar.gz [root@bigboy  
tmp]#
```

Bước 2: Chuyển vào thư mục con và tham khảo các file INSTALL, README.

Ví dụ:

```
[root@bigboy tmp]# cd linux-  
software-1.3.1  
[root@bigboy linux-software-  
1.3.1]# ls  
COPYING  install-sh  missingplugins  
depcomp  LEGAL  mkininstall  dirsplugins-scripts  
FAQ      lib linux-software.spec  README  
Helper.pm  Makefile.am  linux-software.spec.in  REQUIREMENTS  
INSTALL  Makefile.in  NEWS  subst.in  
[root@bigboy linux-software-1.3.1]#
```


Bước 3: Sau đó ta dựa vào chỉ dẫn trong file (INSTALL, README) để cài đặt phần mềm.

Ví dụ:

Cài đặt bộ gõ tiếng việt XVNKB-0.2.9a.tar.gz cho hệ thống ta thực hiện như sau:

4.3.1. Chuẩn bị cài đặt

Vào trang chủ của xvnkb : <http://xvnkb.sourceforge.net>

và down về gói cài đặt : xvnkb-0.2.9a.tar.gz

4.3.2. Tiến hành cài đặt

Down về 1 phân vùng trên linux

- Giải nén bằng lệnh :

```
# tar -jxvf xvnkb-0.2.9a.tar.gz
```

```
# cd ../xvnkb-0.2.9a/
```

```
# ./configure
```

- Thấy kết quả như sau thì được:

```
[root@localhost xvnkb-0.2.9a]# ./configure
```

```
Configuration for xvnkb 0.2.9a on Linux
```

```
Type "./configure --help" for more information
```

```
Checking uchar... no
```

```
Checking ushort... yes
```

```
Checking uint... yes
```

```
Checking ulong... yes
```

```
Checking dynamic linking loader... yes
```

```
Checking X11 lib... /usr/X11R6
```

```
Checking pkg-config... yes
```

```
Checking Xft... yes
```

```
Compile options:
```

```
Enable XFT: yes
```

```
Enable spell checking: yes
```

```
Enable extended keystroke: no
```

```
Enable ABC liked Telex keystroke: no
```

```
done.
```

- Để biên dịch phần mềm gõ lệnh sau:

```
#make
```

```
***** Nếu có lỗi như sau:
```

```
make[1]: Entering directory `/packages/xvnkb-0.2.9/tools'
```

```
make[1]: *** No rule to make target `/usr/X11R6/include/X11/X.h', needed by  
`xvnkb_ctrl.o'. Stop.
```

```
make[1]: Leaving directory `/packages/xvnkb-0.2.9/tools'
```

```
make: *** [tools] Error 2
```

```
***** Thì tạo link từ /usr/X11R6/include đến /usr/X11R6/include/X11 như sau:
```

```
[root@localhost xvnkb-0.2.9]# mkdir /usr/X11R6/include
```

```
[root@localhost xvnkb-0.2.9]# ln -s /usr/include/X11 /usr/X11R6/include/X11
```

- Không thấy báo lỗi gì thì tiến hành tiếp

- Gõ lệnh "make install"

```
[root@localhost xvnkb-0.2.9a]# make install
```

```
make[1]: Entering directory `/SOURCES/Linux/xvnkb-0.2.9a/tools'
```

```
make[1]: Nothing to be done for `all'.
```

```
make[1]: Leaving directory `/SOURCES/Linux/xvnkb-0.2.9a/tools'
```

Copy xvnkb => /usr/local/bin ... ok
Copy xvnkb.so.0.2.9a => /usr/local/lib ... ok
Copy xvnkb_localeconf.sh => /usr/local/bin ... ok
Initialize xvnkb core ... done
You can use xvnkb now! If you are using X, please restart your Window Manager.
It will load xvnkb core control automatically for you and affect to all applications. Right now, xvnkb core control can affect to new starting applications only. Run "xvnkb" to control status.

- Cài đặt đã thành công

Bây giờ kiểm tra xem sau khi giải nén, tar có tạo thêm thư mục nào để chứa tập tin .ttf hay không, nếu có ta phải chuyển vào thư mục chứa những tập tin .ttf (ví dụ /usr/share/fonts/unicode).

Dùng lệnh “ttmkfdir > fonts.scale” và “mkfontdir” để tạo danh sách chứa phông (nhập hai lệnh liên tiếp hoặc gộp chung thành một lệnh “ttmkfdir > fonts.scale && mkfontdir”). Cuối cùng phải báo cho dịch vụ quản lý phông biết vị trí của những phông Unicode

“chkfontpath - add /usr/share/fonts/utf8” và khởi động lại dịch vụ “/etc/rc.d/init.d/xfs restart”.

Tạo fonts unicode

```
/usr/local/bin/xvnkb_localeconf.sh en_US.UTF-8
```

```
/usr/local/bin/xvnkb_localeconf.sh vi_VN.UTF-8
```

Thêm nữa là ta hãy vào /etc/profile và thêm vào cuối cùng dòng sau :

```
export LANG=en_US.UTF-8
```

Khởi động lại máy tính bằng lệnh

```
#reboot
```

Kiểm tra lại tại dòng lệnh chạy xvnkb:

```
#xvnkb
```

Câu hỏi ôn tập chương

1. Hãy nêu khái niệm và các đặc tính của gói phần mềm .RPM?
2. Trình bày các bước cài đặt gói phần mềm rpm và nâng cấp phần mềm?
3. Nêu các bước cài đặt các gói phần mềm dạng TAR?

Chương 5: Các ứng dụng phần mềm phổ biến trong Linux

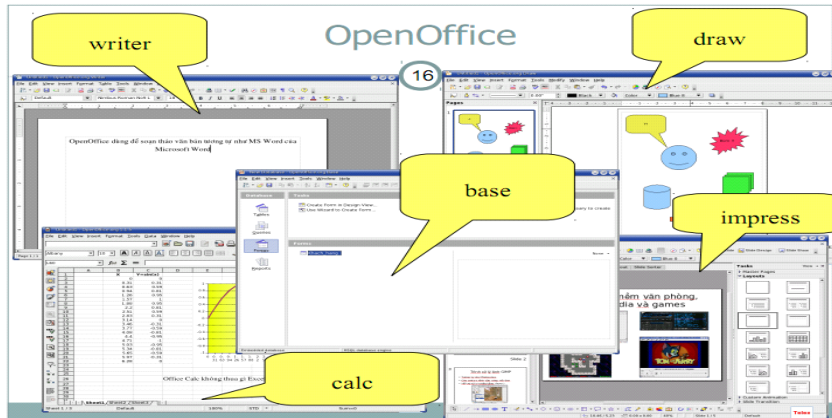
Mục tiêu:

- Biết một số loại phần mềm ứng dụng phổ biến trên nền Linux;
- Cài đặt được một số phần mềm ứng dụng phổ biến trên nền Linux;
- Có thái độ nghiêm túc, chịu khó tìm tòi, học hỏi cách sử dụng các phần mềm phổ biến trên Linux.

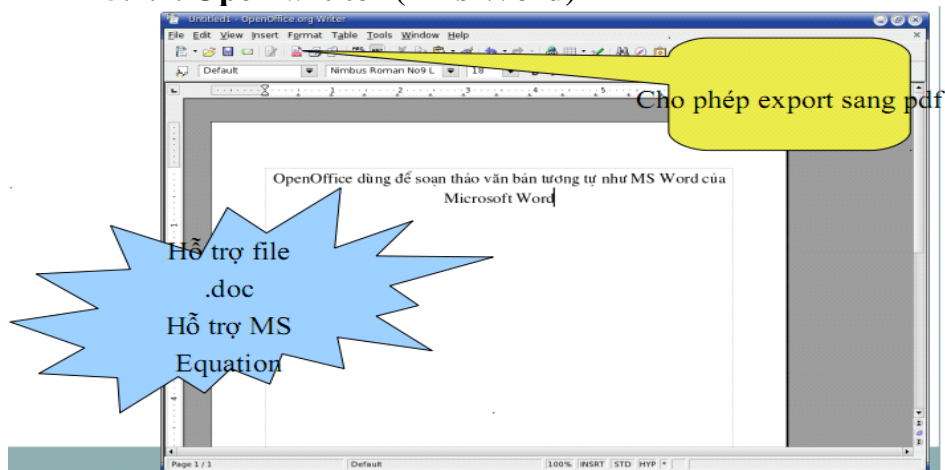
5.1. Bộ phần mềm văn phòng OpenOffice

Là 1 bộ phần mềm văn phòng đa ngôn ngữ, đa nền và là phần mềm nguồn mở. Tương thích với hầu hết các phần mềm văn phòng khác (ví dụ: Ms Office)

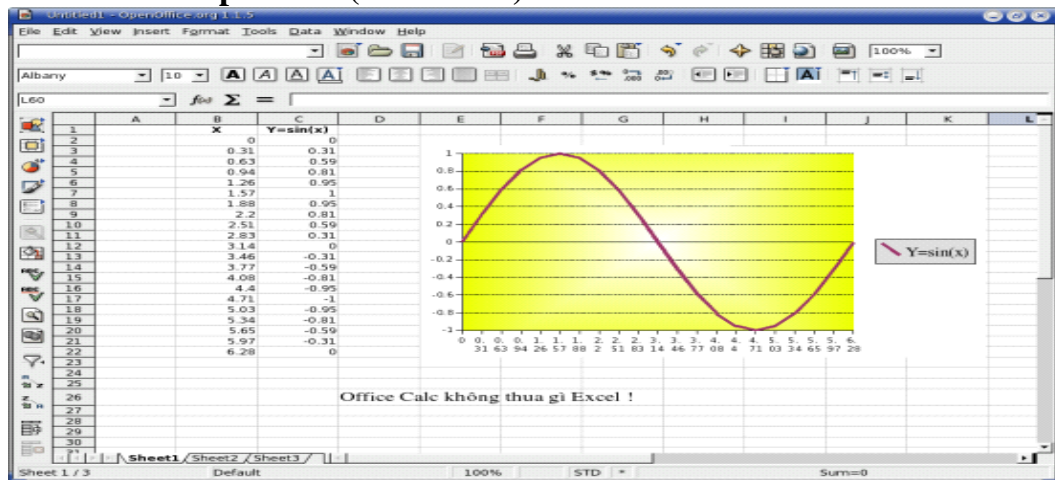
5.1.1. Hỗ trợ unicode



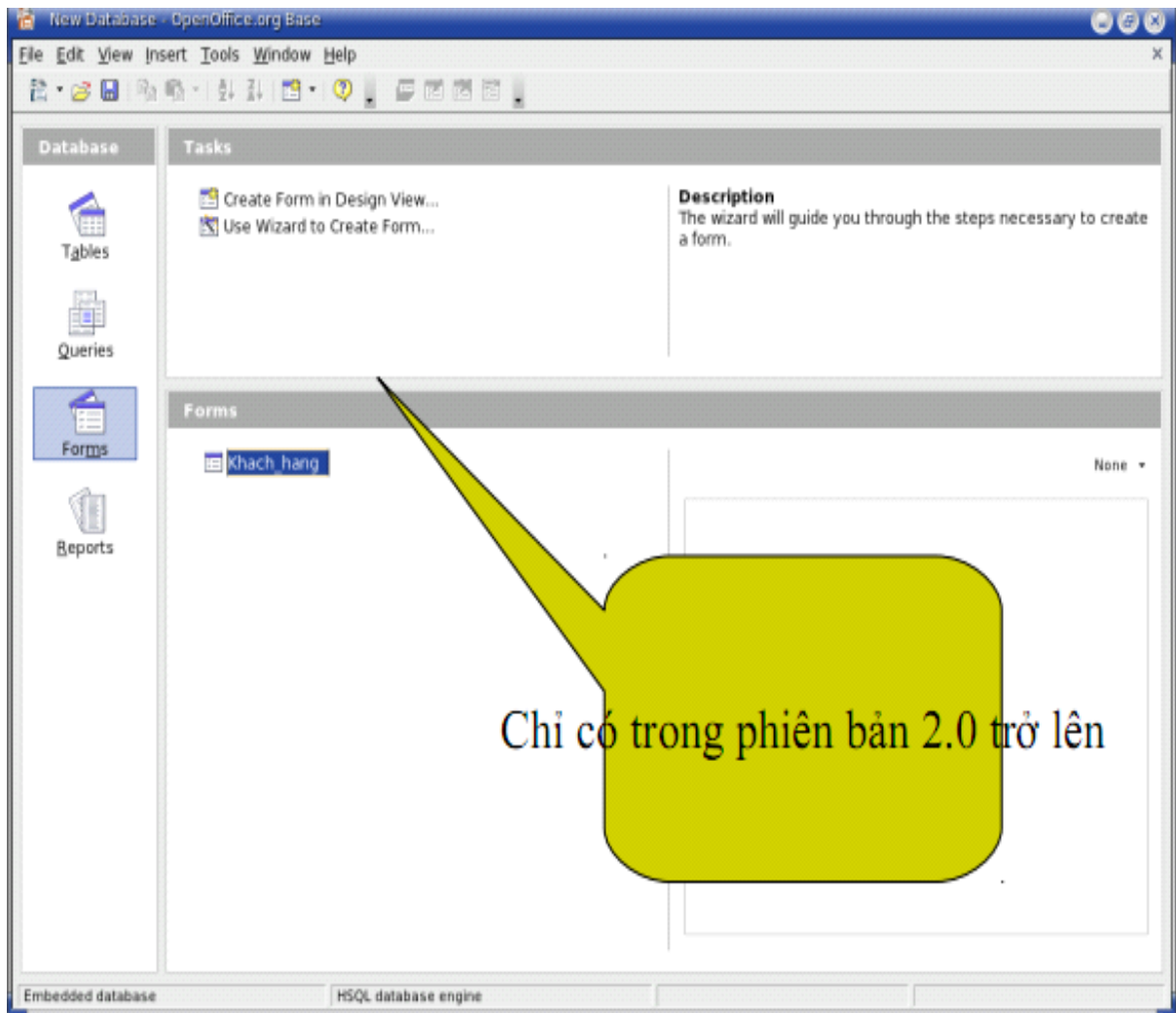
5.1.2. Open writer (~Ms Word)



5.1.3. Open calc (~Ms Excel)



5.1.4. Open base (~Ms Access)



5.2. Các phần mềm Internet

5.2.1. Web Hosting

Nếu như các tập tin và thư mục được lưu trữ trong ổ cứng của máy tính để Quý khách có thể truy cập chúng bất cứ lúc nào thì một trang web cũng giống như một tập tin, một văn bản tài liệu, tất cả các trang web cũng cần có một máy tính để lưu trữ chúng. Tuy nhiên, sự khác biệt giữa website và các tập tin thông thường đó là mục đích để được công chúng nhìn thấy. Để kích hoạt tính năng này, máy tính mà chúng ta sử dụng để lưu trữ phải có một đặc điểm kỹ thuật cao hơn nhiều so với máy tính thường và phải được kết nối với Internet thông qua một liên kết rất mạnh mẽ. Và máy tính mà trên đó các trang web được lưu trữ người ta gọi là Máy chủ (Server). Đồng thời, không gian để lưu trữ các trang web đó sẵn sàng cho nhiều người sử dụng được gọi là web hosting.

Như vậy, có thể hiểu *"Web Hosting là nơi lưu trữ tất cả các trang Web, các thông tin, tư liệu, hình ảnh của Website trên một máy chủ Internet, Web Hosting đồng thời cũng là nơi diễn ra tất cả các hoạt động giao dịch, trao đổi thông tin giữa Website với người sử dụng Internet và hỗ trợ các phần mềm Internet hoạt động"*.

5.2.2. Gửi Và Nhận Thư từ Webmail

Bước 1: Đăng nhập vào trang <https://cpanel.tenten.vn/> để thực hiện tạo một tài khoản Email

Bước 2: Click vào "Thêm Email" như hình ở dưới

Bước 3: sau khi tạo xong tài khoản Email thì click vào mục “[Web mail](#)” Sẽ hiện lên thêm giao diện như sau

Hãy đăng nhập bằng User và Pass mà đã tạo. Sau khi đăng nhập, muốn kiểm tra mail thì hãy click vào “INBOX”

Bước 4: Muốn gửi mail cho một địa chỉ nào đó thì hãy click vào “Compose”

From	Date	Subject
Microsoft Outlook	May 31, 2012	Microsoft Outlook Test Message
support@hotromaytin.com	May 30, 2012	Test Mail

Bước 5: Thực hiện điền các thông tin cần thiết, soạn thảo email và nhấn Send!

From	Date	Subject
Microsoft Outlook	May 31, 2012	Microsoft Outlook Test Message
support@hotromaytin.com	May 30, 2012	Test Mail

5.3. Các phần mềm Multimedia

5.3.1. Một số phần mềm multimedia

Mplayer: (Movie Player cho Linux) Cũng rất phổ biến để chơi nhạc là Amarok, một ứng dụng KDE rằng nước này đang dần được phổ biến, và MPlayer, mà cũng có thể chơi phim.

- **totem:** đồng cả âm thanh và các tập tin video, âm thanh đĩa CD, VCD và DVD.

- **realplay:** từ RealNetworks.

- **hxplay:** một giải pháp thay thế thực. Hỗ trợ RealAudio và RealVideo, mp3, mp4 audio, Flash, wav và nhiều hơn nữa.

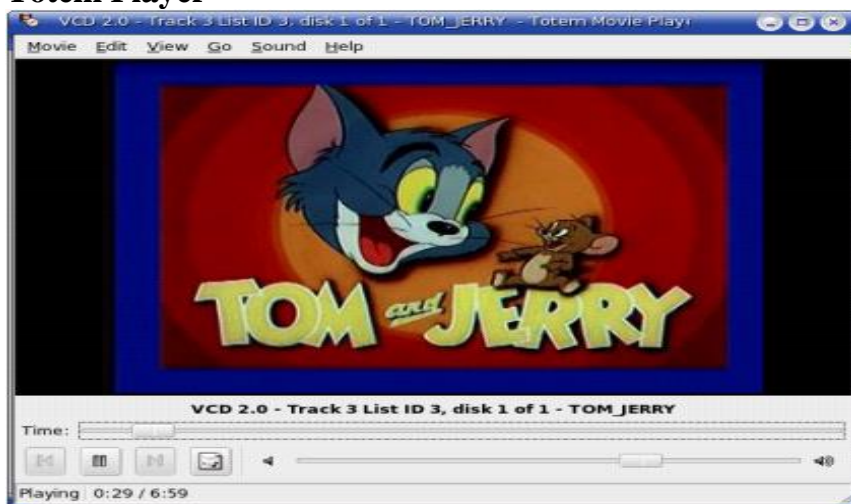
- **kaffeine:** máy nghe nhạc phương tiện truyền thông cho KDE3.

5.3.2. Hình ảnh giao diện của một số phần mềm multimedia

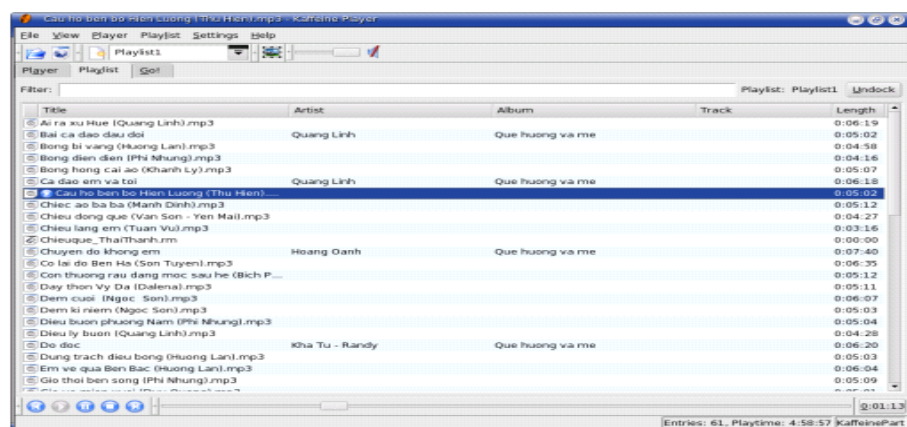
XMMS Player



Totem Player

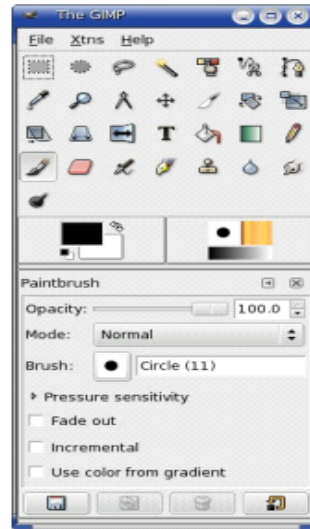


Kaffeine



5.3.3. Trình xử lý ảnh The GIMP

Tương tự như Photoshop cho phép chỉnh sửa, phục hồi ảnh hỗ trợ xử lý nhiều tầng (layer)



5.4. Các phần mềm hệ thống

Ngoài những phần mềm có sẵn ta có thể cài đặt các phần mềm hệ thống mã nguồn mở mà Linux hỗ trợ hiện nay như Simple backup, remove password...

5.4.1. Shell thiết lập các tập tin

Khi nhập lệnh **ls-al** để có được một danh sách dài của tất cả các tập tin, bao gồm cả những người bắt đầu với một dấu chấm, trong thư mục nhà của bạn, bạn sẽ thấy một hoặc nhiều tập tin bắt đầu với một. và kết thúc trong *rc*. Đối với trường hợp của **bash**, đây là *bashrc*. Đây là đối tác của các *bashrc* cấu hình hệ thống tập tin / etc /.

Khi đăng nhập vào một trình bao tương tác đăng nhập, **đăng nhập** sẽ làm chứng thực, thiết lập môi trường và bắt đầu trình bao của bạn. Trong trường hợp của **bash**, bước tiếp theo là đọc hồ sơ cá nhân nói chung từ / etc, nếu tập tin tồn tại **bash** sau đó sẽ cho ~ /. *Bash_profile*, ~ /. *Bash_login* và ~ / hồ sơ, theo thứ tự đó, và đọc và thực hiện lệnh từ một trong những đầu tiên tồn tại và có thể đọc được. Nếu không tồn tại, / etc / *bashrc* được áp dụng.

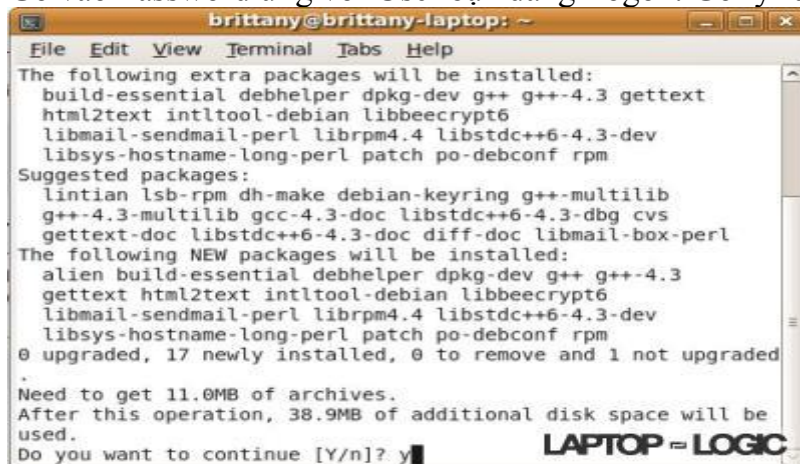
Khi một thoát khỏi trình bao đăng nhập, **bash** đọc và thực hiện lệnh từ tập tin ~ /. *Bash_logout*, nếu nó tồn tại.

5.4.2. Cài đặt các file .rpm

Ta nên sử dụng gói Alien để chuyển từ .rpm sang .deb cho dễ cài đặt - mở Terminal lên, gõ vào `sudo apt-get install alien` để download và cài đặt gói Alien thông qua tiện ích quản lý gói APT.



Gõ vào Password ứng với User bạn đang Login. Gõ 'y' để đồng ý cài đặt gói Alien



```
brittany@brittany-laptop: ~  
File Edit View Terminal Tabs Help  
The following extra packages will be installed:  
  build-essential debhelper dpkg-dev g++ g++-4.3 gettext  
  html2text intltool-debian libbeecrypt6  
  libmail-sendmail-perl librpm4.4 libstdc++6-4.3-dev  
  libsys-hostname-long-perl patch po-debconf rpm  
Suggested packages:  
  lintian lsb-rpm dh-make debian-keyring g++-multilib  
  g++-4.3-multilib gcc-4.3-doc libstdc++6-4.3-dbg cvs  
  gettext-doc libstdc++6-4.3-doc diff-doc libmail-box-perl  
The following NEW packages will be installed:  
  alien build-essential debhelper dpkg-dev g++ g++-4.3  
  gettext html2text intltool-debian libbeecrypt6  
  libmail-sendmail-perl librpm4.4 libstdc++6-4.3-dev  
  libsys-hostname-long-perl patch po-debconf rpm  
0 upgraded, 17 newly installed, 0 to remove and 1 not upgraded  
.  
Need to get 11.0MB of archives.  
After this operation, 38.9MB of additional disk space will be  
used.  
Do you want to continue [Y/n]? y
```

- Sau khi cài xong Alien, move file .rpm tới [Desktop](#) rồi mở Terminal, gõ cd Desktop



```
brittany@brittany-laptop: ~/Desktop  
File Edit View Terminal Tabs Help  
brittany@brittany-laptop:~$ cd Desktop  
brittany@brittany-laptop:~/Desktop$
```

- Bây giờ, gõ sudo alien -k filename.rpm để convert từ file .rpm -> .deb. Sau đó cài file .deb như trên.



```
brittany@brittany-laptop: ~/Desktop  
File Edit View Terminal Tabs Help  
brittany@brittany-laptop:~$ cd Desktop  
brittany@brittany-laptop:~/Desktop$ sudo alien -k flash.rpm  
File "flash.rpm" not found.  
brittany@brittany-laptop:~/Desktop$ sudo alien -k flash.rpm  
Warning: Skipping conversion of scripts in package flash-plugin: postinst prepm  
Warning: Use the --scripts parameter to include the scripts.  
flash-plugin 10.0.15.3-release-1 i386.deb generated  
brittany@brittany-laptop:~/Desktop$
```

Câu hỏi ôn tập chương

1. Nêu ứng dụng của bộ phần mềm OpenOffice?
2. Thế nào được gọi là Web Hosting? Nêu các bước gửi và nhận thư từ Webmail?
3. Hãy nêu một số phần mềm multimedia thông dụng trên hệ điều hành Linux?
4. Trình bày các phần mềm hệ thống của Linux? Cho ví dụ?

Chương 6: Giao diện dòng lệnh

Mục tiêu:

- Hiểu khái niệm Shell;
- Hiểu một số lệnh cơ bản trong Linux;
- Sử dụng một số lệnh cơ bản trong Linux;
- Sử dụng thành thạo một số phím tắt;
- Cấu hình và lập trình Shell;
- Có thái độ nghiêm túc, chịu khó tìm tòi, học hỏi.

6.1. Khái niệm Shell

6.1.1 Khái niệm

Shell là chương trình luôn được thực thi khi chúng ta đăng nhập hệ thống. Nó là chương trình cho phép chúng ta tương tác với hệ thống. Hiện tại có nhiều shell có sẵn trong hệ thống.

Shell là chương trình nằm giữa người sử dụng và kernel, thông thường nó là một bộ biên dịch dòng lệnh từ người sử dụng ở các thiết bị cuối (cũng có thể từ file) và thực hiện chúng.

Không những thế, trong UNIX shell còn là một ngôn ngữ lập trình thực sự với đầy đủ các cú pháp cần thiết như câu lệnh điều kiện, vòng lặp, các chương trình con, thủ tục...

Shell cung cấp cho người dùng một tập lệnh để người dùng thao tác với hệ thống. Khi người dùng thực hiện lệnh shell, shell sẽ dịch chúng thành các lời gọi hệ thống và chuyển cho kernel hệ điều hành xử lý. Shell cũng là một phần trong các ứng dụng mà kernel quản lý. Kernel chịu trách nhiệm cấp phát tài nguyên duy trì các tiến trình shell. Linux là hệ thống đa người dùng, khi mỗi người dùng đăng nhập hệ thống, họ sẽ nhận được một bản copy của shell để thao tác với hệ thống.

6.1.2. Phân loại

Có ba loại shell:

- Bourne shell của Steven Bourne đơn giản và hiệu quả. Nó là mặc định trong đa số các hệ UNIX (hoặc có thể gọi bởi sh).
- C shell của Bill Joy ở trường đại học Berkeley giống như Bourne shell nhưng bổ sung thêm các đặc điểm như bí danh, history vvv. Nó có thể gọi bởi csh.
- Korn shell của David F. Korn kết hợp Bourne shell và C shell nhưng bổ sung thêm các đặc điểm riêng. Nó có thể gọi bởi ksh.

6.2. Một số lệnh cơ bản

Khi mở một shell, bạn cần đến tài khoản kích hoạt vào thư mục chủ (thông thường nằm trong /home/tên_người_dùng).

Mỗi lệnh đều có nhiều tùy chọn riêng của mình. Để xem các tùy chọn cho một lệnh cụ thể, đơn giản bạn chỉ cần gõ "**man <command>**" (trong đó: *man* là từ khóa; *command* là tên lệnh). Một điểm quan trọng cần chú ý là trong các hệ điều hành Linux, câu lệnh có phân biệt chữ hoa, chữ thường. "**A**" sẽ được hệ điều hành hiểu là khác so với "**a**".

6.2.1. Lệnh liên quan đến hệ thống

- * exit: thoát khỏi cửa sổ dòng lệnh.
- * logout: tương tự exit.
- * reboot: khởi động lại hệ thống.
- * halt: tắt máy.
- * startx: khởi động chế độ xwindows từ cửa sổ terminal.

- * mount: gắn hệ thống tập tin từ một thiết bị lưu trữ vào cây thư mục chính.
- ví dụ: mount /mnt/floppy: lắp thêm ổ mềm. ; mount /mnt/cdrom: lắp ổ CD-

ROM

- * umount: ngược với lệnh mount.

6.2.2. Lệnh thao tác trên tập tin

- * ls: lấy danh sách tất cả các file và thư mục trong thư mục hiện hành.
 - * pwd: xuất đường dẫn của thư mục làm việc.
 - * cd: thay đổi thư mục làm việc đến một thư mục mới.
 - Ví dụ : cd /home/nvtienvn : thay đổi thư mục tới /home/nvtienvn
 - * mkdir: tạo thư mục mới.
 - Ví dụ : mkdir /home/nvtienvn/public_html : tạo thư mục public_html trong /home/nvtienvn
 - * rmdir: xóa thư mục rỗng.
 - Ví dụ : rmdir /home/nvtienvn/public_html/backup_empty
 - * cp: copy một hay nhiều tập tin đến thư mục mới.
 - Ví dụ : cp /home/nvtienvn/alo.mp3 /home/nvtienvn/music/alo.mp3 : copy tập tin alo.mp3 từ /home/nvtienvn tới /home/nvtienvn/music
 - * mv: đổi tên hay di chuyển tập tin, thư mục.
 - Ví dụ : mv /home/nvtienvn/music/alo.mp3 /home/nvtienvn/music/bai-hat-zbc.mp3 : đổi tên alo.mp3 sang bai-hat-zbc.mp3
 - * rm: xóa tập tin.
 - Ví dụ : rm /home/nvtienvn/alo.mp3 : xóa tập tin alo.mp3
 - * wc: đếm số dòng, số kí tự... trong tập tin.
 - * touch: tạo một tập tin.
 - * cat: xem nội dung tập tin.
 - * vi: khởi động trình soạn thảo văn bản vi.
 - * df: kiểm tra dung lượng đĩa.
 - * du: xem dung lượng đĩa đã dùng cho một số tập tin nhất định
- ### 6.2.3. Lệnh khi làm việc trên terminal
- * clear: xóa trắng cửa sổ dòng lệnh.
 - * date: xem ngày, giờ hệ thống.
 - * cal: xem lịch hệ thống.

Lệnh quản lí hệ thống

- * rpm: kiểm tra gói đã cài đặt hay chưa, hoặc cài đặt một gói, hoặc sử dụng để gỡ bỏ một gói.
- * ps: kiểm tra hệ thống tiến trình đang chạy.
- * kill: dừng tiến trình khi tiến trình bị treo. Chỉ có người dùng super-user mới có thể dừng tất cả các tiến trình còn người dùng bình thường chỉ có thể dừng tiến trình mà mình tạo ra.
- * top: hiển thị sự hoạt động của các tiến trình, đặc biệt là thông tin về tài nguyên hệ thống và việc sử dụng các tài nguyên đó của từng tiến trình.
- * pstree: hiển thị tất cả các tiến trình dưới dạng cây.
- * sleep: cho hệ thống ngừng hoạt động trong một khoảng thời gian.
- * useradd: tạo một người dùng mới.
- * groupadd: tạo một nhóm người dùng mới.
- * passwd: thay đổi password cho người dùng.
- * userdel: xóa người dùng đã tạo.
- * groupdel: xóa nhóm người dùng đã tạo.

- * `gpasswd`: thay đổi password của một nhóm người dùng.
- * `su`: cho phép đăng nhập với tư cách người dùng khác.
- * `groups`: hiển thị nhóm của user hiện tại.
- * `who`: cho biết ai đang đăng nhập hệ thống.
- * `w`: tương tự như lệnh `who`.
- * `man`: xem hướng dẫn về dòng lệnh như cú pháp, các tham số...

Để thoát hay đóng shell, gõ `exit` hoặc `logout`.

6.3. Sử dụng phím tắt

6.3.1. Trong terminal

- **<Ctrl> + L**: xoá toàn bộ màn hình, giống lệnh **clear**
- **<Ctrl> + D**: exit session, giống lệnh **exit**
- **<Ctrl> + R**: tìm một lệnh đã chạy trước đây, nhấn **<Ctrl> + R** sau đó bắt đầu gõ một phần của câu lệnh, hệ thống sẽ tự hoàn tất phần còn lại dựa trên các câu lệnh đã được thực hiện trước đó
- **<Tab>**: tự động hoàn tất câu lệnh
- **<Shift> + <Insert>**: dán (paste) nội dung đã copy vào terminal
- **<Shift> + PageUp**: cuộn màn hình lên trên một trang
- **<Ctrl> + <Alt> + F2 (<Alt> + F2)** nếu đang ở chế độ console): chuyển sang virtual terminal thứ 2, tương tự với F3, F4 ...

6.3.2. Trong GNOME

- **<Ctrl> + <Alt> + D**: hiển thị desktop, giống **<Windows> + D** trong Windows
- **<Ctrl> + <Alt> + <Left/Right>**: chuyển sang workspace trước/kế tiếp
- **<Ctrl> + W**: đóng cửa sổ hiện thời
- **<Ctrl> + Q**: thoát khỏi chương trình hiện thời
- **<Alt> + F1**: Hiển thị main menu
- **<Alt> + F2**: Hiển thị hộp thoại chạy dòng lệnh, giống **<Windows> + R** trong Windows
- **<Alt> + F5**: Bỏ phóng to cửa sổ hiện thời
- **<Alt> + F9**: Thu nhỏ **<minimize>** cửa sổ hiện thời
- **<Alt> + F10**: Phóng to **<maximize>** cửa sổ hiện thời

6.3.3. Trong OpenOffice

- **<Ctrl> + <Shift> + B**: chuyển font sang dạng subscript, giống **<Ctrl> + '+'** trong MS Word
- **<Ctrl> + <Shift> + P**: chuyển font sang dạng supperscript, giống **<Ctrl> + <Shift> + '+'** trong MS Word
- **<Ctrl> + 1**: Single line spacing
- **<Ctrl> + 2**: Double line spacing
- **<Ctrl> + 5**: 1.5 line spacing

6.3.4. Trong vi (vim)

Các phím sau đây được sử dụng trong chế độ nhập lệnh (không phải chế độ nhập văn bản). Để viết về vi thì cần một bài riêng, nên ở đây cheva chỉ nêu ra một số phím/lệnh chính, chú ý các phím/lệnh sau đây có phân biệt chữ hoa và chữ thường:

- **G**: tới cuối file
- **#G**: tới dòng thứ # (ví dụ 10G, 100G)
- **H**: tới đầu trang
- **dd**: xóa dòng hiện thời

- **yy**: copy dòng hiện thời
- **p**: dán xuống dòng dưới dòng hiện thời
- **P**: dán vào trước vị trí con trỏ
- **/**: bắt đầu tìm kiếm
- **^**: tới đầu dòng
- **\$**: tới cuối dòng
- **%**: tới dấu đóng (mở) ngoặc tương ứng
- **%s/old_text/new_text/g**: thay thế tất cả các old_text bằng new_text
- Ở chế độ nhập văn bản, gõ **<Ctrl> + P** để sử dụng auto text completion

6.4. Cú pháp lệnh

6.4.1. Tạo thư mục

Lệnh **mkdir** tạo một thư mục, cú pháp: **mkdir [tùy-chọn] <thư-mục>**

Lệnh này cho phép tạo một thư mục mới nếu thư mục đó chưa thực sự tồn tại. Để tạo một thư mục, cần đặc tả tên và vị trí của nó trên hệ thống file (vị trí mặc định là thư mục hiện thời). Nếu thư mục đã tồn tại, hệ thống sẽ thông báo cho biết.

Các tùy chọn:

- **m**, **--mode=Mod** : thiết lập quyền truy nhập Mod như trong lệnh **chmod** nhưng không cho quyền **rwrxrwx**.
- **p**, **--parents** : tạo các thư mục cần thiết mà không thông báo lỗi khi nó đã tồn tại.
- **-v**, **--verbose** : hiển thị các thông báo cho mỗi thư mục được tạo.
- **-h**, **--help** : đưa ra trang trợ giúp và thoát.

Nếu muốn tạo thư mục có khoảng cách giữa các từ ta phải sử dụng dấu **"**. Nếu muốn tạo thư mục **My Documents** ta sử dụng lệnh: **mkdir "My Documents"**

Ví dụ: nếu muốn tạo thư mục **test** trong thư mục **home**, hãy gõ lệnh sau: **mkdir /home/test**

6.4.2. Xóa thư mục với lệnh **rmdir**

Lệnh **rmdir** được dùng để xóa bỏ một thư mục.

Cú pháp lệnh: **rmdir [tùy-chọn] <thư-mục>**

Có thể xóa bỏ bất kỳ thư mục nào nếu có quyền đó. Lưu ý rằng, thư mục chỉ bị xóa khi nó "rỗng", tức là không tồn tại file hay thư mục con nào trong đó.

Không có cách gì khôi phục lại các thư mục đã bị xóa, vì thế hãy suy nghĩ cẩn thận trước khi quyết định xóa một thư mục.

Các tùy chọn của lệnh:

- **-i**, **--ignore-fail-on-non-empty** : bỏ qua các lỗi nếu xóa một thư mục không rỗng.
- **p**, **--parents** : xóa bỏ một thư mục, sau đó lần lượt xóa bỏ tiếp các thư mục có trên đường dẫn chứa thư mục vừa xóa. Ví dụ, dòng lệnh **rmdir -p /a/b/c** sẽ tương đương với ba dòng lệnh **rmdir /a/b/c**, **rmdir /a/b**, **rmdir /a** (với điều kiện các thư mục là rỗng).
- **-v**, **--verbose** : đưa ra thông báo khi xóa một thư mục.
- **-h**, **--help** : hiển thị trang trợ giúp và thoát.

Ví dụ:

```
# rmdir -p /test/test1/test2
rmdir: /: No such file or directory
```

Dòng lệnh trên sẽ lần lượt xóa ba thư mục **test2**, **test1**, **test** và hiển thị thông báo trên màn hình kết quả của lệnh.

6.4.3. Xem đường dẫn thư mục hiện thời với lệnh **pwd**

Cú pháp lệnh: **pwd**

Lệnh này cho biết hiện người dùng đang ở trong thư mục nào và hiện ra theo dạng một đường dẫn tuyệt đối.

Ví dụ: gõ lệnh `pwd` tại dấu nhắc lệnh sau khi người dùng `duonglk` vừa đăng nhập thì màn hình hiển thị như sau:

```
# pwd /home/duonglk
```

6.4.4. Lệnh đổi tên thư mục với lệnh `mv`

Cú pháp lệnh: `mv <tên-cũ> <tên-mới>`

Lệnh này cho phép đổi tên một thư mục từ tên-cũ thành tên-mới.

Ví dụ: `# mv Tongket thongke` sẽ đổi tên thư mục *Tongket* thành *thongke*.

Nếu sử dụng lệnh `mv` để đổi tên một thư mục với một cái tên đã được đặt cho một file thì lệnh sẽ gặp lỗi. Nếu tên mới trùng với tên một thư mục đang tồn tại thì nội dung của thư mục được đổi tên sẽ ghi đè lên nội dung của thư mục trùng tên.

6.4.5. Tạo file với lệnh `touch`

Lệnh `touch` có nhiều chức năng, trong đó một chức năng là giúp tạo file mới trên hệ thống: `touch` rất hữu ích cho việc tổ chức một tập hợp các file mới.

Cú pháp lệnh: `touch <file>`

Thực chất lệnh này có tác dụng dùng để cập nhật thời gian truy nhập và sửa chữa lần cuối của một file. Vì lý do này, các file được tạo bằng lệnh `touch` đều được sắp xếp theo thời gian sửa đổi. Nếu sử dụng lệnh `touch` đối với một file chưa tồn tại, chương trình sẽ tạo ra file đó. Sử dụng bất kỳ trình soạn thảo nào để soạn thảo file mới.

Ví dụ: dùng lệnh `touch` để tạo file `newfile`: `# touch newfile`

6.4.6. Tạo file với lệnh `cat`

Lệnh `cat` tuy đơn giản nhưng rất hữu dụng trong Linux. Chúng ta có thể sử dụng lệnh này để lấy thông tin từ đầu vào (bàn phím...) rồi kết xuất ra file hoặc các nguồn khác, hay để xem nội dung của một file ... Phần này trình bày tác dụng của lệnh `cat` đối với việc tạo file.

Cú pháp lệnh: `cat > filename`

Theo ngầm định, lệnh này cho phép lấy thông tin đầu vào từ bàn phím rồi xuất ra màn hình. Soạn thảo nội dung của một file bằng lệnh `cat` tức là đã đổi hướng đầu ra của lệnh từ màn hình vào một file.

6.5. Cấu hình Shell

Khi người dùng đăng nhập vào hệ thống, họ sẽ làm việc trong môi trường do Linux định nghĩa sẵn. Môi trường Linux chứa các thiết lập và dữ liệu có tính năng kiểm tra phiên làm việc của bạn trong suốt thời gian đăng nhập. Tuy nhiên, bạn cũng có thể thay đổi những thiết lập này theo ý riêng của mình. Môi trường phiên làm việc gồm hai thành phần:

- Thành phần thứ nhất gọi là môi trường terminal để điều khiển terminal (chính là màn hình và bàn phím) của bạn.

- Thành phần thứ hai gọi là môi trường shell để điều khiển nhiều khía cạnh khác nhau củashell, cùng với mọi chương trình bạn thực hiện.

6.5.1. Thiết lập môi trường terminal

Thực ra phiên đăng nhập của bạn bao gồm hai chương trình riêng biệt nhưng chạy cùng lúc với nhau, tạo cho bạn cảm giác rằng máy đang phục vụ cho riêng mình. Mặc dù shell là chương trình nhận lệnh và thi hành, song trước khi shell nhận được lệnh, tất cả những gì mà bạn gõ vào đều phải đi qua một trình điều khiển thiết bị gọi là device driver. Driver kiểm soát terminal, nhận những kí tự bạn gõ vào rồi sau đó quyết định xem xử lý như thế nào trước khi giao cho shell thông dịch. Tương tự như thế, mỗi kí tự phát sinh từ shell phải đi ngang driver thiết bị trước khi đến terminal. Khi làm việc trên hệ thống Linux, chương trình xem tất cả các thiết bị nối kết với hệ thống đều như nhau, một số phím quan trọng:

Bảng 6.5.1.1 Mô tả các phím tắt

Phím	Mô tả
Interrupt	Đình chỉ thực hiện một chương trình. Linux dùng tổ hợp phím <Ctrl +C>
Erase	Xóa ký tự cuối cùng trong vùng đệm, đó là phím <backspace>
Kill	Xóa toàn bộ những gì trong vùng đệm trước khi chuyển sang shell hoặc chương trình ứng dụng. Thông thường đó là phím <@>. Không giống như trường hợp bấm phím dừng, sẽ không thấy hiện ra dấu nhắc shell khi bấm phím kill, bởi vì driver chờ gõ tiếp.
End-of-line	Báo cho driver biết đã gõ xong các ký tự và muốn được thông dịch và chuyển sang shell hoặc chương trình, linux sử dụng phím Enter
End-of-file	Báo cho shell thoát ra và hiển thị dấu nhắc đăng nhập, ký tự cuối tập tin là Ctrl+d

6.5.2. Thiết lập môi trường Shell

Khi đăng nhập vào hệ thống, người dùng sẽ làm việc trong môi trường shell của mình do Linux định nghĩa trước. Trong môi trường shell gồm nhiều biến. Khai báo mỗi biến có dạng <BIẾN=giá trị>, ý nghĩa của một biến như thế nào là tùy bạn chỉ định. Tuy nhiên, có một số biến đã được định nghĩa sẵn. Ví dụ như biến: TERM, PATH. Bảng sau đây liệt kê những biến môi trường phổ biến trong shell Bourne:

Bảng 6.5.2.1 liệt kê biến môi trường

Biến	Mô tả
HOME=/home/đăng=nhập	HOME lập home directory của cá nhân, đăng=nhập là ID đăng nhập. Ví dụ nếu ID đăng nhập của ta là jack thì HOME sẽ là /home/jack
LOGNAME=đăng=nhập	Máy sẽ tự động lập LOGNAME bằng ID đăng nhập của bạn
PATH=đường=dẫn	Tùy chọn đường=dẫn trỏ đến danh sách các thư mục mà shell sẽ duyệt qua để tìm lệnh. Ví dụ: PATH=/user:/bin:/user/local/bin
PS1=dấu=nhắc	PS1 là dấu nhắc shell đầu tiên để yêu cầu xác định hình dáng của dấu nhắc riêng theo ý nghĩa của mình. Nếu không có thay đổi gì dấu nhắc mặc định sẽ là dấu S
PWD=thư=mục	Xác định vị trí của bạn trong hệ thống tập tin
SHELL=shell	SHELL xác định shell mà bạn đang sử dụng
dùng TERM=loại=terminal	Kiểu terminal bạn dùng

Lưu ý: nếu muốn xác lập những biến môi trường, bạn hãy xác định trong tập tin .bash_profile (nếu chạy shell bash), trong tập tin .login (nếu chạy shell C) và trong tập tin .profile (nếu chạy shell Bourne).

6.6. Lập trình Shell

6.6.1. Lệnh echo

Chức năng: Hiện một dòng văn bản ra màn hình

Cú pháp: echo [-n] <dòng văn bản>

Trong đó [-n] chỉ định sẽ không xuống dòng sau khi đưa ra màn hình dòng văn bản.

Ví dụ: Echo Khoa Điện tử - Tin học

Trên màn hình sẽ xuất hiện dòng chữ

Khoa Điện tử - Tin học

6.6.2. Lệnh read

Chức năng: đọc từ bàn phím một xâu ký tự và ghi xâu đó vào một biến

Cú pháp: Read <tên biến>

Ví dụ: Echo -n Ho va ten:

Read hoten

6.6.3. Sử dụng biến

Trong lập trình Shell không cần phải khai báo biến trước khi dùng mà biến sẽ được tự động tạo và khai báo khi lần đầu tiên tên biến xuất hiện. Mặc định là tất cả các biến đều được khởi tạo và chứa giá trị kiểu chuỗi. Shell và một vài lệnh tiện ích sẽ tự động chuyển thành chuỗi, thành số để thực hiện phép tính khi nó yêu cầu.

Bên trong các Script của Shell ta có thể lấy về nội dung của biến bằng cách dùng dấu \$ đặt trước tên biến. Để hiển thị nội dung biến có thể dùng lệnh echo. Khi gán nội dung cho biến ta không cần phải dùng ký tự \$.

Ví dụ: tạo file vd1.sh chứa các lệnh sau:

```
#!/bin/sh
Xinchao=hello
Echo $xinchao
Xinchao="I am here"
Xincho=12+1
Echo $xinchao
```

Khi chạy đoạn lệnh này, trên màn hình sẽ xuất hiện như sau:

```
[root@localhost root]# ! /bin/sh
sh-2.05b# xinchao=hello
sh-2.05b# echo $xinchao
hello
sh-2.05b# xinchao="I am here"
sh-2.05b# echo $xinchao
I am here
sh-2.05b# xinchao=12+1
sh-2.05b# echo $xinchao
12+1
sh-2.05b#
```

Ta có thể sử dụng lệnh read để đọc nhập dữ liệu cho người dùng đưa vào và giữ lại trong biến để sử dụng.

Ví dụ:

```
Read -n yourname
Minh Khai
Echo Hello $yourname
Hello Minh Khai
```

+ Dấu bọc chuỗi (quoting)

Thông thường tham số dòng lệnh thường cách nhau bằng một khoảng trắng.

Khoảng trắng có thể là ký tự spacebar, tab hoặc ký tự xuống dòng.

Trong trường hợp ta muốn tham số của mình chứa được cả khoảng trắng thì phải bọc chuỗi bằng dấu nháy đơn hoặc dấu nháy kép.

Dấu nháy kép được dùng trong trường hợp biến chuỗi của ta có khoảng trắng.

Tuy nhiên với dấu nháy kép, ký hiệu biến \$ vẫn có hiệu lực. Nội dung của biến sẽ

được thay thế trong chuỗi, dấu nháy đơn sẽ có hiệu lực mạnh hơn, nếu biến có ký tự \$

đặt trong chuỗi có dấu nhảy đơn, nó sẽ bị vô hiệu hóa thì có thể dùng dấu \ để hiển thị ký tự đặc biệt \$ trong chuỗi.

+ **Biến môi trường (environment variable)**

Khi trình Shell khởi động, nó cung cấp sẵn cho ta một số biến được khai báo và gán trị mặc định, chúng được gọi là các biến môi trường. Các biến này thường được viết hoa để phân biệt với biến do người dùng tự định nghĩa (thường là ký tự viết không hoa). Nội dung các biến này thường tùy vào thiết lập của hệ thống và người quản trị cho phép sử dụng.

Danh sách của các biến môi trường khá nhiều nhưng nhìn chung ta nên nhớ một số biến môi trường chủ yếu sau:

Bảng 6.6.3.1 Một số biến môi trường

Biến môi trường	Ý nghĩa
\$HOME	Chứa nội dung thư mục (Thư mục đầu tiên khi user đăng nhập)
\$PATH	Chứa danh sách các đường dẫn (phân cách bằng dấu hai chấm :). Linux thường tìm các lệnh cần thi hành trong biến \$PATH
\$PS1	Dấu nhắc (prompt) hiển thị trên dòng lệnh, thông thường là \$ cho user không phải root
\$PS2	Dấu nhắc thứ cấp báo cho người dùng nhập thêm thông tin trước khi lệnh thực hiện, thường là dấu >
\$IFS	Dấu phân cách các trường trong danh sách chuỗi. Biến này chứa danh sách các ký tự mà Shell dùng tách chuỗi (thường là tham số trên trong dòng lệnh). Ví dụ: \$IFS thường chứa ký tự Tab, ký tự trắng hoặc ký tự xuống hàng
\$0	Chứa tên chương trình gọi trên dòng lệnh
\$#	Tham số truyền trên dòng lệnh
\$\$	Mã tiến trình (process id) của Shell script khi thực thi bởi một số process id của tiến trình là duy nhất trên toàn hệ thống vào lúc script dùng con số này để tạo ra các tên file tạm.

Câu hỏi ôn tập chương

1. Trình bày khái niệm và phân loại Shell trong Redhat Linux?
2. Trình bày các lệnh cơ bản trên Linux? Nêu cụ thể ứng dụng của từng câu lệnh khi thao tác với Linux?
3. Trình bày các phím tắt trong cửa sổ GNOME và OpenOffice?
4. Nêu chức năng, cú pháp các lệnh tạo, xóa tập tin và thư mục?
5. Trình bày các bước lập trình Shell? Cho ví dụ minh họa?

Chương 7: Hệ thống tập tin

Mục tiêu:

- Hiểu về hệ thống tập tin của Linux;
- Sử dụng các công cụ tìm kiếm, di chuyển trong hệ thống tập tin;
- Quản lý và phân vùng ổ đĩa;
- Bảo trì hệ thống tập tin;
- Có thái độ nghiêm túc, chịu khó tìm tòi học hỏi.

7.1. Khái niệm hệ thống tập tin

7.1.1. Khái niệm

Người dùng đã từng làm việc với hệ điều hành DOS/Windows thì rất quen biết với các khái niệm: file (tập tin), thư mục, thư mục hiện thời ... Để đảm bảo tính hệ thống và thuận tiện cho người dùng chưa từng làm việc thành thạo với một hệ điều hành nào khác, chương này vẫn giới thiệu về các khái niệm này một cách sơ bộ.

Một đối tượng điển hình trong các hệ điều hành đó là file. File là một tập hợp dữ liệu có tổ chức được hệ điều hành quản lý theo yêu cầu của người dùng. Cách tổ chức dữ liệu trong file thuộc về chủ của nó là người đã tạo ra file. File có thể là một văn bản (trường hợp đặc biệt là chương trình nguồn trên C, PASCAL, shell script ...), một chương trình ngôn ngữ máy, một tập hợp dữ liệu ... Hệ điều hành tổ chức việc lưu trữ nội dung file trên các thiết bị nhớ lâu dài (chẳng hạn đĩa từ) và đảm bảo các thao tác lên file. Chính vì có hệ điều hành đảm bảo các chức năng liên quan đến file nên người dùng không cần biết file của mình lưu ở vùng nào trên đĩa từ, bằng cách nào đọc/ghi lên các vùng của đĩa từ mà vẫn thực hiện được yêu cầu tìm kiếm, xử lý lên các file.

Hệ điều hành quản lý file theo tên gọi của file (tên file) và một số thuộc tính liên quan đến file. Trước khi giới thiệu một số nội dung liên quan đến tên file và tên thư mục, chúng ta giới thiệu sơ bộ về khái niệm thư mục.

Để làm việc được với các file, hệ điều hành không chỉ quản lý nội dung file mà còn phải quản lý các thông tin liên quan đến các file. Thư mục (directory) là đối tượng được dùng để chứa thông tin về các file, hay nói theo một cách khác, thư mục chứa các file. Các thư mục cũng được hệ điều hành quản lý trên vật dẫn ngoài và vì vậy, theo nghĩa này, thư mục cũng được coi là file song trong một số trường hợp để phân biệt với "file" thư mục, chúng ta dùng thuật ngữ file thông thường. Khác với file thông thường, hệ điều hành lại quan tâm đến nội dung của thư mục.

7.1.2. Một số nội dung liên quan đến tên file (bao gồm cả tên thư mục)

Tên file trong Linux có thể dài tới 256 ký tự, bao gồm các chữ cái, chữ số, dấu gạch nối, gạch chân, dấu chấm. Tên thư mục/file trong Linux có thể có nhiều hơn một dấu chấm, ví dụ: This_is.a.VERY_long.filename. Nếu trong tên file có dấu chấm "." thì xâu con của tên file từ dấu chấm cuối cùng được gọi là phần mở rộng của tên file (hoặc file). Ví dụ, tên file trên đây có phần mở rộng là .filename. Chú ý rằng khái niệm phần mở rộng ở đây không mang ý nghĩa như một số hệ điều hành khác (chẳng hạn như MS-DOS).

Tên file thường là tham số thực sự khi gõ lệnh và công việc gõ lệnh trở nên rất nặng nề đối với người dùng nếu như trong lệnh phải gõ một đường dẫn dài theo dạng trên (được biết với tên gọi là đường dẫn tuyệt đối). Vì vậy, Linux (cũng như nhiều hệ điều hành khác) sử dụng khái niệm thư mục hiện thời của mỗi người dùng làm việc trong hệ thống. Thư mục hiện thời là một thư mục trong hệ thống file mà hiện thời "người dùng đang ở đó".

Qua thư mục hiện thời, Linux cho phép người dùng chỉ một file trong lệnh ngắn gọn hơn nhiều. Ví dụ, nếu thư mục hiện thời là thư mục `xinit` thì để chỉ file đã nói, người dùng chỉ cần viết `Xclients` hoặc `./Xclients` trong đó kí hiệu `./` để chỉ thư mục hiện thời.

Đường dẫn được xác định qua thư mục hiện thời được gọi là đường dẫn tương đối. Khi một người dùng đăng nhập vào hệ thống, Linux luôn chuyển người dùng vào thư mục riêng, và tại thời điểm đó thư mục riêng là thư mục hiện thời của người dùng. Thư mục riêng của siêu người dùng là `/root`, thư mục riêng của người dùng có tên là `user1` là `/home/user1` ... Linux cho phép dùng lệnh `cd` để chuyển sang thư mục khác (lấy thư mục khác làm thư mục hiện thời). Hai dấu chấm `..` được dùng để chỉ thư mục ngay trên thư mục hiện thời (cha của thư mục hiện thời).

Linux còn cho phép ghép một hệ thống file trên một thiết bị nhớ (đĩa mềm, vùng đĩa cứng chưa được đưa vào hệ thống file) thành một thư mục con trong hệ thống file của hệ thống bằng lệnh `mount`. Các hệ thống file được ghép thuộc vào các kiểu khác nhau.

7.2. Các công cụ tìm kiếm tập tin

7.2.1. Lệnh Find

Điều kiện cần đối với lệnh này là chỉ ra được điểm bắt đầu của việc tìm kiếm trong hệ thống file và những quy tắc cần tuân theo của việc tìm kiếm.

Về hình thức, lệnh **find** được sử dụng một cách đơn giản như sau:

```
# find .
```

```
.helloworld.txt
```

```
.image01.jpg
```

```
.image02.jpg
```

```
.image03.jpg
```

Lệnh này sẽ cung cấp cho ta một danh sách tất cả các tập tin và thư mục trong đường dẫn hiện hành. Lưu ý rằng truy vấn này sẽ hiển thị cả những tập tin trong cả thư mục con, vì vậy danh sách này sẽ rất dài nếu ta có nhiều file trong đó. Để dừng sự hiển thị này hãy ấn phím **Ctrl + C**.

Phương thức trên là cách đơn giản nhất để sử dụng lệnh `find`. Bằng cách kết hợp với một số các tham số và biểu thức thông thường khác, có thể làm cho việc tìm kiếm tập tin được thông minh hơn. Ví dụ, nếu muốn tìm kiếm tất cả tập tin có tên bắt đầu bằng *image*, hãy làm như sau:

```
# find . -name image\*
```

Ở đây chúng ta sử dụng tham số **-name** trong lệnh `find` để tìm kiếm tất cả tập tin có tên bắt đầu bằng *image*. Lưu ý rằng trường hợp này kết quả sẽ phân biệt chữ hoa với chữ thường. Để có kết quả không phân biệt hoa – thường ta có thể dùng lệnh

```
"# find . -iname image\*" .
```

Cũng có thể sử dụng ký tự đại diện cho việc tìm kiếm các tập tin với phần mở rộng nhất định. Để tìm kiếm toàn bộ nội dung file trong thư mục hiện hành và cả thư mục con có phần mở rộng *.php*, sử dụng lệnh sau:

```
# find . -name \*.php
```

Cũng có thể thực hiện việc tìm kiếm theo hướng ngược lại. Tức là tìm tất cả những tập tin không có phần mở rộng *.php* như sau:

```
# find . \! -name "*.php"
```

Lưu ý: dấu chấm than (!) được dùng để thực thi việc tìm kiếm theo chiều ngược lại, điều này có thể được dùng cho toàn bộ những tùy chọn khác.

Một trong những vấn đề thường gặp phải khi dùng lệnh `find` là theo mặc định nó sẽ tìm cả những thư mục con. Trong khi đó nhiều khi ta chỉ cần tìm trong thư mục hiện hành. Để làm được điều này ta cần sử dụng đến tùy chọn **-maxdepth**. Với tùy chọn này, lệnh `find` sẽ được “*thông báo*” rằng có nhiều cấp độ trong thư mục con nên nó cần “*xem xét*” khi tìm kiếm. Vì vậy nếu chỉ muốn `find` tìm trong thư mục hiện hành, chỉ cần thêm chỉ số 0 vào sau `-maxdepth`:

```
# find . \! -name "*.php" -maxdepth 0
```

(Dòng lệnh trên sẽ tìm tất cả file trong thư mục hiện tại không có phần mở rộng là `.php`)

Nếu muốn tìm kiếm sâu hơn vào cả thư mục con, chỉ cần tăng chỉ số cho `-maxdepth` lên thành 1,2... tùy theo các cấp của thư mục ta có. Tương tự, ta cũng có thể sử dụng lệnh **-mindepth** để thiết lập giới hạn thấp nhất cho cấp của thư mục được tìm kiếm. Bằng cách kết hợp hai tùy chọn `-maxdepth` và `-mindepth` một cách linh hoạt bạn sẽ có được nhiều kết quả theo nhiều nhu cầu của mình.

7.2.2. Tìm kiếm với các tiêu chí khác

Nếu như ở phần trên chúng ta đã biết cách sử dụng tùy chọn `-name` để tìm kiếm theo tên tập tin. Tương tự như vậy, trong trường hợp ta muốn tìm những kết quả theo tên người dùng thì có thể thêm tùy chọn **-user**. Ví dụ ta muốn hiển thị những tập tin có phần mở rộng `.php` của user có tên `Quantrimang`, hãy làm như sau:

```
# find . -name "*.php" -maxdepth 2 -user Quantrimang
```

7.2.3. Khắc phục lỗi thường gặp

Thông thường ta có thể sử dụng lệnh `find` trong shell scripts. Bản thân lệnh này sẽ ném ra vài lỗi. Tuy nhiên, khi chạy nó như một user bình thường các lỗi nếu gặp phải sẽ được bật lên dạng pop-up. Ví dụ, nếu ta chạy lệnh tìm kiếm trong thư mục root nhưng không có quyền root sẽ gặp lỗi kiểu **Permission denied**. Điều này sẽ gây khó chịu khi sử dụng lệnh trong một kịch bản. Mặc dù bản thân lệnh `find` không thể tự khắc phục lỗi này nhưng chúng ta vẫn có thể giải quyết dễ dàng bằng cách chuyển hướng toàn bộ lỗi về `/dev/null` như sau:

```
# find / -name StewieGriffin\*  
/root: Permission denied  
/home/peterg: Permission denied  
/home/stewie/StewieGriffin-resume.doc  
Sẽ được chuyển thành  
# find / -name StewieGriffin\* 2>/dev/null  
/home/stewie/StewieGriffin-resume.doc
```

Bây giờ ta có thể yên tâm sử dụng lệnh `find` trong các script của mình mà không cần lo lắng đến các lỗi ở đầu ra.

7.3. Di chuyển trong hệ thống tập tin

7.3.1. Sao chép file với lệnh `cp`

Lệnh `cp` có hai dạng như sau:

```
cp [tùy-chọn] <file-nguồn> ... <file-đích>
```

```
cp [tùy-chọn] --target-directory=<thư-mục> <file-nguồn>...
```

Lệnh này cho phép sao file-nguồn thành file-đích hoặc sao chép từ nhiều file-nguồn vào một thư mục đích (tham số `<file-đích>` hay `<thư-mục>`). Dạng thứ hai là một cách viết khác đối thứ tự hai tham số vị trí.

Các tùy chọn:

+ `-a`, `--archive`: giống như `-dpR` (tổ hợp ba tham số `-d`, `-p`, `-R`, như dưới đây).

+ -b, --backup[=CONTROL]: tạo file lưu cho mỗi file đích nếu như nó đang tồn tại.

+ -d, --no-dereference : duy trì các liên kết.

+ -f, --force : ghi đè file đích đang tồn tại mà không nhắc nhở.

+ -i, --interactive : có thông báo nhắc nhở trước khi ghi đè.

+ -l, --link : chỉ tạo liên kết giữa file-đích từ file-nguồn mà không sao chép.

+ -p, --preserve : duy trì các thuộc tính của file-nguồn sang file-đích.

+ -r: cho phép sao chép một cách đệ quy file thông thường.

+ -R : cho phép sao chép một cách đệ quy thư mục.

+ -s, --symbolic-link : tạo liên kết tượng trưng thay cho việc sao chép các file.

+ -S, --suffix=<hậu-tổ> : bỏ qua các hậu tố thông thường (hoặc được chỉ ra).

+ -u, --update : chỉ sao chép khi file nguồn mới hơn file đích hoặc khi file đích chưa có.

+ -v, --verbose : đưa ra thông báo về quá trình sao chép.

+ --help : hiển thị trang trợ giúp và thoát.

File đích được tạo ra có cùng kích thước và các quyền truy nhập như file nguồn, tuy nhiên file đích có thời gian tạo lập là thời điểm thực hiện lệnh nên các thuộc tính thời gian sẽ khác.

Ví dụ, lệnh

```
# cp /home/ftp/vd /home/test/vd1
```

Nếu ở vị trí đích, mô tả đầy đủ tên file đích thì nội dung file nguồn sẽ được sao chép sang file đích. Trong trường hợp chỉ đưa ra vị trí file đích được đặt trong thư mục nào thì tên của file nguồn sẽ là tên của file đích.

```
# cp /home/ftp/vd /home/test/
```

Trong ví dụ này, tên file đích sẽ là vd nghĩa là tạo một file mới /home/test/vd.

Nếu sử dụng lệnh này để sao một thư mục, sẽ có một thông báo được đưa ra cho biết nguồn là một thư mục và vì vậy không thể dùng lệnh cp để sao chép.

```
# cp . newdir
```

```
cp: .: omitting directory
```

Ví dụ về việc lệnh cp cho phép sao nhiều file cùng một lúc vào một thư mục. #

cp vd vd1 newdir

```
# pwd
```

```
/newdir
```

```
# ls -l
```

```
total 8
```

```
-rw-r--r-- 1 root ftp 15 Nov 14 11:00 vd
```

```
-rw-r--r-- 1 root ftp 12 Nov 14 11:00 vd1
```

Lưu ý:

+ Đối với nhiều lệnh làm việc với file, khi gõ lệnh có thể sử dụng kí hiệu mô tả nhóm để xác định một nhóm file làm cho tăng hiệu lực của các lệnh đó.

Ví dụ, lệnh:

```
# cp * bak
```

thực hiện việc sao chép mọi file có trong thư mục hiện thời sang thư mục con của nó có tên là bak.

Dùng lệnh

```
# cp /usr/src/linux-2.2.14/include/linux/*.h bak
```

cho phép sao chép mọi file với tên có hai kí hiệu cuối cùng là ".h" sang thư mục con bak.

Chính vì lí do nói trên, dù trong nhiều lệnh tuy không nói đến việc sử dụng kí hiệu mô tả nhóm file nhưng chúng ta có thể áp dụng chúng nếu điều đó không trái với

suy luận thông thường. Do những tình huống như thế là quá phong phú cho nên không thể giới thiệu hết trong tài liệu. Chúng ta chú ý một giải pháp là mỗi khi sử dụng một lệnh nào đó, nên thử nghiệm cách thức hiệu quả này.

7.3.2. Di chuyển tới thư mục khác

`cd <tên thư mục>` (`cd` viết tắt từ `change directory`)

Lệnh này sẽ di chuyển bạn tới thư mục mới. Nếu bạn không gõ tên thư mục, lệnh sẽ tự động chuyển bạn trở về thư mục chủ. Tương đương với lệnh `cd ~` (dấu ngã) `cd ..`

Di chuyển lên một cấp thư mục. Ví dụ đang ở thư mục `/home/sti`, sau khi gõ lệnh sẽ di chuyển tới thư mục `/home`

7.3.3. Sao chép thư mục

`cp <tên thư mục 1> <tên thư mục 2>` (`cp` viết tắt của từ `copy`)

Lệnh này sẽ sao chép nội dung thư mục 1 sang thư mục 2.

Chú ý: lệnh trên chỉ sao chép được các tập tin trong thư mục 1 sang thư mục 2 mà thôi. Nếu có thư mục con, nó sẽ báo lỗi. Vì thế bạn nên sử dụng thêm tùy chọn `-r` để có thể sao chép được các thư mục con trong thư mục 1 sang thư mục 2

`cp -r <tên thư mục 1> <tên thư mục 2>`

Chú ý: nếu thư mục 2 tồn tại trên máy bạn, thư mục 1 sẽ được chép vào thư mục 2 và trở thành thư mục con của thư mục 2.

7.3.4. Di chuyển thư mục

`mv <tên thư mục 1> <tên thư mục 2>` (`mv` viết tắt của từ `move`)

Ta cũng dùng tùy chọn `-r` nếu muốn di chuyển tất cả nội dung trong thư mục.

Chú ý: nếu thư mục 2 tồn tại trên máy bạn, thư mục 1 trở thành thư mục con của thư mục 2.

7.4. Quản lý ổ đĩa và phân vùng

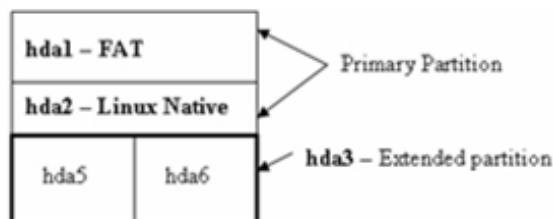
7.4.1. Các lệnh quản lý ổ đĩa

Linux sử dụng cơ chế truy xuất ổ đĩa thông qua tập tin. Mỗi ổ đĩa được gán với một tập tin trong thư mục `/dev/`. Ký hiệu ổ đĩa `fd` cho ổ mềm, `hd` cho ổ cứng, `sd` dành cho ổ SCSI. Ký tự `a, b, c ...`, gán thêm vào để xác định các ổ đĩa khác nhau cùng loại.

Ký tự mô tả ổ đĩa	Physical block devices(Các thiết bị lưu trữ)
Hda	Primary Master
Hdb	Primary Slave
Hdc	Secondary Master
Hdd	Secondary Slave
Sda	First SCSI disk
Sdb	Second SCSI disk

Ví dụ :

Ổ cứng thứ nhất `hda`, ổ cứng thứ 2 `hdb` ...xác định các partition trong ổ đĩa người ta dùng các số đi kèm. Theo qui định partition chính và mở rộng được gán số từ 1 - 4. Các logical partition được gán các giá trị từ 5 trở đi.



Như hình vẽ trên là các partition của ổ cứng thứ nhất `hda`: có 2 partition chính ký hiệu là `hda1` và `hda2`, một partition mở rộng là `hda3`. Trong partition mở rộng `hda3` có 2 partition logic có ký hiệu là `hda6` và `hda5`.

7.4.2. Phân vùng

Trong Linux bắt buộc phải có tối thiểu 2 partition sau:

-Partition chính chứa thư mục gốc (/) và hạt nhân (gọi là Linux Native partition)

-Partition swap được dùng làm không gian hoán đổi dữ liệu khi vùng nhớ chính được sử dụng hết. Kích thước của phần swap sử dụng tùy thuộc hệ thống mình sử dụng nhiều hay ít ứng dụng. Thông thường thì kích thước vùng swap bằng kích thước bộ nhớ chính.

Vì vậy đĩa cứng được phân ra nhiều vùng khác nhau gọi là partition. Mỗi partition sử dụng một hệ thống tập tin và lưu trữ dữ liệu. Mỗi đĩa bạn chỉ chia được tối đa 4 partition chính (primary). Giới hạn như vậy là do Master Boot Record của đĩa chỉ ghi tối đa 4 chỉ mục tới 4 partition.

Để tạo nhiều partition lưu trữ dữ liệu (hơn 4) người ta dùng partition mở rộng (extended partition). Thực ra partition mở rộng cũng là primary partition nhưng cho phép tạo các partition con được gọi là logical partition trong nó.

7.5. Bảo trì hệ thống tập tin

7.5.1. Quyền hạn

Do Linux là một hệ điều hành đa nhiệm (multitasking) và đa người dùng (multiuser), nhiều người có thể cùng sử dụng một máy Linux và một người có thể cho chạy nhiều chương trình khác nhau. Có hai vấn đề lớn được đặt ra: quyền sở hữu các dữ liệu trên đĩa và phân chia tài nguyên hệ thống như CPU, RAM giữa các tiến trình (process).

Tất cả các tập tin và thư mục của Linux đều có người sở hữu và quyền truy cập. Ta có thể thay đổi các tính chất này đối với tập tin hay thư mục. Quyền của tập tin còn cho phép xác định tập tin có phải là một chương trình (application) hay không (khác với MSDOS và MSWindows xác định tính chất này qua phần mở rộng của tên tập tin). Ví dụ với lệnh `ls -l`:

```
-rw-r--r-- 1 fido users 163 Dec 7 14:31 myfile
```

Cột đầu chỉ ra quyền hạn truy cập của tập tin, ví dụ trên, các ký tự `-rw-r--r--` biểu thị quyền truy cập của tập tin myfile. Linux cho phép người sử dụng xác định các quyền đọc (read), viết (write) và thực thi (execute) cho từng đối tượng. Có 3 dạng đối tượng

- + Người sở hữu (the owner)
- + Nhóm sở hữu (the group owner)
- + Người khác ("other users" hay everyone else).

Quyền đọc cho phép bạn đọc nội dung của tập tin. Đối với thư mục, quyền đọc cho phép bạn di chuyển vào thư mục và xem nội dung của thư mục.

Quyền viết cho phép bạn thay đổi nội dung hay xóa tập tin. Đối với thư mục, quyền viết cho phép bạn tạo ra, xóa hay thay đổi tên các tập tin trong thư mục không phụ thuộc vào quyền cụ thể của tập tin trong thư mục. Như vậy, quyền viết của thư mục sẽ vô hiệu hóa các quyền truy cập của tập tin trong thư mục và bạn đọc phải để ý tính chất này.

Quyền thực thi cho phép bạn gọi chương trình lên bộ nhớ bằng cách nhập từ bàn phím tên của tập tin. Đối với thư mục, bạn chỉ có thể vào thư mục bởi lệnh `cd` nếu bạn có quyền thực thi với thư mục.

```
-rw-r--r-- 1 fido users 163 Dec 7 14:31 myfile
```

Ký tự đầu tiên của quyền là ký tự `-` cho biết đó là một tập tin bình thường. Nếu ký tự thay thế cho dấu `-` thì myfile là một thư mục. Ngoài ra còn có c cho thiết

bị ngoại vi dạng ký tự (như bàn phím), b cho thiết bị ngoại vi dạng block (như ổ đĩa cứng).

Chín ký tự tiếp theo chia thành 3 nhóm, cho phép xác định quyền của 3 nhóm: người sở hữu (owner), nhóm sở hữu(group) và những người còn lại (other). Mỗi cặp ba này cho phép xác định quyền đọc, viết và thực thi theo thứ tự kể trên. Quyền đọc viết tắt là “r” ở vị trí đầu, quyền viết tắt bằng “w” ở vị trí thứ hai và vị trí thứ ba là quyền thực thi ký hiệu bằng chữ “x”. Nếu một quyền không được cho thì tại vị trí đó sẽ có ký tự “-”.

Ký tự	r	w	x	r	w	x	r	w	X
Loại tập tin	Owner			group owner			other users		

Trong trường hợp của tập tin myfile, người sở hữu có quyền rw tức là đọc và viết. Nhóm sở hữu và những người còn lại chỉ có quyền đọc tập tin (read-only). Bên cạnh đó, bạn còn biết myfile không phải là một chương trình.

Song song với cách ký hiệu miêu tả bằng ký tự ở trên, quyền hạn truy cập còn có thể biểu diễn dưới dạng 3 số. Quyền hạn cho từng loại người dùng sử dụng một số có 3 bit tương ứng cho 3 quyền read, write và execute. Theo đó nếu cấp quyền thì bit đó là 1, ngược lại là 0. Giá trị nhị phân của số 3 bit này xác định các quyền cho nhóm người đó.

Bit 2	bit 1	bit 0
read	write	execute

Ví dụ: chỉ có quyền đọc: 100 có giá trị là 4 có quyền đọc và thực thi : 101 có giá trị là 5

Theo cách tính số thập phân ta cũng có thể xác định số quyền hạn bằng cách tính tổng giá trị của các quyền. Theo quy định trên ta có giá trị tương ứng như sau:

Quyền	Giá trị
Read permission	4
Write permission	2
Execute permission	1

Ví dụ: Nếu quyền read và execute thì số của quyền là 5: 4+1=5

Read, write và execute: 4+2+1=7

Tổ hợp của 3 quyền trên có giá trị từ 0 đến 7

- + 0 or ---: Không có quyền
- + 1 or --x: execute
- + 2 or -w-: write-only (rare)
- + 3 or -wx: write và execute
- + 4 or r--: read-only
- + 5 or r-x: read và execute
- + 6 or rw-: read và write
- + 7 or rwx: read, write và execute

Như vậy khi cấp quyền trên một tập tin/thư mục, bạn có thể dùng số thập phân gồm 3 con số. Số đầu tiên miêu tả quyền của sở hữu, số thứ hai cho nhóm và số thứ ba cho những người còn lại.

Ví dụ: Một tập tin với quyền 751 có nghĩa là sở hữu có quyền read, write và execute bằng 4+2+1=7. Nhóm có quyền read và execute bằng 4+1=5 và những người còn lại có quyền execute bằng 1.

Chú ý: Người sử dụng có quyền đọc thì có quyền copy tập tin. Khi đó, tập tin sao chép sẽ thuộc sở hữu người làm copy. Ví dụ minh họa sau:

```
$ ls -l /etc/passwd
-rw-r--r-- 1 root root 1113 Oct 13 12:30 /etc/passwd
$ cp /etc/passwd ./
$ ls -l passwd
-rw-r--r-- 1 ndhung admin 1113 Oct 15 10:37 passwd
```

7.5.2. Lệnh chmod, chown, chgrp

+ Lệnh chmod

Đây là lệnh được sử dụng rất phổ biến, dùng cấp phép quyền hạn truy cập của tập tin hay thư mục. Chỉ có chủ sở hữu và superuser mới có quyền thực hiện các lệnh này.

Cú pháp của lệnh: \$chmod [nhóm-người-dùng] [thao-tác] [quyền-hạn] [tên-tập-tin].

Nhóm-người-dùng	Thao tác	Quyền
u - user	+ : thêm quyền	r - read
g - group	- : xóa quyền	w - write
o - others	= : gán ngang quyền	x - excute
a - all		

Một số ví dụ : gán quyền trên tập tin myfile

Gán thêm quyền write cho group : \$ chmod g+w myfile

Xóa quyền read trên group và others : \$ chmod go-w myfile Cấp quyền x cho mọi người:

\$ chmod ugo+x myfile hoặc

\$ chmod a+x myfile hoặc

\$ chmod +x myfile

Đây là cách thay đổi tương đối vì kết quả cuối cùng phụ thuộc vào quyền đã có trước đó mà lệnh này không liên quan đến. Trên quan điểm bảo mật hệ thống, cách thay đổi tuyệt đối dẫn đến ít sai sót hơn. Thay đổi quyền truy cập của một thư mục cũng được thực hiện giống như đối với một tập tin. Mọi người sử dụng có quyền viết vào thư mục đều có quyền xóa tập tin trong thư mục đó, không phụ thuộc vào quyền của người đó đối với các tập tin trong thư mục. Vì vậy, đa số các thư mục có quyền drwxr-xr-x. Như vậy chỉ có người sở hữu của thư mục mới có quyền tạo và xóa tập tin trong thư mục. Ngoài ra, thư mục còn có một quyền đặc biệt, đó là cho phép mọi người đều có quyền tạo tập tin trong thư mục, mọi người đều có quyền thay đổi nội dung tập tin trong thư mục, nhưng chỉ có người tạo ra mới có quyền xóa tập tin. Đó là dùng sticky bit cho thư mục. Thư mục /tmp thường có sticky bit bật lên.

```
drwxrwxrwt 7 root root 16384 Oct 21 15:33 tmp
```

Ta thấy chữ t, cuối cùng trong nhóm các quyền, thể hiện cho sticky bit của /tmp. Để có sticky bit, ta sử dụng lệnh: chmod 1????????? tên_thư_mục.

Ngoài cách gán quyền trên, chúng ta cũng có thể gán quyền trực tiếp thông qua 3 chữ số xác định quyền như sau : \$chmod [giá-trị-quyền] [tên-tập-tin]

Ví dụ: Cấp quyền cho tập tin myfile

Quyền	Lệnh
-wrxr-xr-x	\$chmod 755 myfile
-r-xr--r--	\$chmod 522 myfile
-rwxrwxrwx	\$chmod 777 myfile

Phương pháp thay đổi tuyệt đối này có một số ưu điểm vì nó là cách định quyền tuyệt đối, kết quả cuối cùng không phụ thuộc vào quyền truy cập trước đó của tập tin. Đồng thời, để nói “thay quyền tập tin thành 755” thì dễ hơn là “thay quyền tập tin thành read-write-execute, read-execute, read-execute”

+ **Lệnh chown**

Lệnh chown dùng để thay đổi người sở hữu trên tập tin, thư mục. Cú pháp:

\$chown [tên-user:tên-nhóm] [tên-tập-tin/thư-mục]

\$chown -R [tên-user:tên-nhóm] [thư-mục]

Dòng lệnh cuối cùng với tùy chọn -R (recursive) cho phép thay đổi người sở hữu của thư mục <tên_thư_mục> và tất cả các thư mục con của nó.

+ **Lệnh chgrp**

Lệnh chgrp dùng để thay đổi nhóm sở hữu của một tập tin, thư mục. Cú pháp:

\$chgrp [nhóm-sở-hữu] [tên-tập-tin/thư-mục]

Câu hỏi ôn tập chương

1. Trình bày khái niệm hệ thống tập tin? Các công cụ tìm kiếm tập tin trong Linux?
2. Trình bày các lệnh quản lý tập tin, quản lý ổ đĩa và phân vùng trên Linux?
3. Trình bày các cách bảo trì hệ thống tập tin trong Linux?

Chương 8: Quản lý người dùng

Mục tiêu:

- Hiểu khái niệm quản lý người dùng trên Linux;
- Quản lý tài khoản người dùng và các nhóm người dùng trên Linux;
- Phân quyền trên hệ thống tập tin;
- Có thái độ nghiêm túc, cẩn thận và chính xác.

8.1. Khái niệm tài khoản người dùng

Như đã biết, trong hệ điều hành đa người dùng, cần phân biệt người dùng khác nhau do quyền sở hữu các tài nguyên trong hệ thống, chẳng hạn như, mỗi người dùng có quyền hạn với file, quá trình của riêng họ. Điều này vẫn rất quan trọng thậm chí cả khi máy tính chỉ có một người sử dụng tại một thời điểm. Mọi truy cập hệ thống Linux đều thông qua tài khoản người dùng. Vì thế, mỗi người sử dụng được gán với tên duy nhất (đã được đăng ký) và tên đó được sử dụng để đăng nhập. Tuy nhiên một người dùng thực sự có thể có nhiều tên đăng nhập khác nhau. Tài khoản người dùng có thể hiểu là tất cả các file, các tài nguyên, và các thông tin thuộc về người dùng đó.

Khi cài đặt hệ điều hành Linux, đăng nhập root sẽ được tự động tạo ra. Đăng nhập này được xem là thuộc về siêu người dùng (người dùng cấp cao, người quản trị), vì khi đăng nhập với tư cách người dùng root, có thể làm bất cứ điều gì muốn trên hệ thống. Tốt nhất chỉ nên đăng nhập root khi thực sự cần thiết, và hãy đăng nhập vào hệ thống với tư cách là một người dùng bình thường.

Nội dung chương này giới thiệu các lệnh để tạo một người dùng mới, thay đổi thuộc tính của một người dùng cũng như xóa bỏ một người dùng. Lưu ý, chỉ có thể thực hiện được các lệnh trên nếu có quyền của một siêu người dùng.

8.2. Hệ thống quản lý người dùng

8.2.1. Quyền truy nhập

Mỗi file và thư mục trong Linux đều có một chủ sở hữu và một nhóm sở hữu, cũng như một tập hợp các quyền truy nhập. Cho phép thay đổi các quyền truy nhập và quyền sở hữu file và thư mục nhằm cung cấp truy nhập nhiều hơn hay ít hơn. Thông tin về một file có dạng sau (được hiện ra theo lệnh hiện danh sách file **ls -l**):

drwxr-xr-x	12	root	root	4096	Oct 23 2000	LinuxVN.com
Tập hợp	Số liên	Người chủ	Nhóm chủ	Kích	Ngày giờ	Tên file
quyền truy	kết	file (thư	file	thước	Tạo file	
nhập	đến file	mục)		File (byte)		

Trong đó, dãy 10 ký tự đầu tiên mô tả kiểu file và quyền truy nhập đối với tập tin đó.

Theo mặc định, người dùng tạo một file chính là người chủ (sở hữu) của file đó và là người có quyền sở hữu nó. Người chủ của file có đặc quyền thay đổi quyền truy nhập hay quyền sở hữu đối với file đó. Tất nhiên, một khi đã chuyển quyền sở hữu của mình cho người dùng khác thì người chủ cũ không được phép chuyển quyền sở hữu và quyền truy nhập được nữa.

Tập hợp một chuỗi có 10 ký tự đã giới thiệu trên đây được chia ra làm 4 phần: kiểu file, các quyền truy nhập đến file của chủ sở hữu, của nhóm sở hữu và người dùng khác.

Có một số kiểu file trong Linux. Ký tự đầu tiên trong tập hợp 10 ký tự mô tả kiểu file và quyền truy nhập sẽ cho biết file thuộc kiểu nào (chữ cái đó được gọi là chữ cái biểu diễn).

Bảng 8.2.1.1 Liệt kê các kiểu file trong Linux

Chữ cái biểu diễn

Kiểu file

d	Thư mục (directory)
b	File kiểu khối (block-type special file)
c	File kiểu ký tự (character-type special file)
l	Liên kết tượng trưng (symbolic link)
p	File đường ống (pipe)
s	Socket
-	File bình thường (regular file)

Chín ký tự tiếp theo trong chuỗi là quyền truy nhập được chia ra làm 3 nhóm tương ứng với quyền truy nhập của người sử hữu, nhóm sở hữu và người dùng khác. Ví dụ, 10 ký tự đầu tiên trong dòng ví dụ ngay trước đây sẽ được phân tích thành:

d rwx r-x r-x

Kiểu file: thư	Quyền của	Quyền của	Quyền của người
mục	người chủ	nhóm chủ	dùng khác

Để hiểu được chính xác quyền truy nhập có ý nghĩa như thế nào đối với hệ thống máy tính, phải nhớ rằng **Linux xem mọi thứ đều là file**. Nếu cài đặt một ứng dụng, nó cũng sẽ được xem như mọi chương trình khác, trừ một điều: hệ thống nhận biết rằng một ứng dụng là một chương trình khả thi, tức là nó có thể chạy được. Một bức thư gửi cho mẹ là một dạng file văn bản bình thường, nhưng nếu thông báo cho hệ thống biết đó là một chương trình khả thi, hệ thống sẽ cố để chạy chương trình (và tất nhiên là lỗi).

Có ba loại quyền truy nhập chính đối với thư mục/file, đó là: đọc (read - r), ghi (write -w) và thực hiện (execute - x). Quyền đọc cho phép người dùng có thể xem nội dung của file với rất nhiều chương trình khác nhau, nhưng họ sẽ không thể thay đổi, sửa chữa hoặc xóa bất kỳ thông tin nào trong đó. Tuy nhiên, họ có thể sao chép file đó thành file của họ và sửa chữa file bản sao.

Quyền ghi là quyền truy nhập tiếp theo. Người sử dụng với quyền ghi khi truy nhập vào file có thể thêm thông tin vào file. Nếu có quyền ghi và quyền đọc đối với một file, có thể soạn thảo lại file đó - quyền đọc cho phép xem nội dung, và quyền ghi cho phép thay đổi nội dung file. Nếu chỉ có quyền ghi, sẽ thêm được thông tin vào file, nhưng lại không thể xem được nội dung của file. Loại quyền truy nhập thứ ba là quyền thực hiện, quyền này cho phép người dùng có thể chạy được file, nếu đó là một chương trình khả thi. Quyền thực hiện độc lập với các quyền truy nhập khác, vì thế hoàn toàn có thể có một chương trình với quyền đọc và quyền thực hiện, nhưng không có quyền ghi. Cũng có trường hợp một chương trình chỉ có quyền thực hiện, có nghĩa là người dùng có thể chạy ứng dụng, nhưng họ không thể xem được cách nó làm việc hay sao chép nó.

Bảng 8.2.1.2 Cách ký hiệu của các quyền truy nhập:

Quyền truy nhập	Ý nghĩa
---	Không cho phép một quyền truy nhập nào
r--	Chỉ được quyền đọc
r-x	Quyền đọc và thực hiện (cho chương trình và shell script)
rw-	Quyền đọc và ghi
rwx	Cho phép tất cả các quyền truy nhập (cho chương trình)

Tuy nhiên, đối với thư mục thì chỉ có ba loại ký hiệu của các quyền truy nhập là: ---, **r-x** và **rwx**, vì nội dung của thư mục là danh sách của các file và các thư mục

con có bên trong thư mục đó. Quyền đọc một thư mục là được xem nội dung của thư mục đó và quyền thực hiện đối với một thư mục là quyền tìm được file và thư mục con có trong thư mục. Như vậy, với ví dụ đang được xem xét, chúng ta nhận được đây là một thư mục và quyền truy nhập nó được giải thích như sau:

d rwx r-x r-x

Kiểu file:	Quyền của	Quyền của	Quyền người
thư mục	người chủ:	nhóm chủ:	dùng khác:
	đọc, ghi,	đọc, thực	đọc, thực hiện
	thực hiện	hiện	(không ghi)
		(không ghi)	

8.2.2. Tạo tài khoản người dùng

Để tạo một tài khoản, bạn có thể sử dụng lệnh `useradd`, cú pháp lệnh `useradd` như sau: `#useradd [-c lời_mô_tả_về_người_dùng] [-d thư_mục_cá_nhân] [-m] [-g nhóm_của_người_dùng] [tên_tài_khoản]`

Lưu ý: Tham số `-m` được sử dụng để tạo thư mục cá nhân nếu nó chưa tồn tại. Và chỉ có root được phép sử dụng lệnh này.

Ví dụ: `# useradd -c "Nguyễn van B" nvb`

Dùng lệnh `passwd <username>` để đặt mật khẩu cho tài khoản. `# passwd nvb`

Changing password for user nvb

New UNIX password: ****

Retype new UNIX password: ****

passwd: all authentication tokens updated successfully

Vì vấn đề an ninh cho máy Linux và sự an toàn của toàn hệ thống mạng, việc chọn đúng password là rất quan trọng. Một password gọi là tốt nếu:

- Có độ dài tối thiểu 6 ký tự.
- Phối hợp giữa chữ thường, chữ hoa, số và các ký tự đặc biệt.
- Không liên quan đến tên tuổi, ngày sinh ... của bạn và người thân.

Trong ví dụ trên, bạn tạo tài khoản người dùng và không quan tâm gì đến nhóm (group) của người dùng. Sẽ thuận lợi nếu bạn nhóm nhiều người dùng có cùng một chức năng và cùng chia sẻ nhau dữ liệu vào chung một nhóm. Mặc định khi bạn tạo một tài khoản, Linux sẽ tạo cho mỗi tài khoản một nhóm, tên nhóm trùng với tên tài khoản. Đọc tập tin `/etc/passwd` ta thấy:

```
nvb:x:1013:1013::/home/nvb:/bin/bash
```

nvb có user_ID 1012 và thuộc nhóm 1013.

Xem tập tin `/etc/group` ta thấy:

```
# more /etc/group
```

```
root:x:0:root
```

```
.....
```

```
users:x:100:
```

```
.....
```

```
nvb:x:1013:
```

Ta có thể kết nạp tài khoản nvb vào nhóm users bằng cách thay số 1013 bằng 100, là group_ID của nhóm users. Ta có thể dùng lệnh `useradd -d` để xem các thông số mặc định khi ta tạo tài khoản người dùng (các thông tin này được lưu trong thư mục `/etc/default/useradd`):

```
# useradd -D
```

```
GROUP=100
```

```
HOME=/home
```



```
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
```

8.2.3. Thay đổi thông tin của tài khoản

Ta có thể thay đổi lại thông tin tài khoản từ tập tin `/etc/passwd` hoặc dùng lệnh `usermod`. Cú pháp của lệnh `usermod`:

```
#usermod
nhóm_của_người_dùng] [tên_tài_khoản].
Ví dụ: Cho tài khoản nvb vào nhóm admin
```

```
#usermod -g admin nvb
```

8.2.4. Tạm khóa tài khoản người dùng

Để tạm thời khóa tài khoản trong hệ thống ta có thể dùng nhiều cách:

Khóa (locking) Mở khóa (unlock)

```
passwd -l <username>        passwd -u
```

```
usermod -L <username>        usermod -U
```

Ta có tạm khóa tài khoản bằng cách chỉnh sửa tập tin `/etc/shadow` và thay thế từ khóa `x` bằng từ khóa `*` hoặc có gán `/bin/false` vào shell mặc định của user trong file `/etc/passwd`

8.2.5. Hủy tài khoản

Lệnh `userdel` dùng để xóa một tài khoản. Ngoài ra, bạn cũng có thể xóa một tài khoản bằng cách xóa đi dòng dữ liệu tương ứng với tài khoản đó trong tập tin `/etc/passwd`. Cú pháp của lệnh: `#userdel <option> [username]`

Ví dụ xóa tài khoản nvb (dùng tùy chọn `-r` để xóa toàn bộ thông tin liên quan tới user đó): `#userdel -r nvb`

8.3. Các lệnh quản lý người dùng

Người dùng được quản lý thông qua tên người dùng (thực ra là chỉ số người dùng). Nhân hệ thống quản lý người dùng theo chỉ số, vì việc quản lý theo chỉ số sẽ dễ dàng và nhanh thông qua một cơ sở dữ liệu lưu trữ các thông tin về người dùng. Việc thêm một người dùng mới chỉ có thể thực hiện được nếu đăng nhập với tư cách là siêu người dùng.

Để tạo một người dùng mới, cần phải thêm thông tin về người dùng đó vào trong cơ sở dữ liệu người dùng, và tạo một thư mục cá nhân cho riêng người dùng đó. Điều này rất cần thiết để thiết lập các biến môi trường phù hợp cho người dùng.

Lệnh chính để thêm người dùng trong hệ thống Linux là `useradd` (hoặc `adduser`).

8.3.1. File `/etc/passwd`

Danh sách người dùng cũng như các thông tin tương ứng được lưu trữ trong file `/etc/passwd`.

Ví dụ dưới đây là nội dung của file `/etc/passwd`:

```
mail:x:8:12:mail:/var/spool/mail:
games:x:12:100:games:/usr/games:
gopher:x:13:30:gopher:/usr/lib/gopher-data:
bien:x:500:0:Nguyen Thanh Bien:/home/bien:/bin/bash
sangnm:x:17:100:Nguyen Minh Sang:/home/sangnm:/bin/bash
lan:x:501:0:Lan GNU:/home/lan:/bin/bash
```

Mỗi dòng trong file tương ứng với bảy trường thông tin của một người dùng, và các trường này được ngăn cách nhau bởi dấu `'.'`. ý nghĩa của các trường thông tin đó lần lượt như sau:

- + Tên người dùng (username)
- + Mật khẩu người dùng (passwd - được mã hóa)
- + Chỉ số người dùng (user id)
- + Các chỉ số nhóm người dùng (group id)
- + Tên đầy đủ hoặc các thông tin khác về tài khoản người dùng (comment) TM

Thư mục để người dùng đăng nhập

- + Shell đăng nhập (chương trình chạy lúc đăng nhập)

Bất kỳ người dùng nào trên hệ thống đều có thể đọc được nội dung file /etc/passwd, và có thể đăng nhập với tư cách người dùng khác nếu họ biết được mật khẩu, đây chính là lý do vì sao mật khẩu đăng nhập của người dùng không hiển thị trong nội dung file.

8.3.2. Thêm người dùng với lệnh useradd

Siêu người dùng sử dụng lệnh useradd để tạo một người dùng mới hoặc cập nhật ngầm định các thông tin về người dùng.

Cú pháp lệnh:

```
useradd [tùy-chọn] <tên-người-dùng> useradd -D [tùy-chọn]
```

Nếu không có tùy chọn -D, lệnh useradd sẽ tạo một tài khoản người dùng mới sử dụng các giá trị được chỉ ra trên dòng lệnh và các giá trị mặc định của hệ thống. Tài khoản người dùng mới sẽ được nhập vào trong các file hệ thống, thư mục cá nhân sẽ được tạo, hay các file khởi tạo được sao chép, điều này tùy thuộc vào các tùy chọn được đưa ra.

Các tùy chọn như sau:

- c, comment : soạn thảo trường thông tin về người dùng.
- d, home_dir : tạo thư mục đăng nhập cho người dùng.
- e, expire_date : thiết đặt thời gian (YYYY-MM-DD) tài khoản người dùng sẽ bị hủy bỏ.

-f, inactive_days : tùy chọn này xác định số ngày trước khi mật khẩu của người dùng hết hiệu lực khi tài khoản bị hủy bỏ. Nếu =0 thì hủy bỏ tài khoản người dùng ngay sau khi mật khẩu hết hiệu lực, =-1 thì ngược lại (mặc định là -1).

-g, initial_group : tùy chọn này xác định tên hoặc số khởi tạo đăng nhập nhóm người dùng.

Tên nhóm phải tồn tại, và số của nhóm phải tham chiếu đến một nhóm đã tồn tại. Số nhóm ngầm định là 1.

-G, group : danh sách các nhóm phụ mà người dùng cũng là thành viên thuộc các nhóm đó.

Mỗi nhóm sẽ được ngăn cách với nhóm khác bởi dấu ',', mặc định người dùng sẽ thuộc vào nhóm khởi tạo.

-m : với tùy chọn này, thư mục cá nhân của người dùng sẽ được tạo nếu nó chưa tồn tại.

-M : không tạo thư mục người dùng.

-n : ngầm định khi thêm người dùng, một nhóm cùng tên với người dùng sẽ được tạo. Tùy chọn này sẽ loại bỏ sự ngầm định trên.

-p, passwd : tạo mật khẩu đăng nhập cho người dùng.

-s, shell : thiết lập shell đăng nhập cho người dùng.

Thay đổi các giá trị ngầm định

Khi tùy chọn -D được sử dụng, lệnh useradd sẽ bỏ qua các giá trị ngầm định và cập nhật các giá trị mới.

-b, default_home : thêm tên người dùng vào cuối thư mục cá nhân để tạo tên thư mục cá nhân mới.

-e, default_expire_date : thay đổi thời hạn hết giá trị của tài khoản người dùng.

-f, default_inactive : xác định thời điểm hết hiệu lực của mật khẩu đăng nhập khi tài khoản người dùng bị xóa bỏ.

-g, default_group : thay đổi chỉ số nhóm người dùng.

-s, default_shell : thay đổi shell đăng nhập.

Ngoài lệnh useradd, có thể tạo người dùng mới bằng cách sau:

Soạn thảo file /etc/passwd bằng vipw. Lệnh vipw mở trình soạn thảo trên hệ thống và hiệu chỉnh bản sao tạm của file /etc/passwd. Việc sử dụng file tạm và khóa file sẽ có tác dụng như một cơ chế khóa để ngăn việc hai người dùng cùng soạn thảo file một lúc. Lúc đó sẽ thêm dòng thông tin mới về người dùng cần tạo. Hãy cẩn thận trong việc soạn thảo tránh nhầm lẫn. Riêng trường mật khẩu nên để trống và tạo mật khẩu sau. Khi file này được lưu, vipw sẽ kiểm tra sự đồng nhất trên file bị thay đổi. Nếu tất cả mọi thứ dường như thích hợp thì có nghĩa là file /etc/passwd đã được cập nhật.

Ví dụ: thêm người dùng có tên là new, chỉ số người dùng 503, chỉ số nhóm là 100, thư mục cá nhân là /home/new và shell đăng nhập là shell bash:

```
# vipw
```

```
mail:x:8:12:mail:/var/spool/mail:
```

```
games:x:12:100:games:/usr/games:
```

```
gopher:x:13:30:gopher:/usr/lib/gopher-data:
```

```
bien:x:500:0:Nguyen Thanh Bien:/home/bien:/bin/bash
```

```
sang:x:17:100:Nguyen Minh Sang:/home/sangnm:/bin/bash lan:x:501:0:Lan GNU:/home/lan:/bin/bash
```

```
new::503:100:them mot nguoi moi:/home/new:/bin/bash
```

Tạo thư mục cá nhân của người dùng mới với lệnh mkdir

```
# mkdir /home/new
```

Sao chép các file từ thư mục /etc/skel/ (đây là thư mục lưu trữ các file cần thiết cho người dùng) vào file cá nhân vừa tạo

Thay đổi quyền sở hữu và các quyền truy nhập file /home/new với các lệnh chown và chmod

```
# chown new /home/new
```

```
# chmod go=u,go-w /home/new
```

Thiết lập mật khẩu của người dùng với lệnh passwd # passwd new passwd:

Sau khi thiết lập mật khẩu cho người dùng ở bước cuối cùng, tài khoản người dùng sẽ làm việc. Nên thiết lập mật khẩu người dùng ở bước cuối cùng, nếu không họ có thể vô tình đăng nhập trong khi đang sao chép các file.

8.3.3. Thay đổi thuộc tính người dùng

Trong Linux có rất nhiều lệnh cho phép thay đổi một số các thuộc tính của tài khoản người dùng như:

chfn: thay đổi thông tin cá nhân của người dùng.

chsh: thay đổi shell đăng nhập.

passwd: thay đổi mật khẩu.

Một số các thuộc tính khác sẽ phải thay đổi bằng tay. Ví dụ, để thay đổi tên người dùng, cần soạn thảo lại trực tiếp trên file /etc/passwd (với lệnh vipw). Nhưng có một lệnh tổng quát cho phép có thể thay đổi bất kỳ thông tin nào về tài khoản người dùng, đó là lệnh usermod.

Cú pháp lệnh:

`usermod [tùy-chọn] <tên-đăng-nhập>`

Lệnh `usermod` sửa đổi các file tài khoản hệ thống theo các thuộc tính được xác định trên dòng lệnh.

Các tùy chọn của lệnh:

- c, comment : thay đổi thông tin cá nhân của tài khoản người dùng.
- d, home_dir : thay đổi thư mục cá nhân của tài khoản người dùng.
- e, expire_date : thay đổi thời điểm hết hạn của tài khoản người dùng
- f, inactive_days : thiết đặt số ngày hết hiệu lực của mật khẩu trước khi tài khoản người dùng hết hạn sử dụng.

- g, initial_group : tùy chọn này thay đổi tên hoặc số khởi tạo đăng nhập nhóm người dùng.

Tên nhóm phải tồn tại, và số của nhóm phải tham chiếu đến một nhóm đã tồn tại. Số nhóm ngầm định là 1.

- G, group : thay đổi danh sách các nhóm phụ mà người dùng cũng là thành viên thuộc các nhóm đó. Mỗi nhóm sẽ được ngăn cách với nhóm khác bởi dấu ',' mặc định người dùng sẽ thuộc vào nhóm khởi tạo.

- l, login_name : thay đổi tên đăng nhập của người dùng. Trong một số trường hợp, tên thumục riêng của người dùng có thể sẽ thay đổi để tham chiếu đến tên đăng nhập mới.

- p, passwd : thay đổi mật khẩu đăng nhập của tài khoản người dùng.

- s, shell : thay đổi shell đăng nhập.

- u, uid : thay đổi chỉ số người dùng.

Lệnh `usermod` không cho phép thay đổi tên của người dùng đang đăng nhập. Phải đảm bảo rằng người dùng đó không thực hiện bất kỳ quá trình nào trong khi lệnh `usermod` đang thực hiện thay đổi các thuộc tính của người dùng đó.

Ví dụ muốn thay đổi tên người dùng `new` thành tên mới là `newuser`, hãy gõ lệnh sau: `# usermod -l new newuser`

8.3.4. Xóa bỏ một người dùng (lệnh `userdel`)

Để xóa bỏ một người dùng, trước hết phải xóa bỏ mọi thứ có liên quan đến người dùng đó. Lệnh hay được dùng để xóa bỏ một tài khoản người dùng là lệnh `userdel` với cú pháp:

`userdel[-r] <tên-người-dùng>`

Lệnh này sẽ thay đổi nội dung của các file tài khoản hệ thống bằng cách xóa bỏ các thông tin về người dùng được đưa ra trên dòng lệnh. Người dùng này phải thực sự tồn tại. Tùy chọn `-r` có ý nghĩa:

- r : các file tồn tại trong thư mục riêng của người dùng cũng như các file nằm trong các thư mục khác có liên quan đến người dùng bị xóa bỏ cùng lúc với thư mục người dùng.

Lệnh `userdel` sẽ không cho phép xóa bỏ người dùng khi họ đang đăng nhập vào hệ thống. Phải hủy bỏ mọi quá trình có liên quan đến người dùng trước khi xóa bỏ người dùng đó. Ngoài ra cũng có thể xóa bỏ tài khoản của một người dùng bằng cách hiệu chỉnh lại file `/etc/passwd`.

8.4. Phân quyền trên hệ thống tập tin

Hệ thống tập tin của Linux và Unix được tổ chức theo một hệ thống phân bậc tương tự cấu trúc của một cây phân cấp. Bậc cao nhất của hệ thống tập tin là thư mục gốc, được ký hiệu bằng vạch chéo "/" (root directory). Đối với các hệ điều hành Unix và Linux tất cả các thiết bị kết nối vào máy tính đều được nhận dạng như các tập tin, kể

cả những linh kiện như ổ đĩa cứng, các phân vùng đĩa cứng và các ổ USB. Điều này có nghĩa là tất cả các tập tin và thư mục đều nằm dưới thư mục gốc, ngay cả những tập tin biểu tượng cho các ổ đĩa cứng.

Ví dụ, `/home/nttvinh/nguyen/scnp.odt` chỉ toàn bộ đường dẫn đến tập tin `scnp.odt` có trong thư mục `nttvinh` là thư mục phụ nằm trong thư mục `home`, ngay dưới thư mục gốc(`/`).

Nằm dưới thư mục gốc (`/`) có một loạt các thư mục quan trọng của hệ thống tập tin được công nhận ở tất cả các bản phân phối Linux khác nhau. Sau đây là danh sách các thư mục thông thường được nhìn thấy dưới thư mục gốc (`/`) :

- `/bin` – chứa các ứng dụng quan trọng (binary applications),
 - `/boot` – các tập tin cấu hình cho quá trình khởi động hệ thống (boot configuration files),
 - `/dev` – chứa các tập tin là chứng nhận cho các thiết bị của hệ thống (device files)
 - `/etc` – chứa các tập tin cấu hình của hệ thống, các tập tin lệnh để khởi động các dịch vụ của hệ thống...
 - `/home` – thư mục này chứa các thư mục cá nhân của những người có quyền truy cập vào hệ thống (local users' home directories),
 - `/lib` – thư mục này lưu các thư viện chia sẻ của hệ thống (system libraries)
 - `/lost+found` – thư mục này được dùng để lưu các tập tin không có thư mục mẹ mà được tìm thấy dưới thư mục gốc (`/`) sau khi thực hiện lệnh kiểm tra hệ thống tập tin (`fsck`).
 - `/media` – thư mục này được dùng để tạo ra các tập tin gắn (loaded) tạm thời được hệ thống tạo ra khi một thiết bị lưu động (removable media) được cắm vào như đĩa CDs, máy ảnh kỹ thuật số...
 - `/mnt` – thư mục này được dùng để gắn các hệ thống tập tin tạm thời (mounted filesystems),
 - `/opt` – thư mục dùng để chứa các phần mềm ứng dụng (optional applications) đã được cài đặt thêm,
 - `/proc` – đây là một thư mục đặc biệt linh động để lưu các thông tin về tình trạng của hệ thống, đặc biệt về các tiến trình (processes) đang hoạt động,
 - `/root` – đây là thư mục nhà của người quản trị hệ thống (root),
 - `/sbin` – thư mục này lưu lại các tập tin thực thi của hệ thống (system binaries)
 - `/sys` – thư mục này lưu các tập tin của hệ thống (system files),
 - `/tmp` – thư mục này lưu lại các tập tin được tạo ra tạm thời (temporary files),
 - `/usr` – thư mục này lưu và chứa những tập tin của các ứng dụng chính đã được cài đặt cho mọi người dùng (all users),
- `/var` – thư mục này lưu lại tập tin ghi các số liệu biến đổi (variable files) như các tập tin dữ liệu và tập tin bản ghi (logs and databases).

Câu hỏi ôn tập chương

1. Hãy trình bày khái niệm về quản lý người dùng trong linux
2. Hãy tạo hệ thống quản lý người dùng như: Tạo tài khoản người dùng, thay đổi thông tin của tài khoản, tạm khóa tài khoản, hủy tài khoản
3. Nêu các lệnh quản lý người dùng
4. Nêu các bước phân quyền trên hệ thống tập tin

TÀI LIỆU THAM KHẢO

- Nguyễn Minh Hoàng, *Linux - Giáo trình lý thuyết và thực hành*, Nxb Lao động Xã hội, 2002;
- Hà Quang Thụy – Nguyễn Trí Thành – *Giáo trình hệ điều hành Unix – Linux*, Đại học quốc gia Hà nội, 2004
- TS. Phạm Nguyên Khang – TS. Đỗ Thanh Nghị – *Giáo trình Linux và phần mềm mở*, NXB Cần Thơ, 2011
- Tham khảo Website: www.tailieuhocTap.vn