Nhập môn Linux và phần mềm mã nguồn mở

TS Hà Quốc Trung, ThS Lê Xuân Thành

Ngày 22 tháng 7 năm 2010

Phần I

Giới thiệu về phần mềm mã nguồn mở và Linux

Chương 1

Phần mềm mã nguồn mở

1.1 Khái niệm phần mềm mã nguồn mở

1.1.1 Khái niệm phần mềm tự do-mã nguồn mở

Các thao tác có thể thực hiện trên phần mềm Phần mềm là một sản phẩm trí tuệ đặc biệt, đặc trưng cho ngành CNTT và CNPM. Trên các phần mềm, có thể thực hiện các thao tác:

Sản xuất phần mềm : nghiên cứu nhu cầu NSD, thiết kế, coding, compiling and releasing.

Cài đặt phần mềm : Để có thể được sử dụng, phần mềm cần được cài đặt. Cài đặt là thao tác ghi các mã cần thiết cho việc thực hiện môi trường vào bộ nhớ thích hợp để NSD có thể sử dụng. Như vậy để cài đặt phần mềm cần có các mã máy cần thiết cho việc thực hiện phần mềm. Các mã này có thể để dưới dạng hiểu được bởi con người hoặc dưới dạng ngôn ngữ máy.

Sử dụng phần mềm : cài đặt và sử dụng phần mềm trên máy tính. Máy tính này có thể là máy tính cá nhân, máy chủ, máy tính công cộng, ... Tùy theo từng bối cảnh việc sử dụng phần mềm có thể có các ràng buộc khác nhau. (cài trên 1 máy, trên nhiều máy, trên nhiều CPU, ...). Các phần mềm có bản quyền thường bảo vệ việc sử dụng phần mềm bằng serial key, activate code và có những trường hợp bằng khóa vật lý!

Thay đổi phần mềm Trong quá trình sử dụng có thể xuất hiện nhu cầu thay đổi. Việc thay đổi này có thể được tiến hành bởi tác giả phần mềm hoặc có thể do một người khác. Để thay đổi tính năng phần mềm cần có mã nguồn của phần mềm. Nếu không có mã nguồn, có thể dịch ngược để thu được mã nguồn từ mã thực hiện. Mã nguồn phần mềm có thể được phân phối theo nhiều kênh khác nhau (mạng, lưu trữ, truyền tay, lây nhiễm)

Các thao tác khác Phân tích ngược mã nguồn, phân tích giao diện, mô phỏng, thực hiện luân phiên ...

Phần mềm được quản lý bởi các qui tắc về bản quyền và sở hữu trí tuệ, cho phép thực hiện hoặc không thực hiện các thao tác nói trên trong các điều kiện khác nhau.

Bản quyền phần mềm (BQPM) là tài liệu qui định việc thực hiện các thao tác trên phần mềm. Có thể có các bản quyền phần mềm sở hữu, bản quyền cho phần mềm miễn phí/phần mềm chia sẻ, bản quyền cho phần mềm tự do và mã nguồn mở

1.1.2 Phần mềm sở hữu

là phần mềm có bản quyền ràng buộc chặt chẽ các thao tác trên phần mềm, đảm bảo quyền lợi của người làm ra phần mềm. Copy Right (bản quyền)là thuật ngữ chỉ quyền quản lý đối với phần mềm, cho phép/không cho phép thực hiện các thao tác khác trên phần mềm. Với các phần mềm sở hữu, thông thường bản quyền có các ràng buộc chặt chẽ đảm bảo quyền lợi của người làm ra phần mềm, nhất là việc bảo lưu bản quyền khi thực hiện các thao tác trên phần mềm. Do đó, bản quyền của các phần mềm chủ sở hữu thường rất chặt chẽ về quyền phân phối và quản lý, hạn chế quyền thay đổi và cải tiến và hầu như không cho phép việc phân tích ngược mã. Một số phần mềm sở hữu còn phân biệt các quyền này cho các đối tượng sử dụng Bạn đọc có thể tham khảo trong các thỏa thuận bản quyền dành cho NSD được phân phối kèm theo các phần mềm sở hữu.

Việc ràng buộc chặt chẽ các quyền phân phối và quản lý trên phần mềm một mặt đảm bảo quyền lợi và từ đó là động lực cho người phát triển phần mềm, mặt khác hạn chế những thành phần khác hoàn thiện và bổ sung trực tiếp các tính năng, chức năng của phần mềm.

Trong thực tế, các chủ sở hữu phần mềm chỉ cung cấp 1 phần quyền sử dụng (ví dụ sử dụng trên một máy tính, không được sử dụng trên máy chủ, không được sử dụng dịch vụ kết nối từ xa để sử dụng phần mềm). Khi NSD muốn có quyền sử dụng bổ sung cần trả tiền bổ sung theo tính chất và qui mô của quyền sử dụng. Các quyền phân phối thường bị hạn chế. NSD không có quyền phân phối cho NSD khác. Để đảm bảo kỹ thuật cho các hạn chế này, các phần mềm sở hữu thường có một mô đun để xác thực và kiểm tra quyền sử dụng. Đây cũng là một lý do mà quyền thay đổi phần mềm không bao giờ được cung cấp, trừ khi chủ sở hữu có ý định chuyển đổi sở hữu của phần mềm. Quyền quản lý phần mềm có giá rất cao, có thể coi là giá trị trí tuệ của phần mềm.

1.1.3 Phần mềm tự do mã nguồn mở

Một xu hướng khác trong việc phân phối các phần mềm là không hạn chế các quyền thực hiện trên phần mềm. Hiển nhiên là các quyền quản lý phần mềm không thể không bị hạn chế, nếu không phần mềm sẽ trở thành sở hữu của một chủ thể khác có quyền hạn chế các quyền thực hiện khác của phần

mềm. Như vậy, các phần mềm này sẽ được phân phối kèm theo tất cả các quyền, trừ quyền quản lý. Các chủ thể có thể sử dụng hoàn toàn tự do phần mềm, trừ việc sử dụng quyền quản lý để áp đặt hạn chế lên các quyền còn lại. Các phần mềm được phân phối theo cách thức này gọi là phần mềm tự do. Để đảm bảo cho việc thực hiện các quyền chỉnh sửa, nâng cấp, phân tích ngược phần mềm, các phần mềm này thường được phân phối kèm với mã nguồn. Chính vì nguyên nhân này nên thuật ngữ phần mềm tự do thường được gọi là phần mềm tự do mã nguồn mở hoặc phần mềm mã nguồn mở.

Chú ý Trong khái niệm phần mềm mã nguồn mở, không qui định việc trả phí cho việc thực hiện các thao tác trên phần mềm. Điều này có nghĩa là phần mềm mã nguồn mở hoàn toàn có thể được bán, được kinh doanh giống như phần mềm sở hữu. Tất nhiên, việc NSD có trong tay mã nguồn, mã thực hiện từ một nguồn khác không mất phí có động lực để trả một khoản phí nào đó cho nhà phát triển phần mềm mang tính chất tài trợ nhiều hơn là thanh toán phí.

Chú ý Cũng liên quan đến phí của phần mềm, cần phân biệt phần mềm mã nguồn mở với các phần mềm miễn phí. Với các phần mềm miễn phí, NSD sẽ có quyền sử dụng chứ không có quyền phân phối lại, thay đổi, chỉnh sửa, ...

Chú ý Do có hạn chế về quyền quản lý phần mềm, nên phần mềm tự do mã nguồn mở khi phân phối vẫn cần kèm theo bản quyền. Bản quyền của phần mềm mã nguồn mở chỉ ra NSD có thể sử dụng bất cứ quyền nào trên phần mềm, trừ việc hạn chế bớt quyền trên phần mềm. Đây cũng là lý do bản quyền của phần mềm mã nguồn mở thường được gọi bằng thuật ngữ Copy Left thay cho Copy Right.

Ranh giới giữa quyền quản lý và các quyền khác là một ranh giới mờ, do đo khái niệm mã nguồn mở được hiểu một cách khác nhau bởi các chủ thể khác nhau, phụ thuộc vào tập hợp quyền được cung cấp. Bản quyền GPL (Global Public License) tập hợp các tiêu chí chính để một phần mềm có thể được coi là phần mềm mã nguồn mở:

- Tự do phân phối
- Luôn kèm mã nguồn
- Cho phép thay đổi phần mềm
- Không cho phép thay đổi các ràng buộc bản quyền

- Có thể có ràng buộc về việc
- Tích hợp mã nguồn
- Đặt tên phiên bản
- Không phân biệt cá nhân/nhóm khác nhau
- Không phân biệt mục đích sử dụng
- Không hạn chế các phần mềm khác
- Trung lập về công nghệ

Một số các nhà phát triển khác không coi việc phân biệt nhóm, cá nhân khác nhau, hạn chế các phần mềm khác là một đặc điểm của PMMNM. Ví vậy, trước khi sử dụng phần mềm mã nguồn mở, cần kiểm tra xem bản quyền của phần mềm mã nguồn mở này qui định những gì.Trái với suy nghĩ của nhiều NSD, PMMNM có bản quyền và có thể bị vi phạm bản quyền. Có rất nhiều trường hợp mã nguồn sau khi chỉnh sửa đã bị đóng lại.

1.2 Phát triển PMMNM

Nếu như các phần mềm sở hữu do một chủ thể duy nhất phát triển, quá trình phân tích thiết kế xây dựng phần mềm được hoạch định và kiểm soát chặt chẽ (mô hình dàn nhạc) thì PMMNM được phát triển theo mô hình chợ trời, trong đó NSD đóng vai trò của người phát triển phần mềm. Quá trình ra quyết định là động, không có một định hướng cứng nhắc từ thời điểm ban đầu. Độ tự do của nhà phát triển là rất lớn, có thể lựa chọn các quyết định theo xu hướng cá nhân, thiểu số và cũng có khi là đa số. Có rất nhiều trường hợp khi các ý kiến không thống nhất đã sinh ra 2 dòng phần mềm từ một phần mềm ban đầu trong quá trình phát triển(ví dụ iTexMac và TexShop).

Kịch bản phát triển phổ biến của PMMNM là: có một nhà phát triển đưa ra một phiên bản đầu tiên+ý tưởng về phần mềm. Các nhà phát triển khác hoàn thiện các chức năng đề ra trong ý tưởng đó, tiếp tục đề xuất tính nằng mới. Quá trình liên tục được lặp lại. Để thuận tiện hơn cho các loại NSD, các phiên bản của PMMNM thường được qui định như sau:

- Phiên bản dịch đêm: với mã nguồn được thay đổi thường xuyên, hàng ngày vào buổi đêm bản nhị phân của phiên bản mới nhất này được dịch. Phiên bản này chứa các tính năng mới nhất, tuy nhiên chưa được kiểm tra và rà soát kỹ càng, còn tiềm ẩn nhiều lỗi, chưa ổn định. Phiên bản này chủ yếu cho các nhà phát triển thử nghiệm và hoàn thiện.

- Phiên bản thử nghiệm: Đã được rà soát các lỗi, tuy nhiên vẫn chưa ổn định. Dành cho NSD thử nghiệm để có ý kiến phản hồi.
- Phiên bản bền vững: không tích hợp các tính năng chưa ổn định. Dành cho NSD đinh khai thác phần mềm.

1.3 Lịch sử phát triển PMMNM

Việc sử dụng hệ điều hành UNIX và các công cụ hỗ trợ đi kèm đã khiến cho các nhà phát triển phần mềm cảm thấy bản quyền hạn chế sự sáng tạo của họ. Năm 1983, dự án GNU (GNU is NOT UNIX) ra đời, do Richard Stallman sáng lập. Dự án này phát triển thành Tổ chức phần mềm tự do (FSF-Free Software Foundation). Tổ chức này tập hợp các nhà phát triển thường xuyên sử dụng UNIX, hướng tới mục tiêu là phát triển các công cụ tương tự như của UNIX nhưng hoàn toàn tự do và mã nguồn mở. gcc (GNU C Complier) là sản phẩm đầu tiên, cho phép phát triển các sản phẩm khác. vi là chương trình soạn thảo thông dụng, ... và rất nhiều sản phẩm khác.

Năm 1998 các nỗ lực ủng hộ PMMNM đã hình thành OSI (Open Source Initiative). OSI nỗ lực để tạo ra các khung pháp lý, cung cấp các thông tin cần thiết cho NSD, các nhà phát triển, các công ty dịch vụ có thể phát triển, khai thác, cung cấp dịch vụ, kinh doanh PMMNM.

Mặc dù có một quá trình phát triển khá lâu dài, tuy nhiên trên thực tế phải đến năm 2008 mới có những qui định chặt chẽ của pháp luật một số nước bảo hộ PMMNM. Ví dụ khi vi phạm bản quyền của phần mềm, tất cả các quyền được gán trong bản quyền lập tức trở thành vô hiệu. Qui định này không tác động nhiều đến phần mềm sở hữu, nhưng với PMMNM, khi các quyền trở thành vô hiệu hầu như chắc chắn NSD sẽ vi phạm các sở hữu trí tuệ.

1.4 Nguồn lực phát triển phần mềm mã nguồn mở

Khái niệm PMMNM không ràng buộc việc phần mềm có thể được bán hay không, tuy nhiên, với việc cung cấp kèm theo mã nguồn và cho phép NSD có thể tùy ý sửa đổi, việc thu một khoản phí từ NSD với các PMMNM không có cơ sở hợp lý, trừ những trường hợp rất đặc biệt khi phần mềm chỉ phục vụ cho số lượng ít NSD nào đó. Việc phát triển phần mềm, cho dù là sở hữu hay tự do, đều cần có nguồn lực về con người, tài chính. Câu hỏi đặt ra là

làm thế nào để thu hút được nguồn lực để phát triển một PMMNM nào đó. Có thể liệt kê một số cách thức để thu hút các nguồn lực.

Tư vấn Nguồn lực để phát triển mã nguồn mở có thể thu được từ các đơn vị chịu trách nhiệm tư vấn cho tổ chức sử dụng cuối cùng. Việc làm chủ được các PMMNM, các giải pháp sử dụng chúng cho phép các chuyên gia về PMMNM có thể tư vấn hiệu quả cho các tổ chức để lựa chọn các giải pháp, để quản lý kỹ thuận hệ thống thông tin của mình.

Hỗ trợ kỹ thuật Nắm vững mã nguồn và cách thức khai thác PMMNM cho phép cung cấp dịc vụ hỗ trợ kỹ thuật cho các tổ chức không chuyên về IT.

Đào tạo Khi các giải pháp PMMNM được sử dụng rộng rãi, sẽ xuất hiện nhu cầu về nhân lực phát triển, khai thác các PMMNM. Những công ty đi trước có thể cung cấp các dịch vụ đào tạo, dịch vụ cấp chứng chỉ để đáp ứng nhu cầu này.

Cung cấp các giải pháp mã nguồn mở Không chỉ cung cấp các PMMNM, hoàn toàn có thể cung cấp các giải pháp tích hợp một hoặc nhiều PMMNM để đáp ứng nhu cầu chung về phần mềm của một tổ chức. Người cung cấp dịch vụ có thể không phải là người phát triển phần mềm, mà chỉ là người tích hợp các PMMNM khác lại với nhau, tuy nhiên đã cấu hình các PMMNM này để có hiệu năng tối ưu, có giao diện thuận tiện, ... nói chung là đáp ứng yêu cầu của NSD.

Tài trợ/quảng cáo Khi một tổ chức cần một phần mềm, tổ chức này có thể tự phát triển phần mềm, có thể mua một phần mềm khác, có thể tài trợ cho một nhóm các nhà phát triển PMMNM. Nếu một số tổ chức có cùng nhu cầu về một phần mềm, các tổ chức này còn phối hợp với nhau, tài trợ các nguồn lực (con người, tài chính, cơ sở vật chất) để xây dựng một PMMNM, chia sẻ bớt kinh phí phát triển phần mềm. PMMNM không bị hạn chế về quyền sử dụng và phân phối, do đó có số lượng NSD lớn. Hoàn toàn có thể sử dụng lợi thế này để tạo nguồn kinh phí từ quảng cáo trên phần mềm hoặc trên các thông tin liên quan đến phần mềm. Có nhiều trường hợp có 2 phiên bản của phần mềm: phiên bản MNM tuân thủ GPL nhưng hạn chế về chức năng, phiên bản sở hữu (hoặc chuyên nghiệp) có đầy đủ các tính năng. Có thể thấy phiên bản MNM sẽ đóng vai trò quảng cáo cho phiên bản đầy đủ/chuyên nghiệp.

Thương mại hóa (một phần/tất cả) Một cách thức nữa để có nguồn lực phát triển là sau một thời gian phát triển PMMNM có thể tiến hành

thương mại hóa phần mềm để thu hồi chi phí. Tuy nhiên, hiệu quả của việc này phụ thuộc vào chất lượng của phần mềm có thuyết phục được NSD đang dùng phiên bản MNM chuyển sang phiên bản thu phí.

1.5 So sánh phần mềm mã nguồn mở và phần mềm mã nguồn đóng

Tồn tại nhiều ý kiến ủng hộ và không ủng hộ xu hướng phát triển PMMNM. Các ý kiến ủng hộ cho rằng:

- PM MNM có thể phát triển theo nhu cầu NSD
- Không bị giới hạn sự sáng tạo
- Tin cậy và bảo mật: Mã nguồn được đông đảo NSD kiểm tra.
- Giảm chi phí phát triển
- Không bị cản trở bởi động lực kinh tế

Các ý kiến không ủng hộ tập trung chủ yếu vào một số luận điểm

Triệt tiêu động lực phát triển Việc xuất hiện các phần mềm mã nguồn mở làm cho không còn động lực để phát triển phần mềm nói chung.

Thiếu tính chuyên nghiệp Do PMMNM do nhiều người cùng tham gia phát triển, do đó khó có thể kiểm soát được qui trình phát triển và chất lượng của phần mềm. Chính vì thế nên PMMNM khó có thể thuyết phục được NSD không chuyên về IT

Không bảo mật Mã nguồn công khai cho tất cả NSD, kể cả những NSD muốn tấn công hệ thống.

1.6 Một số phần mềm mã nguồn mở thông dụng

Phần mềm mã nguồn mở hiện nay đã đạt đến mức phát triển ổn định, các lỗi cơ bản được khắc phục, được NSD chấp nhận rộng rãi. Có thể kể ra một vài phần mềm/bộ phần mềm được sử dụng rộng rãi hiện nay là:

FireFox Trình duyệt của Mozilla, cho phép có thể phát triển các plug-in bổ sung

Open Office Bộ soạn thảo văn bản của Sun Micro System, có thể thay thế MS Office

Apache Web server được sử dụng rộng rãi

PHP-MySQL Application Server

Thunder Bird Mail Client của Mozilla

Unikey Chương trình gõ tiếng Việt

1.6.1 Kho dữ liệu PMMNM

PMMNM có thể được tải về theo cách thông thường như với các phần mềm miễn phí hoặc chia sẻ. Kho dữ liệu sourceforge.net định nghĩa khung thông tin cần thiết để cập nhật các thông tin chi tiết về một dự án PMMNM.

Để thuận tiện cho việcsử dụng mã nguồn, các mã nguồn theo phiên bản của các phần mềm được lưu trữ tại các kho phần mềm. Các kho phần mềm này cho phép NSD tải mã nguồn và cập nhật mã nguồn mởi. Các sản phẩm thường được sử dụng là:

CVS Concurrent Versions System: Hệ thống cho phép lưu trữ mã nguồn, kiểm soát các thay đổi trong mã nguồn và kiểm soát phân nhánh khi cần thiết

SVN Hệ thống kiểm soát mã nguồn và quá trình chỉnh sửa mã nguồn, thay thế CVS.

Chương 2

Khái niệm Linux

2.1 Linux: Nhân, hệ điều hành, bản phân phối hay hệ thống

Thuật ngữ Linux được sử dụng rộng rãi trong thực tế. Tuy nhiên, trong các ngữ cảnh khác nhau, thuật ngữ này có thể được hiểu với nghĩa khác nhau.

Khi sử dụng Linux trên các thiết bị nhúng, thiết bị di động, trong trường hợp này, chỉ có nhân của HĐH Linux được sử dụng. Thuật ngữ Linux được dùng để chỉ nhân của hệ điều hành Linux. Nhân của HĐH bao gồm các phần mềm cần thiết để quản lý và sử dụng các phần cứng của hệ thống.

Khi cài đặt các phần mềm trên máy tính, có thể có nhiều lựa chọn: Windows, Linux, Sun, MacOS. Trong ngữ cảnh này. Linux được hiểu là một Hệ Điều Hành.

Linux thường được phân phối không phải chỉ gồm có nhân và các phần mềm hệ thống. Đi kèm theo HĐH, còn có các phần mềm ứng dụng phục vụ các nhu cầu của từng lớp NSD. Giao diện đồ họa, các chương trình hỗ trợ. ... Tất cả các thành phần đó kết hợp với nhân của hệ điều hành và HĐH tạo ra một bản phân phối của Linux. Có rất nhiều bản phân phối khác nhau, có thể sử dụng chung một phiên bản của nhân, một tập hợp chung các phần mềm hệ thống, nhưng được phân phối với các bộ phần mềm khác nhau dành cho máy chủ, máy để bàn, máy xách tay, ...

Khi so sánh hiệu năng giữa các sản phẩm Linux, MacOS, Sun Solaris, ... thực tế ta đã so sánh các hệ thống trên đó các HĐH tương ứng đã được cài đặt. Linux trong trường hợp này được sử dụng để chỉ một hệ thống máy tính cài đặt hệ điều hành Linux.

Khi sử dụng thuật ngữ Linux, cần xác định rõ bỗi cảnh để tránh hiểu nhầm.

2.2 Lịch sử phát triển của Linux

Linux ra đời dựa trên một số yếu tố lịch sử đặc biệt.

Hệ điều hành Unix Thế hệ thứ nhất của các máy tính lớn chủ yếu sử dụng hệ điều hành Unix. Đây là một hệ điều hành được viết và sử dụng ngôn ngữ lập trình C. Nhược điểm duy nhất của HĐH Unix là giá thành cao. Với sự ra đời của các máy tính cá nhân, nhu cầu về một hệ điều hành đa nhiệm, đa NSD giá thành tương xứng với máy tính ngày càng trở nên mạnh mẽ. HĐH DOS của IBM và Microsoft đáp ứng được nhu cầu về giá thành, tuy nhiên lại là đơn nhiệm.

FSF-GNU Hurd Các nỗ lực của FSF hướng tới mục tiêu là viết lại các công cụ của Linux để có thể phổ biến chúng dưới GPL. Một trong các dự án đó hướng tới việc xây dựng một HĐH mã nguồn mở có tên là GNU-Hurd. Rất tiếc, dự án này đã bị đóng băng và không có một HĐH mã nguồn mở nào có tên là Hurd.

Andrew Tanenbaum là một giáo sư tại trường Đại học Vrije của Hà Lan. Ông là tác giả của rất nhiều cuốn sách kinh điển có giá trị trong CNTT. Một trong những cuốn sách đó là cuốn "Hệ Diều Hành", xuất bản lần đầu tiên năm 1987. Cuốn sách này mô tả chi tiết hoạt động của một HĐH hiện đại, đa nhiệm, đa NSD. Đặc biệt, cuốn sách này được cung cấp kèm theo mã nguồn của HĐH Minix, điều này cho phép bạn đọc có thể tự mình kiểm nghiệm các tính năng của HĐH. Sau khi cuốn sách được phổ biến, việc có thể tìm hiểu, nghiên cứu, thử nghiệm một HĐH chi tiết đã tạo cho các SV cơ hội "viết lại" các HĐH. Một số trong đó có thể phát triển ra những phiên bản HĐH mới. Việc xuất hiện các máy tính cá nhân càng làm cho việc thử nghiệm HĐH trở nên dễ dàng. Tuy nhiên, Tanenbaum chỉ cho phép SV phân phối mã nguồn ban đầu của Minix, không cho phép họ phân phối mã nguồn đã được sửa đổi. Do đó các đóng góp của nhiều SV sẽ không được tích hợp lại.

Một trong những sinh viên của Trường, Linux Tovard đã hoàn thành một nhân hệ điều hành có tính năng gần giống với nhân HĐH Linux. Linux Tovald thay vì giữ sản phẩm cho riêng mình, đã công bố mã nguồn cho cộng đồng các nhà phát triển (1991). Đó là sự xuất hiện của nhân Linux 1.0. Hiện tại nhân Linux đang có phiên bản 2.6 bền vững. Được sự đóng góp của cộng đồng, nhân của HĐH Linux đã trở nên ổn định, có thể chạy trên rất nhiều máy tính khác nhau, phục vụ nhiều loại nhu cầu khác nhau của NSD như dùng máy tính để bàn, server, ...

Tanenbaum đã cho rằng Linux không có tương lai phát triển. Ông cho rằng Linux được thiết kế theo mô hình nhân khối (tất cả các tính năng của nhân được tích hợp vào trong một mã duy nhất) trong khi mô hình phù hợp là mô hình vi nhân. Một số ý kiến khác thì cho rằng Linux đã sử dụng phần lớn mã từ mã của Minix. Tanenbaum tuy có những ý kiến phản biện, nhưng cũng khẳng định là mã Linux được phát triển từ đầu.

Với một nhân HĐH hoạt động ổn định (1993), các công cụ hỗ trợ từ Unix được xây dựng bởi các dự án FSF cho phép NSD có một HĐH tương đối đầy đủ, thuận tiện cho NSD, chi phí thấp hơn nhiều so với Unix. Có rất nhiều các phiên bản của nhân Linux, do đó nhân Linux được thống nhất đánh số theo dạng X.Y.Z-D:

- X: thế hệ. Hiện tại có thế hệ 1 và 2.

- Y: phiên bản chính
- Z: phiên bản phụ
- Các số lẻ là phiên bản thử nghiệm, số chẵn là phiên bản bền vững
- D: Phần còn lại bổ sung bởi các nhà phân phối

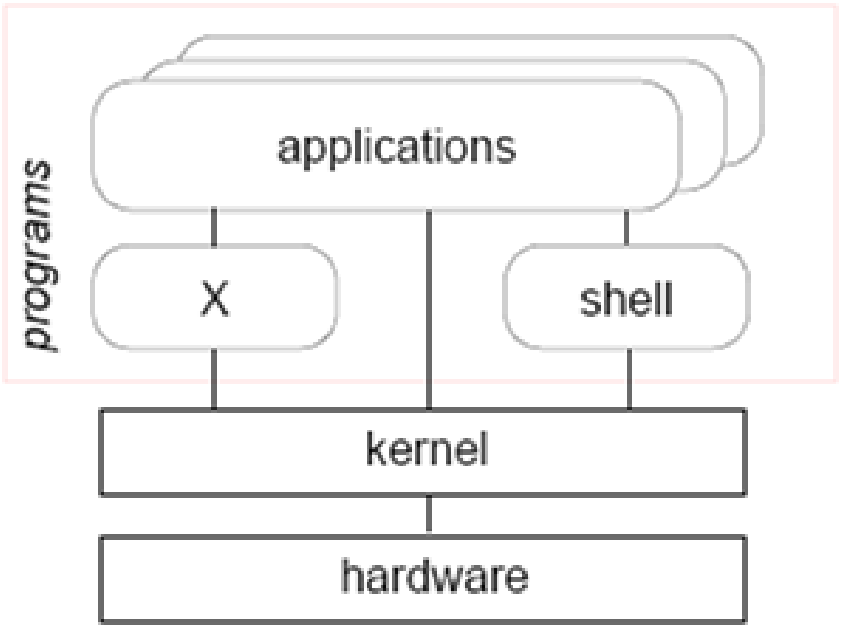
Ví dụ 2.6.31-14-generic-pae biểu diễn nhân Linux thế hệ 2, phiên bản 6, phiên bản phụ 31. Phần bổ sung bởi nhàn phân phối thường là tên nhà phân phối và dòng máy tính phù hợp.

2.3 Ứng dụng của Linux

Mặc dù ban đầu chỉ được xây dựng cho các máy tính i386, tuy nhiên do tính chất mã nguồn mở, NSD với phần cứng hoặc nhu cầu khác nhau đều có thể thay đổi Linux cho phù hợp, nên Linux có thể được ứng dụng trong nhiều lĩnh vực khác nhau, trên các dòng phần cứng khác nhau.

Máy tính để bàn Linux được sử dụng ngày càng nhiều trên máy tính để bàn. Linux ngày nay thường được phân phối cùng với các giao diện đồ họa như GNOME, KDE, ... Các phần mềm ứng dụng xuất hiện ngày càng nhiều, hoạt động ổn định, cung cấp cho NSD những công cụ mạnh mẽ để xử lý văn bản, chỉnh sửa đồ họa, duyệt Internet, .. tóm lại tất cả các thao tác mà người sử dụng mong chờ ở một máy tính để bàn. Các chương trình phổ biến trên các HĐH thương mại Windows và MacOSX hầu hết đều có các phần mềm có tính năng tương đương trên Linux.

Máy chủ Linux được sử dụng phổ biến hơn trên các máy chủ. Một máy tính Linux có thể được kết nối và thực hiện các thao tác quản trị máy tính thông qua một giao diện văn bản. Việc truy cập vào giao diện console này nhanh và thuận tiện hơn nhiều so với truy cập vào giao diện đồ họa. Linux có thị phần vượt trội và có xu hướng tăng đân trong thị trường máy chủ. Điều này có thể giải thích dựa trên tổ hợp LAMP (Linux-Apache-PHP-MySQL) rất thuận tiện cho việc triển khai các web site và ứng dụng web. Trên các máy tính lớn, Linux cũng được dùng phổ biến bởi 2 nguyên nhân: giá thành rẻ và tính tương thích tương tự Unix. Các siêu máy tính hầu hết được thiết kế để có thể hoạt động với Unix, nên có thể hoạt động dễ dàng trên Linux. Một số siêu máy tính còn được phân phối cùng Linux. Máy tính IBM Sequoi cũng sẽ sử dụng HĐH Linux.



Hình 2.4.1: Các thành phần của Linux

Các hệ nhúng Linux còn được sử dụng rộng rãi trên các thiết bị nhúng vì khả năng tùy biến và giá thành hạ. HĐH Maemo mà Nokia sẽ sử dụng trong một loạt các điện thoại thông minh sắp ra đời là một HĐH dựa trên Linux. Trong nhiều router, switch cũng sử dụng HĐH Linux.

2.4 Các thành phần của Linux

HĐH Linux gồm các thành phần: nhân hệ điều hành, các công cụ hệ thống, giao diện đồ họa và ứng dụng. Hình 2.4.1 mô tả các thành phần của Linux.

Nhân hệ điều hành Nhân hệ điều hành cung cấp một giao diện cho các chương trình và NSD có thể quản lý và khai thác phần cứng máy tính. Nhân HĐH bao gồm các driver của một số phần cứng cơ bản, các chương trình lập lịch CPU, quản lý bộ nhớ và quản lý các thiết bị vào ra.

Các drivers Ngoài các phần cứng cơ bản của các hệ thống máy tính, còn nhiều các phần cứng khác được quản lý bởi các driver chưa được tích hợp trong nhân. Các driver này có thể được tải cùng với nhân hoặc sau khi nhân

đã được tải. Việc một driver được tích hợp vào trong nhân hay đặt dưới dạng một mô đun hoàn toàn do người dịch nhân quyết định.

Các phần mềm hệ thống Các phần mềm liên quan đến cấu hình hệ thống, giám sát hệ thống, thực hiện các thao tác quản trị

Các phần mềm ứng dụng Các phần mềm ứng dụng cho NSD: bộ soạn thảo, mail client, trình biên dịch, thông dịch, ...

X Windows và Các phần mềm ứng dụng với giao diện đồ họa Giao diện đồ họa được xây dựng trên cơ sở X, phần mềm cho phép quản lý các vùng logic của màn hình theo các cửa sổ. Trên nền của X, có các chương trình quản lý cửa sổ như KDE, GNOME cho phép quản lý các cửa sổ một cách thống nhất. NSD có thể sử dụng các chương trình chạy trên nền đồ họa của X.

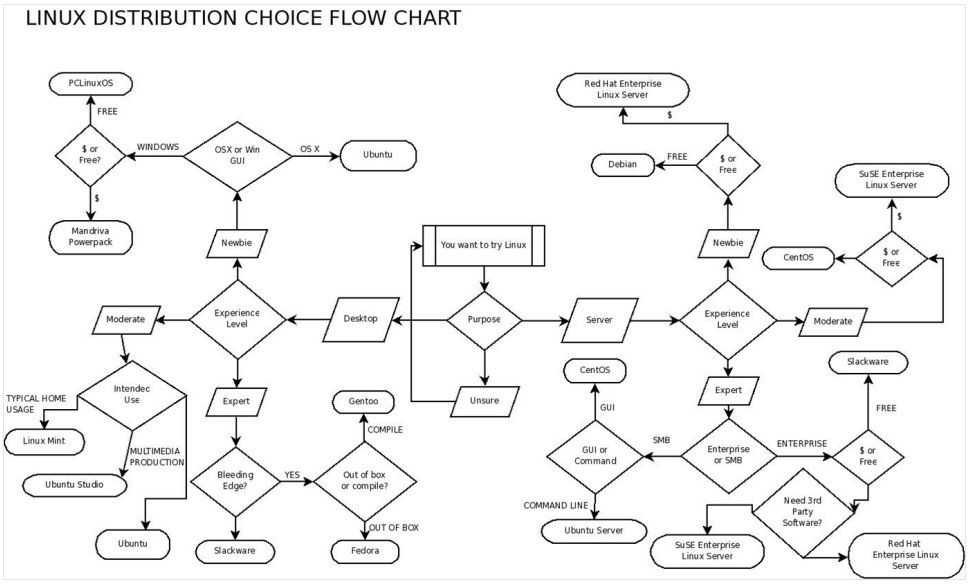
2.5 Các bản phân phối của Linux

Linux được các nhà phân phối đóng gói để tạo điều kiện thuận lợi cho NSD. Ban đầu việc đóng gói đơn giản chỉ là tích hợp các phần mềm cần thiết vào một ổ lưu trữ ngoài như đĩa cứng, đĩa mềm. Dần dần, khái niệm bản phân phối là một hoặc nhiều phương tiện lưu trữ có thể tạo ra một môi trường làm việc tương đối đầy đủ cho NSD, chỉ cần NSD khai báo các thông tin cơ bản về hệ thống chứ không cần phải can thiệp vào các chi tiết của quá trình cài đặt. Một bản phân phối hiện đại thường gồm các thành phần sau: nhân HĐH, các gói phần mềm cơ bản, công cụ quản lý các phần mềm, công cụ hỗ trợ cài đặt.

2.5.1 Các thành phần của một bản phân phối Linux

Nhân hệ điều hành Nhà phân phối lựa chọn một phiên bản của nhân Linux, chỉnh sửa, đóng gói để làm nền tảng cho bản phân phối.

Các gói phần mềm cơ bản Linux được phân phối kèm theo các gói phần mềm. Số lượng các gói phần mềm lên đến cỡ hàng ngàn, do đó các nhà phân phối thường phân chia các gói phần mềm thành các nhóm phần mềm phục vụ các yêu cầu khác nhau: nhóm công cụ quản trị hệ thống, nhóm công cụ phát triển, nhóm công cụ xử lý văn bản, nhóm các công cụ đồ họa, ... để thuận tiện hơn cho người sử dụng cho việc lựa chọn.



Hình 2.5.2: Lựa chọn các bản phân phối của Linux

Công cụ quản lý phần mềm Với số lượng phần mềm lớn, không tránh khỏi có xung đột và ràng buộc lẫn nhau. Các bản phân phối thường sử dụng các công cụ chuyên biệt, có CSDL riêng để có thể quản lý các xung đột và ràng buộc này. Có 2 công cụ được sử dụng phổ biến: Redhat package manager và Debian Package Manager. Các bản phân phối khác thường dựa trên một trong 2 công cụ này.

Công cụ hỗ trợ cài đặt Là công cụ hỗ trợ NSD trong quá trình cài đặt, nhận thông tin về nhu cầu của NSD và thực hiện các thao tác cài đặt chi tiết thay cho NSD. Các công cụ được các bản phân phối hiện tại thường cho phép NSD thực hiện các tùy chọn: Chạy thử nghiệm từ CD, không cần cài đặt, cài đặt với giao diện đồ họa, cài đặt kiểu quản trị viên, thực hiện chế độ phục hồi. Các chương trình cài đặt này cho phép NSD có thể dễ dàng có một hệ thống Linux phù hợp với nhu cầu sử dụng của mình.

2.5.2 Lựa chọn bản phân phối Linux phù hợp

Mỗi bản phân phối thường hướng đến một đối tượng sử dụng cụ thể. Người sử dụng có thể lựa chọn bản phân phối phù hợp với nhu cầu sử dụng của mình. Hình **??** đề xuất một số bản phân phối phù hợp theo nhu cầu của NSD.

Phần II

Quản trị hệ thống Linux

Chương 3

Các thao tác cơ bản

3.1 Cài đặt linux

3.1.1 Lựa chọn cách thức cài đặt

Bạn đọc có thể cài đặt Linux bằng một trong các phương án sau:

- Cài từ bộ cài trên đĩa CD
- Cài qua mạng từ đĩa Net Install
- Cài từ bộ cài trên máy
- Cài thông qua một máy chủ Linux trong mạng nội bộ bằng NFS
- Cài từ bộ cài trong file .iso (dùng máy ảo)

Bạn đọc cũng có thể lựa chọn cài đặt hệ thống Linux trên:

- Máy tính đã cài sẵn hệ điều hành Windows và chạy song song hai hệ điều hành
- Máy tính chỉ cài Linux
- Máy ảo

Trên các phương án trên, tác giả khuyến cáo bạn đọc nên cài Linux trên máy ảo và cài từ bộ cài đầy đủ trên file iso. Phương án này không làm ảnh hưởng đến hệ thống hiện tại đang hoạt động của bạn đọc, không cần phải ghi bộ cài ra đĩa CD và dễ dàng sao chép, sử dụng lại sau này.

3.1.2 Các bước chuẩn bị trước khi cài đặt

Để tiến hành cài đặt hệ thống bạn đọc cần các công việc sau:

- Tải về máy bộ cài linux dưới dạng file iso
- Tải về máy và cài đặt phần mềm máy ảo VirtualBox cho Windows bản mới nhất tại:
- Chuẩn bị các thông tin về cấu hình địa chỉ IP cho máy tính
- Tham khảo các bước cài đặt cụ thể trên mạng Internet

3.1.3 Các lưu ý trong quá trình cài đặt

Tạo máy ảo bằng VitualBox . Máy ảo này chính là hệ thống để bạn đọc tiếp cận với Linux trong những phần sau nên không cần phần cứng quá mạnh. Trong bước lựa chọn bộ nhớ RAM, dung lượng 256Mb là đủ. Trong bước lựa chọn ổ đĩa cứng, dung lượng 8Gb và tùy chọn "Dynamically expanding storage" (dùng đến đâu file ổ cứng lớn lên đến đó) là một lựa chọn thông minh. Sau khi kết thúc phần khởi tạo, bạn đọc có thể tinh chỉnh một chút cấu hình hệ thống như sau: Cấu hình card mạng nên để ở chế độ NAT để máy vẫn vào mạng được và không ảnh hưởng đến hệ thống mạng nội bộ đang dùng. Bạn đọc cũng có thể remove phần ổ đĩa mềm đi để hệ thống nhanh hơn chút ít. Trỏ ổ đĩa CD vào file .iso mà bạn đã tải về.

Tiến hành cài đặt . Sau khi cho chạy máy ảo hệ thống sẽ boot từ đĩa CD ảo và bắt đầu cài đặt hệ thống như đối với máy thật thông thường. Trong quá trình này tác giả lưu ý với bạn đọc vài điểm sau:

- Phần lựa chọn ngôn ngữ cài đặt nên để tiếng anh mỹ (US) tránh trường hợp một số phần mềm không tương thích với hệ điều hành tiếng việt
- Phần lựa chọn quốc gia nên chọn Việt Nam để có thể chọn đúng được múi giờ ở phần lựa chọn múi giờ hệ thống
- Phần lựa chọn ổ cứng nên để mặc định (cài tất cả cây thư mục lên một ổ cứng). Việc tùy biến ổ cứng nên để sau này khi bạn đọc đã thành thạo Linux và xác định rõ mình cài đặt Linux cho một hệ thống cụ thể nào đó.
- Phần đặt mật khẩu cho tài khoản "root" cần được lưu ý tuân thủ các quy tắc sau: lớn hơn 10 ký tự, kết hợp chữ hoa, thường với ký tự số hoặc ký tự đặc biệt để bảo mật
- Không được bỏ qua phần tạo thêm một tài khoản người sử dụng khác vì bạn đọc sẽ chủ yếu dùng tài khoản này

3.2 Đăng nhập

3.2.1 Khởi động hệ thống

Sau khi bấm nút bật máy tính, hệ thống được khởi động. Sau quá trình POST của máy tính, trình khởi động hệ thống (hiện nay thường là GRUB) sẽ đưa ra danh sách các cách khởi động hệ điều hành, bạn đọc chọn chế độ

muốn khởi động hoặc để hệ thống chạy mặc định. Trong quá trình khởi chạy hệ điều hành màn hình sẽ hiện lên rất nhiều thông tin nhưng ta không xem xét đến vấn đề này (sau này, khi đã thành thạo và làm quen với Linux nâng cao bạn đọc sẽ có thể tìm hiểu được chúng), cuối cùng sẽ xuất hiện màn hình cho người dùng đăng nhập. Ở đây chúng ta không cài đặt chế độ đồ họa nên màn hình đăng nhập sẽ hiện như sau: (hình ảnh minh họa).

3.2.2 Đăng nhập vào hệ thống

Khi máy tính hiện ra màn hình như trên, bạn đọc cần nhập tên tài khoản người dùng và mật khẩu để đăng nhập vào hệ thống. Nếu trong quá trình cài đặt người dùng đã tạo tài khoản người dùng thì nhập tên truy cập và mật khẩu của tài khoản này để đăng nhập. Còn nếu đã bỏ qua việc tạo một tài khoản sử dụng thì ở lần đăng nhập này cần phải đăng nhập bằng tài khoản "root". Bạn đọc nhập tên truy cập là "root"và mật khẩu đã tạo trong quá trình cài đặt để đăng nhập vào. Tác giả khuyến cáo bạn đọc không nên sử dụng thường xuyên đến tài khoản "root" này vì đây là tài khoản có quyền cao nhất của hệ thống, nhưng thao tác không cẩn thận sẽ làm hỏng hệ thống bất cứ lúc nào. Nếu người dùng gõ sai mật khẩu hệ thống sẽ hiện ra thông báo dưới đây để người dùng nhập lại (hình minh họa). Bạn đọc lưu ý kiểm tra các phím số, caplock... trước khi nhập mật khẩu để tránh gõ nhầm, gõ sai. Nếu gõ đúng phiên làm viêc sẽ được thiết lập với màn hình sử dụng và dấu nhắc hệ thống như hình dưới (hình minh họa) Linux còn hỗ trợ người dùng đăng nhập thông qua thiết bị giao tiếp (console). Có hai kiểu console đó là:

Console với giao diện dòng lệnh (CLI - Command Line Interface) Ở kiểu đăng nhập này một trình thông dịch lệnh được tự động kích hoạt khi phiên làm việc, trình thông dịch này giúp hệ thống tương tác với người sử dụng thông qua các câu lệnh. Người sử dụng sẽ nhập lệnh bằng bàn phím và kết quả đưa ra trên màn hình dưới dạng văn bản như hình minh họa dưới đây. Kiểu giao diện sử dụng này hoạt động dựa trên một ngôn ngữ lập trình dạng kịch bản (script) và tiêu tốn cực ít tài nguyên của hệ thống nên rất thích hợp khi người quản trị tương tác với hệ thống từ xa.

Console với giao diện đồ họa (GUI - Graphic User Interface) . Khi đăng nhập bằng giao diện đồ họa, chế độ cửa sổ (windows) được kích hoạt, người sử dụng sử dụng hệ thống với giao diện đồ họa như trên các máy sử dụng hệ điều hành Windows bình thường. Kiểu giao tiếp này tiêu tốn nhiều tài nguyên hơn tuy nhiên lại dễ sử dụng, thích hợp với những người quản trị

hệ thống không chuyên (làm việc kiêm nhiệm thêm do thiếu người và không được đào tạo bài bản).

3.3 Giao diện dòng lệnh CLI

3.3.1 Terminal và console ảo

Trước khi tìm hiểu về giao diện dòng lệnh, tác giả sẽ trình bày hai khái niệm rất hay dùng trong Linux là Terminal và Console.

Terminal Khái niệm Terminal xuất hiện từ xa xưa khi các hệ thống máy tính rất lớn, người sử dụng không tương tác trực tiếp với hệ thống mà thông qua các Terminal ở xa. Các hệ thống Terminal này gồm màn hình và bàn phím, ngày nay do kích thước bé đi nên các terminal này chính là máy tính của người sử dụng.

Console Ngoài ra hệ thống Linux nói chung hay các máy chủ dịch vụ của các hệ điều hành khác nói riêng đều cung cấp cho người quản trị một giao diện Terminal đặc biệt gọi là Console. Trước kia console tồn tại dưới dạng một cổng giao tiếp riêng biết, còn ngày nay nó xuất hiện trong các hệ thống Linux dưới dạng một Console ảo cho phép mở cùng lúc nhiều phiên làm việc trên một máy tính. Để chọn console ảo người sử dụng có thể ấn tổ hợp phím <Ctrl>+<Alt>+<F*> - F* chạy từ F1 - F8 - Trong đó <Ctrl>+<Alt>+<F7> là danh cho chế độ đăng nhập bằng đồ họa. Để thoát khỏi console nào người sử dụng gõ lệnh "exit" hoặc tổ hợp phím <Ctrl>+<D>

3.3.2 Shell

Shell là trình thông dịch dòng lệnh trong hệ thống Linux. Mỗi khi người sử dụng đăng nhập vào hệ thống bằng giao điện dòng lệnh tức là người sử dụng khởi động một shell để sử dụng. Nếu người sử dụng mở thêm một hay vài console ảo cũng tức là người sử dụng mở thêm một vài shell mới và khi thoát khỏi các Console này tức là đã tắt shell đó đi. Chú ý dấu nhắc của shell "root" là ký tự <thăng> còn tất cả các shell khác là ký tự "đô la" Mỗi Shell làm các nhiệm vụ sau:

- Thông dịch và thực hiện các lệnh
- Soạn thảo dòng lênh, lịch sử các lệnh đã gõ

- Scripting
- Quản lý tác vụ

Bạn đọc cũng có thể tìm hiểu thêm thông tin về các Shell thông dụng sau: sh, csh, tcsh, ksh, zsh...

3.4 Một số lệnh thường dùng

3.4.1 Cấu trúc dòng lệnh trong Linux

Một lệnh trong Linux được cấu tạo bởi 3 phần như sau: <Command> [Options]> [Arguments] Trong đó:

- <Command>: tên của lệnh, giúp cho hệ thống biết được: hệ thống sẽ làm gì

Option : tham số của lệnh (thường được truyền vào bằng cách thêm vào sau dấu <+> hoặc <->), giúp hệ thống hiểu được: hệ thống sẽ làm việc đó như thế nào

Argument : đối số của lệnh, giúp hệ thống biết được: hệ thống sẽ thực hiện lệnh này ở đâu (trên thư mục nào hay trên file nào)

Có một vài lưu ý cho bạn đọc khi sử dụng lệnh trong Linux như sau:

- Câu lệnh trong Linux phân biệt chữ hoa và chữ thường
- Giữa các phần của câu lệnh phải có khoảng trắng
- Đằng sau các dấu <+> và <-> không được có khoảng trắng
- Các phần đặt trong dấu ngoặc vuông [] là không bắt buộc, không có lệnh vẫn thực hiện được.

Hình minh họa ví dụ

3.4.2 Các phím tắt để sửa lỗi

Trong quá trình gõ lệnh, sẽ có lúc bạn đọc gõ sai cú pháp lệnh hay nhầm lẫn khi truyền vào các tùy chọn và tham số, tuy có thể gõ lại lệnh mới hoàn toàn nhưng đối với những lệnh dài, sửa chữa lại chỗ gõ nhầm là phương án hợp lý hơn. Ở dưới đây tác giả đưa ra một bảng với các tổ hợp phím thường sử dụng trong quá trình sửa chữa lệnh, người dùng cũng có thể tự tích lũy

thêm trong quá trình sử dụng của riêng mình. Chèn hình hay bảng mình họa Có một lưu ý với người sử dụng là hệ thống Linux hỗ trợ một cơ chế tự động hoàn thiện lệnh giúp giảm thiểu sai sót của người sử dụng bằng cách sử dụng phím <Tab>. Ví dụ: thay vì phải gõ đầy đủ lệnh <mkdir> người sử dụng gõ <mkd> và gõ thêm <Tab>. Điều này cũng đúng khi người dùng gõ phần <Arguments> của lênh. Ví dụ thay vì gõ <cat /etc/passwd> người sử dụng có thể gõ <cat /e> + <Tab> + <pas> + <Tab>. Bạn đọc nên tập thói quen sử dụng phím này ngay cả với các lệnh ngắn như cat, more, tail....

3.4.3 Nhóm các lệnh xem thông tin hệ thống

man : Lệnh xem help của hệ thống, được sử dụng để xem hướng dẫn sử dụng lệnh hay thông tin về các tiện ích của hệ thống. Cú pháp: <man [option] [-S section] command name/utility name>. Ví dụ: <man ls> hay <man gzip>.

Info : Lệnh này tương tự như lệnh Man nhưng có một ưu điểm là nội dung đưa ra dưới dạng siêu văn bản (như website) nên người dùng có thể xem các phần khác nhau của trợ giúp mà không cần thoát ra ngoài, vì thế người dùng hoàn toàn có thể bật info lên ở một console ảo và làm việc ở một console ảo khác, khi cần tra cứu lệnh thì chuyển sang.

date : Lệnh hiển thị thông tin về ngày tháng, lệnh này sẽ hiển thị chi tiết thông tin về ngày giờ hiện tại. Để biết thêm thông tin về các <option> của lệnh này người sử dụng có thể sử dụng lệnh <man date>

logname : lệnh hiển thị tên người sử dụng đang ở phiên làm việc. Trong quá trình cài đặt hệ thống bạn đọc đã trải qua bước tạo một tài khoản để sử dụng, trong quá trình đó có 3 bước khai báo: Khai báo họ tên người dùng tài khoản, khai báo tên đăng nhập tài khoản và khai báo mật khẩu. Lệnh này sẽ hiển thị thông tin về Họ tên người sử dụng.

hostname : Lệnh này sẽ hiển thị tên của trạm làm việc. Nếu bạn đọc sử dụng lệnh này trên chính máy cài đặt linux thì tên của máy Linux sẽ được hiển thị

ps : Lệnh hiển thị các tiến trình đang chạy. Sử dụng lệnh này với tham số <-f> bạn đọc sẽ xem được cả thông tin về user khởi chạy các tiến trình đó

3.4.4 Nhóm lệnh về tài khoản người dùng

useradd : lệnh tạo tài khoản người dùng. Cú pháp: <useradd username [PathToHomeDirectory]>. Nếu bạn đọc không đưa vào đường dẫn đến thư mục chứa tài khoản, thư mục này sẽ được thiết lập mặc định tại </home/TenUser>

passwd : lệnh thay đổi mật khẩu. Cú pháp <passwd [username]>. Nếu chỉ gõ <passwd> mà không thi hệ thống hiểu là đổi mật khẩu cho tài khoản hiện tại đang đăng nhập. Sau khi gõ lệnh hệ thống sẽ đưa ra thông báo yêu cầu gõ mật khẩu mới 2 lần để người sử dụng gõ vào mật khẩu mới.

3.4.5 Nhóm lệnh thao tác với file và thư mục

ls : lệnh dùng để liệt kê thông tin về file và thư mục chứa trong thư mục muốn xem. Cú pháp <ls [option] [PathToDirectory]>. Ví dụ: <ls -l /var> sẽ hiển thị thông tin về các file và thư mục trong thư mục </var>

mkdir Lệnh tạo thư mục, cú pháp: <mkdir TenThuMuc>. Ví dụ <mkdir tailieu> hay <mkdir /var/tailieu>, lệnh đầu tiên sẽ tạo thư mục tên là <tailieu> ở thư mục làm việc hiện hành, lệnh thứ 2 sẽ tạo thư mục <tailieu> trong thư mục </var>

cp Lệnh sao chép, được dùng để sao chép thư mục hoặc file. Cú pháp <cp Nguồn Đích>, với nguồn là thư mục hoặc file cần copy và đích là nơi cần đặt thư mục hoặc tài liệu vào. Chú ý rằng các thao tác với thư mục cần có option là <-r>. Ví dụ: <cp congvan.txt /var/tailieu> hoặc <cp -r /var/tailieu /etc>, ví dụ đầu sẽ sao chép file congvan.txt ở thư mục hiện hành vào thư mục </var/tailieu>, ví dụ thứ hai sẽ sao chép thư mục </var/tailieu> vào thư mục </etc>

mv Lệnh di chuyển, dùng để di chuyển thư mục hoặc file. Lệnh này có cú pháp và cách sử dụng giống hệt lệnh <cp> tuy nhiên nó không sao chép mà di chuyển file hoặc thư mục đến đích. Bạn đọc có thể vận dụng hai lệnh <cp> và <mv> một cách khéo léo để đổi tên thư mục ví dụ như sau: <mv tailieu.txt vanban.txt> lệnh này sẽ đổi tên file <tailieu.txt> ở thư mục hiện hành thành <vanban.txt>, hoặc <cp congvan.txt /var/tailieu/giaygioithieu.doc> lệnh này sẽ copy file <vanban.txt> ở thư mục hiện hành đến thư mục </var/tailieu> và đổi tên file đó thành <giaygioithieu.doc>

rm Lệnh xóa, dùng để xóa file và thư mục, cú pháp và cách sử dụng giống hai lệnh <cp> và <mv>. Để thao tác lên thư mục bạn đọc nhớ thêm <-r>. Ngoài ra còn có một lệnh xóa thư mục nữa nhưng lệnh này chỉ có tác dụng xóa các thư mục rỗng <rmdir TenThuMuc>.

touch Lệnh tạo file, lệnh này sẽ tạo một file rỗng, người sử dụng thêm nội dung vào sau. Cú pháp <touch TenFile>, ví dụ <touch baitap.doc> ví dụ này tạo một file rỗng tên là <baitap.doc> trong thư mục hiện hành

echo Lệnh này cho phép ghi một thông tin vào file, cú pháp <echo NoiDung > TenFile> hoặc <echo NoiDung » TenFile>. Lệnh đầu tiên sẽ xóa nội dung file và ghi đè nội dung mới vào, lệnh thứ hai sẽ ghi tiếp nội dung mới và cuối file

3.5 Bài tập

Bài tập 3.1 *Cài đặt hệ điều hành*

- *Cài Debian với giao diện dòng lệnh. Yêu cầu: Cài đặt, cấu hình địa chỉ IP, cập nhật hệ điều hành lên bản mới nhất*
- *Cài Ubuntu với giao diện đồ họa. Yêu cầu: Cài đặ, cấu hình địa chỉ IP, cập nhật hệ điều hành và cài phần mềm bằng cách tải gói về*

Bài tập 3.2 *Thực hành các lệnh cơ bản Đưa vào sau*

Chương 4

Hệ thống tệp

4.1 Khái niệm hệ thống tệp

Các hệ thống máy tính sử dụng các thiết bị lưu trữ ngoài để lưu trữ thông tin một cách bền vững. Các thiết bị lưu trữ quản lý không gian bộ nhớ ngoài theo từng khối dữ liệu. Giữa các khối dữ liệu chỉ có liên quan về mặt vật lý, không có liên quan gì về ngữ nghĩa. Để có thể sử dụng các khối dữ liệu này một cách thuận tiện, các khối dữ liệu có chung ngữ nghĩa, có chung mục đích sử dụng được gộp lại với nhau và được quản lý bởi một khối dữ liệu điều khiển. Các khối dữ liệu được gộp lại như vậy gọi là một tệp (file). Khi người sử dụng có nhiều tệp, để có thể quản lý các tệp dễ dàng hơn, các tệp được gộp lại với nhau theo yêu cầu của NSD, bổ sung thêm một tệp chứa danh mục và vị trí của các tệp được gộp. Tệp chứa danh mục này được gọi là tệp thư mục. Về phần mình, tệp thư mục cũng có thể được gộp vào với các tệp khác để tạo thành thư mục. Với cách nhóm các tệp như vậy, trong hệ thống sẽ có 2 loại tệp cơ bản:

- Tệp thông thường chỉ chứa dữ liệu.
- Tệp thư mục chỉ chứa danh mục các tệp và thư mục con nằm trong thư mục đó.

Các tệp và các thư mục kết hợp với nhau tạo ra một hoặc nhiều cây thư mục, trong đó các tệp thông thường luôn luôn là các nút lá. Nút gốc của các cây là điểm cố định để từ đó có thể truy cập được các nút trong cây. Ở dưới HĐH Linux, các tệp và thư mục tạo thành một cây duy nhất có thư mục gốc ký hiệu là / - (thư mục gốc). Các thư mục con thường gặp của thư mục gốc là các thư mục:

- /bin : thư mục tệp chương trình cơ bản
- /boot : thư mục chứa hạt nhân của HĐH
- /etc : thư mục các tệp cấu hình
- /dev : thư mục các tệp thiết bị
- /home : thư mục chứa dữ liệu NSD
- /lib : thư viện hệ thống
- /usr : thư mục ứng dụng
- /var : thư mục dữ liệu cập nhật
- /proc : thư mục chứa các dữ liệu của nhân hệ điều hành và BIOS

Các tệp thư mục lưu trữ các thư mục con và tệp. Các thư mục con và tệp đều được đặt tên. Giống như trong HĐH Windows, Linux cho phép tên tệp có thể dài đến 255 ký tự, có thể bao gồm các ký tự đặc biệt.

Để truy cập được vào các thư mục và tệp, xuất phát từ nút gốc truy cập vào các thư mục con cho đến khi đến được tệp cần thiết. Tập hợp tên của các thư mục con từ nút gốc đến tệp cần truy cập, phân cách các tên bằng dấu / gọi là đường dẫn tuyệt đối đến tệp. Trong mọi trường hợp, luôn luôn có thể dùng đường dẫn tuyệt đối để tham chiếu tới tệp.

Khi NSD truy cập vào hệ thống hoặc khi các chương trình đang thực hiện, một thư mục được sử dụng để tham chiếu tới tất cả các tệp và thư mục khác trong hệ thống. Với NSD đó thường là thư mục nhà (/home, đượctham chiếu tới bằng /). Với chương trình, đó thường là thư mục gọi câu lệnh thực hiện. Thư mục này gọi là thư mục làm việc hiện tại.

Trong một thư mục luôn luôn có 2 thư mục đặc biệt: ./ để biểu diễn thư mục hiện tại và ../ biểu diễn thư mục cha của thư mục hiện tại.

Trong nhiều trường hợp, sẽ hiệu quả hơn nếu truy cập vào một tệp thông qua đường đi trong cây từ thư mục hiện tại đến tệp cần truy cập bằng cách sử dụng ./ và ../. Một đường dẫn như vậy sẽ phụ thuộc vào thư mục làm việc hiện tại, được gọi là đường dẫn tương đối.

4.2 Quản lý thư mục

Các câu lệnh thực hiện thao tác cơ bản để làm việc với các thư mục là:

- cd (change directory) Chuyển đến một thư mục.
- pwd Xác định thư mục hiện tại:

```
trunghq@ubuntu:~/temp$ whoami
trunghq
trunghq@ubuntu:~/$ pwd
/home/trunghq
trunghq@ubuntu:~/$ cd temp (or cd ./temp)
trunghq@ubuntu:~/temp$ pwd
/home/trunghq/temp
trunghq@ubuntu:~/temp$ cd .. (or cd ../)
trunghq@ubuntu:~/$ pwd
/home/trunghq
```

- ls Liệt kê nội dung của thư mục.

```
drwxr-xr-x 2 trunghq trunghq 4096 2010-04-15 22:40 .
drwxr-xr-x 5 trunghq trunghq 4096 2010-04-15 22:40 ..
trunghq@ubuntu:~/temp$ ls -la
total 8
drwxr-xr-x 2 trunghq trunghq 4096 2010-04-15 22:40 .
drwxr-xr-x 5 trunghq trunghq 4096 2010-04-15 22:40 ..
trunghq@ubuntu:~/temp$ mkdir d1
trunghq@ubuntu:~/temp$ ls -la
total 12
drwxr-xr-x 3 trunghq trunghq 4096 2010-04-15 22:42 .
drwxr-xr-x 5 trunghq trunghq 4096 2010-04-15 22:40 ..
drwxr-xr-x 2 trunghq trunghq 4096 2010-04-15 22:42 d1
```

- mkdir Tạo ra thư mục mới khi thư mục cha đã tồn tại. Trường hợp muốn tạo tự động thư mục cha sử dụng -p.

```
trunghq@ubuntu:~/temp$ mkdir d2/d3
mkdir: cannot create directory ‘d2/d3’: No such file or directory
trunghq@ubuntu:~/temp$ mkdir -p d2/d3
trunghq@ubuntu:~/temp$ ls -la
total 16
drwxr-xr-x 4 trunghq trunghq 4096 2010-04-15 22:42 .
drwxr-xr-x 5 trunghq trunghq 4096 2010-04-15 22:40 ..
drwxr-xr-x 2 trunghq trunghq 4096 2010-04-15 22:42 d1
drwxr-xr-x 3 trunghq trunghq 4096 2010-04-15 22:42 d2
trunghq@ubuntu:~/temp$ ls -la d2/
total 12
drwxr-xr-x 3 trunghq trunghq 4096 2010-04-15 22:42 .
drwxr-xr-x 4 trunghq trunghq 4096 2010-04-15 22:42 ..
drwxr-xr-x 2 trunghq trunghq 4096 2010-04-15 22:42 d3
```

- rmdir Xóa một thư mục cũ với điều kiện là thư mục rỗng.

```
trunghq@ubuntu:~/temp$ rmdir d1
trunghq@ubuntu:~/temp$ ls -la
total 12
drwxr-xr-x 3 trunghq trunghq 4096 2010-04-15 22:53 .
drwxr-xr-x 5 trunghq trunghq 4096 2010-04-15 22:40 ..
drwxr-xr-x 3 trunghq trunghq 4096 2010-04-15 22:42 d2
trunghq@ubuntu:~/temp$ rmdir d2
rmdir: failed to remove ‘d2’: Directory not empty
```

```
trunghq@ubuntu:~/temp$ rmdir -p d2/d3
trunghq@ubuntu:~/temp$ ls -la
total 8
drwxr-xr-x 2 trunghq trunghq 4096 2010-04-15 22:54 .
drwxr-xr-x 5 trunghq trunghq 4096 2010-04-15 22:40 ..
```

4.3 Quản lý tệp

4.3.1 Các loại tệp

Linux phân biệt 9 loại tệp. Kiểu của tệp được hiển thị khi sử dụng câu lệnh ls -la, biểu diễn bằng chữ cái đầu tiên trong hàng. Tệp thông thường được biểu diễn bằng ký hiệu -. Thư mục được biểu diễn bằng ký hiệu d. Các thiết bị cũng được biểu diễn bằng tệp. Các tệp ứng với các thiết bị trao đổi thông tin theo ký tự (character device) được biểu diễn bằng ký hiệu c. Các thiết bị trao đổi thông tin theo khối (block device) được biểu diễn bằng b. Khi trao đổi thông tin với nhau, các tiến trình sử dụng socket. Các socket cũng được biểu diễn bằng các tệp , ký hiệu bằng s. Đường ống được biểu diễn bằng p (pipeline). Các liên kết biểu tượng được biểu diễn bằng l. Các tệp biểu diễn một vùng bộ nhớ ký hiệu bằng m.

```
[trunghq@localhost trunghq]$ ls -la
total 24
drwx------    2 trunghq  trunghq      4096 Apr  7 13:17 .
drwxr-xr-x    3 root     root         4096 Apr  7 09:29 ..
-rw-------    1 trunghq  trunghq      1930 Apr 14 08:39 .bash_history
-rw-r--r--    1 trunghq  trunghq        24 Apr  7 09:29 .bash_logout
-rw-r--r--    1 trunghq  trunghq       191 Apr  7 09:29 .bash_profile
-rw-r--r--    1 trunghq  trunghq       124 Apr  7 09:29 .bashrc
[trunghq@localhost trunghq]$ ls -la /dev/sda1
brw-rw----    1 root     disk       8,   1 Jan 30  2003 /dev/sda1
[trunghq@localhost trunghq]$ ls -la /dev/tty1
crw--w----    1 root     tty        4,   1 Apr 16 18:55 /dev/tty1
[trunghq@localhost trunghq]$ ls -la /etc/init.d/network
-rwxr-xr-x    1 root     root         6784 Feb  4  2003 /etc/init.d/network
[trunghq@localhost trunghq]$ ls -la /etc/rc3.d/S
S05kudzu       S13portmap     S25netfs       S80sendmail    S95atd
S08iptables    S14nfslock     S26apmd        S85gpm         S97rhnsd
S09isdn        S17keytable    S28autofs      S90crond       S99local
S10network     S20random      S55sshd        S90xfs
S12syslog      S24pcmcia      S56rawdevices  S95anacron
```

```
[trunghq@localhost trunghq]$ ls -la /etc/rc3.d/S12syslog
lrwxrwxrwx    1 root     root           16 Apr  2 19:34 /etc/rc3.d/S12syslog ->.
./init.d/syslog
```

4.3.2 Các thao tác trên tệp

Để tạo ra một tệp mới, có thể dùng một chương trình soạn thảo như vi, nano, pico, ... hoặc có thể dùng lệnh touch. Một cách khác để tạo ra một tệp là sử dụng câu lệnh echo > tentep. Ví dụ:

```
[trunghq@localhost trunghq]$ vi test.txt
[trunghq@localhost trunghq]$ cat test.txt
Tao tep bang vi
[trunghq@localhost trunghq]$ touch test1.txt
[trunghq@localhost trunghq]$ ls test*
test1.txt  test.txt
[trunghq@localhost trunghq]$ echo "Tao tep bang echo" >test2.txt
[trunghq@localhost trunghq]$ cat test2.txt
Tao tep bang echo
```

Lệnh rm cho phép có thể xóa một hoặc nhiều tệp. Để có thể xóa đệ qui thư mục với các thư mục con, sử dụng câu lệnh rm -Rf. Câu lệnh này có thể thay thế câu lệnh rmdir. Chú ý, nếu không có tùy biến -r, sẽ không xóa được thư mục.

```
 [trunghq@localhost trunghq]$ mkdir d1
 [trunghq@localhost trunghq]$ mkdir d1/d1
 [trunghq@localhost trunghq]$ rm d1
 rm: cannot remove ‘d1’: Is a directory
 [trunghq@localhost trunghq]$ rm -Rf d1
 [trunghq@localhost trunghq]$ ls
 test1.txt  test2.txt  test.txt
 [trunghq@localhost trunghq]$
```

Việc sao chép và dịch chuyển tệp/thư mục được thực hiện bằng câu lệnh mv và cp. Có thể thực hiện việc sao chép và dịch chuyển nhiều tệp, thư mục.

```
[trunghq@localhost trunghq]$ pwd
/home/trunghq
[trunghq@localhost trunghq]$ ls -la temp1
ls: temp1: No such file or directory
```

```
[trunghq@localhost trunghq]$ mkdir temp1
[trunghq@localhost trunghq]$ touch temp1/t1
[trunghq@localhost trunghq]$ touch temp1/t2
[trunghq@localhost trunghq]$ touch temp1/t3
[trunghq@localhost trunghq]$ touch temp1/t4
[trunghq@localhost trunghq]$ touch temp1/t5
[trunghq@localhost trunghq]$ nkdir temo
-bash: nkdir: command not found
[trunghq@localhost trunghq]$ mkdir temp
[trunghq@localhost trunghq]$ ls -la
total 40
drwx------    4 trunghq  trunghq      4096 Apr 18 07:17 .
drwxr-xr-x    3 root     root         4096 Apr  7 09:29 ..
-rw-------    1 trunghq  trunghq      2170 Apr 16 20:24 .bash_history
-rw-r--r--    1 trunghq  trunghq        24 Apr  7 09:29 .bash_logout
-rw-r--r--    1 trunghq  trunghq       191 Apr  7 09:29 .bash_profile
-rw-r--r--    1 trunghq  trunghq       124 Apr  7 09:29 .bashrc
drwxrwxr-x    2 trunghq  trunghq      4096 Apr 18 07:17 temp
drwxrwxr-x    2 trunghq  trunghq      4096 Apr 18 07:17 temp1
-rw-rw-r--    1 trunghq  trunghq         0 Apr 16 19:28 test1.txt
-rw-rw-r--    1 trunghq  trunghq        18 Apr 16 19:28 test2.txt
-rw-rw-r--    1 trunghq  trunghq        16 Apr 16 19:27 test.txt
[trunghq@localhost trunghq]$ mv temp1 temp
[trunghq@localhost trunghq]$ ls temp/
temp1
[trunghq@localhost trunghq]$ ls temp/temp1
t1  t2  t3  t4  t5

#Copy thư mục cần dùng -R

[trunghq@localhost trunghq]$ mkdir temp2
[trunghq@localhost trunghq]$ cp temp/temp1 temp2
cp: omitting directory ‘temp/temp1’
[trunghq@localhost trunghq]$ cp -R temp/temp1 temp2
[trunghq@localhost trunghq]$ ls temp2/temp1/
t1  t2  t3  t4  t5
[trunghq@localhost trunghq]$
```

Để đọc nội dung của tệp, có thể sử dụng nhiều câu lệnh khác nhau. Bảng dưới đây mô tả các câu lệnh thường dùng

Câu lệnh	Mô tả
cat	Hiển thị tệp từ đầu đến cuối
tac	Hiển thị tệp từ cuối đến đầu
head -n	Hiển thị n dòng đầu của tệp
tail -n	Hiển thị n dòng cuối của tệp
wc	Đếm số từ và số dòng của tệp
more	Hiển thị tệp theo trang
less	Hiển thị tệp theo dòng

Bảng 4.1: Một số câu lệnh hiển thị tệp

4.4 inode và liên kết

4.4.1 Khái niệm inode

Trong mục trước, cấu trúc hệ thống tệp logic đã được đề cập. Mỗi tệp sẽ được xác đinh bởi tên và đường dẫn tuyệt đối. Tuy nhiên để truy cập các tệp trong Linux, cần có một ánh xạ giữa các địa chỉ logic này và vị trí của các khối dữ liệu trên ổ cứng. Các thông tin về vị trí, về quyền truy cập, về chủ sở hữu của một tệp được lưu trữ trong cấu trúc dữ liệu đặc biệt gọi là inode. Việc kết hợp giữa không gian địa chỉ logic và các inode cần đảm bảo hiệu quả của việc truy cập nội dung của các tệp. Trong các phân vùng, mỗi inode có một mã số riêng. Số lượng các inode trong một phân vùng được giới hạn khi định dạng phân vùng.

Để có thể xem số thứ tự inode của các tệp, sử dụng câu lệnh ls -lia

```
[trunghq@localhost trunghq]$ ls -lia
total 40
 113635 drwx------    4 trunghq  trunghq      4096 Apr 18 07:19 .
 452541 drwxr-xr-x    3 root     root         4096 Apr  7 09:29 ..
 113639 -rw-------    1 trunghq  trunghq      2411 Apr 18 09:00 .bash_history
 113636 -rw-r--r--    1 trunghq  trunghq        24 Apr  7 09:29 .bash_logout
 113637 -rw-r--r--    1 trunghq  trunghq       191 Apr  7 09:29 .bash_profile
 113638 -rw-r--r--    1 trunghq  trunghq       124 Apr  7 09:29 .bashrc
 113661 drwxrwxr-x    3 trunghq  trunghq      4096 Apr 18 07:18 temp
 549879 drwxrwxr-x    3 trunghq  trunghq      4096 Apr 18 07:19 temp2
 113652 -rw-rw-r--    1 trunghq  trunghq         0 Apr 16 19:28 test1.txt
 113654 -rw-rw-r--    1 trunghq  trunghq        18 Apr 16 19:28 test2.txt
 113653 -rw-rw-r--    1 trunghq  trunghq        16 Apr 16 19:27 test.txt
```

Để có thể truy cập được vào một tệp, cần phải biết được số thứ tự inode của

tệp đó. Số thứ tự này được Linux lưu trong nội dung của thư mục chứa tệp. Để đọc được nội dung của thư mục, cần biết được inode của tệp thư mục. Inode này được lưu trữ trong nội dung của thư mục cha. Như vậy quá trình tìm inode của một tệp là quá trình đệ qui, với điểm xuất phát là inode của thư mục gốc.

Để quá trình truy cập tệp có thể được thực hiện nhanh hơn, nếu vị trí hiện tại và tệp cần truy cập tương đối gần nhau, có thể sử dụng các tên . và .. cho thư mục hiện tại và thư mục cha. Các tệp được truy cập bằng đường dẫn tương đối sẽ lấy mốc xuất phát là thư mục hiện tại, có thể không cần đi qua nút gốc, giảm số bước truy cập vào các tệp trung gian.

4.4.2 Liên kết vật lý

Có nhiều trường hợp một tệp được sử dụng trong các bối cảnh khác nhau, cần sử dụng 2 tên tuyệt đối khác nhau để truy cập vào tệp. Trường hợp này chỉ cần tạo ra một bản ghi trong thư mục mới, trong đó chứa inode của tệp cũ. Bản ghi này được gọi là một liên kết vật lý đến tệp, được tạo ra bởi câu lệnh ln

```
[trunghq@localhost temp1]$ ln t1 ../t1-link
[trunghq@localhost temp1]$ ls -la t1
-rw-rw-r--    2 trunghq  trunghq         0 Apr 18 07:17 t1
[trunghq@localhost temp1]$ ls -la ../t
t1-link  temp1
[trunghq@localhost temp1]$ ls -la ../t1-link
-rw-rw-r--    2 trunghq  trunghq         0 Apr 18 07:17 ../t1-link
[trunghq@localhost temp1]$ ls -lia t1
 113656 -rw-rw-r--    2 trunghq  trunghq         0 Apr 18 07:17 t1
[trunghq@localhost temp1]$ ls -lia ../t1-link
 113656 -rw-rw-r--    2 trunghq  trunghq         0 Apr 18 07:17 ../t1-link
```

Câu lệnh ls -lia còn cho biết số liên kết vật lý trỏ vào inode đang xem xét.

Sau khi tạo ra, liên kết vật lý và tên tệp ban đầu có ý nghĩa hoàn toàn như nhau, không phân biệt được đâu là liên kết và đâu là tệp liên kết. Mọi thay đổi trên liên kết đều có tác động đến tệp.

```
[trunghq@localhost temp1]$ echo "Thi nghiem lan 1" >t1
[trunghq@localhost temp1]$ cat ../t1-link
Thi nghiem lan 1
[trunghq@localhost temp1]$ echo "Thi nghiem lan 2" > ../t1-link
[trunghq@localhost temp1]$ cat t1
Thi nghiem lan 2
```

Khi một trong các liên kết được xóa đi, hệ thống chỉ xóa bản ghi trong thư mục tương ứng. Trường hợp không còn bản ghi nào trỏ vào inode, inode được đánh dấu là đã bị xóa, nội dung của tệp sẽ bị ghi đè khi hệ thống cần không gian ổ đia.

```
[trunghq@localhost temp1]$ ln t1 ../t1-link-02
[trunghq@localhost temp1]$ ln t1 ../t1-link-03
[trunghq@localhost temp1]$ ls -lia t1
 113656 -rw-rw-r--    4 trunghq  trunghq        17 Apr 18 23:44 t1
[trunghq@localhost temp1]$ rm ../t1-link-01
rm: cannot lstat '../t1-link-01': No such file or directory
[trunghq@localhost temp1]$ rm ../t1-link-02
[trunghq@localhost temp1]$ ls -lia t1
 113656 -rw-rw-r--    3 trunghq  trunghq        17 Apr 18 23:44 t1
[trunghq@localhost temp1]$ rm ../t1-link-03
[trunghq@localhost temp1]$ ls -lia t1
 113656 -rw-rw-r--    2 trunghq  trunghq        17 Apr 18 23:44 t1
[trunghq@localhost temp1]$ rm ../t1-link
[trunghq@localhost temp1]$ ls -lia t1
 113656 -rw-rw-r--    1 trunghq  trunghq        17 Apr 18 23:44 t1
[trunghq@localhost temp1]$ rm t1
[trunghq@localhost temp1]$ ls -lia t1
ls: t1: No such file or directory
```

Trong Linux, dùng câu lệnh ln không tạo được liên kết cứng đến các thư mục. Tuy nhiên mặc định khi tạo ra một thư mục, luôn luôn có 2 thư mục con được tạo ra một cách mặc định là . và .. Như vậy một thư mục tối thiểu phải có 2 liên kết cứng trỏ vào, cứ thêm 01 thư mục con sẽ có thêm 01 liên kết cứng nữa

```
[trunghq@localhost temp]$ ls -lia temp*
total 8
 113655 drwxrwxr-x    2 trunghq  trunghq      4096 Apr 19 00:17 .
 113661 drwxrwxr-x    3 trunghq  trunghq      4096 Apr 19 00:16 ..
[trunghq@localhost temp]$ ls -lia temp1
total 8
 113655 drwxrwxr-x    2 trunghq  trunghq      4096 Apr 19 00:17 .
 113661 drwxrwxr-x    3 trunghq  trunghq      4096 Apr 19 00:16 ..

[trunghq@localhost temp]$ cd temp1/
[trunghq@localhost temp1]$ ls -la
total 8
```

```
drwxrwxr-x    2 trunghq  trunghq      4096 Apr 19 00:17 .
drwxrwxr-x    3 trunghq  trunghq      4096 Apr 19 00:16 ..
[trunghq@localhost temp1]$ mkdir t1
[trunghq@localhost temp1]$ ls -la
total 12
drwxrwxr-x    3 trunghq  trunghq      4096 Apr 19 00:18 .
drwxrwxr-x    3 trunghq  trunghq      4096 Apr 19 00:16 ..
drwxrwxr-x    2 trunghq  trunghq      4096 Apr 19 00:18 t1
```

Việc sử dụng liên kết cứng có thể tạo ra các bối cảnh, trong đó nội dung của tệp có thể bị thay đổi mà NSD không hề hay biết. Chính vì vậy liên kết cứng không được khuyến khích sử dụng.

4.4.3 Liên kết biểu tượng

Một cách đơn giản và an toàn hơn để truy cập vào các tệp từ các bối cảnh khác nhau là sử dụng liên kết biểu tượng. Liên kết biểu tượng là các tệp có nội dung chứa đường dẫn tới tệp đích. Khi truy cập vào tệp này, hệ thống sẽ tự động truy cập vào tệp đích. Các liên kết biểu tượng được xếp vào tệp loại l. Đường dẫn tới tệp đích có thể là tương đối hay tuyệt đối theo vị trí của liên kết biểu tượng. Chú ý là khi tạo liên kết biểu tượng bằng câu lệnh ls -s, shell sẽ tự động điền đường dẫn tương đối theo vị trí gõ lệnh chứ không theo vị tí của liên kết, nên có khả năng đường dẫn được kiểm tra bởi shell sẽ không trỏ đến tệp đích hợp lệ của liên kết biểu tượng.

```
[trunghq@localhost temp]$ touch temp1/test1
[trunghq@localhost temp]$ echo "Thu lien ket bieu tuong" temp1/test1
Thu lien ket bieu tuong temp1/test1
[trunghq@localhost temp]$ echo "Thu lien ket bieu tuong" > temp1/test1
[trunghq@localhost temp]$ ln -s temp1/test1 slink-01
[trunghq@localhost temp]$ cat slink-01
Thu lien ket bieu tuong
[trunghq@localhost temp]$ ls -la
total 16
drwxrwxr-x    4 trunghq  trunghq      4096 Apr 19 00:44 .
drwx------    4 trunghq  trunghq      4096 Apr 18 07:19 ..
lrwxrwxrwx    1 trunghq  trunghq        11 Apr 19 00:44 slink-01 -> temp1/test1
drwxrwxr-x    3 trunghq  trunghq      4096 Apr 19 00:43 temp1
drwxrwxr-x    2 trunghq  trunghq      4096 Apr 19 00:43 temp2
[trunghq@localhost temp]$ ls -l
total 8
lrwxrwxrwx    1 trunghq  trunghq        11 Apr 19 00:44 slink-01 -> temp1/test1
```

Tùy biến	Mô tả
-name	Tìm kiếm theo tên tệp
-iname	Tìm kiếm theo tên tệp, không phân biện hoa thường
-atime n	Tìm kiếm các tệp đã được truy cập n ngày trước
-amin n	Tìm kiếm các tệp đã được truy cập n phút trước
-mtime n	Tìm kiếm các tệp đã được thay đổi n ngày trước
-mmin n	Tìm kiếm các tệp đã được thay đổi n phút trước
-ctime n	Tìm kiếm các tệp đã được thay đổi thuộc tính n ngày trước
-cmin n	Tìm kiếm các tệp đã được thay đổi thuộc tính n phút trước

Bảng 4.2: Một số tham số chính của lệnh find

```
drwxrwxr-x    3 trunghq  trunghq      4096 Apr 19 00:43 temp1
drwxrwxr-x    2 trunghq  trunghq      4096 Apr 19 00:43 temp2
[trunghq@localhost temp]$ ls -a
.  ..  slink-01  temp1  temp2
[trunghq@localhost temp]$ ln -s temp1/test1 ./tempslink-01
temp1  temp2
[trunghq@localhost temp]$ ln -s temp1/test1 ./temp2/slink-01
[trunghq@localhost temp]$ ls -la temp2/
total 8
drwxrwxr-x    2 trunghq  trunghq      4096 Apr 19 00:45 .
drwxrwxr-x    4 trunghq  trunghq      4096 Apr 19 00:44 ..
lrwxrwxrwx    1 trunghq  trunghq        11 Apr 19 00:45 slink-01 -> temp1/test1
```

4.5 Chỉ số và tìm kiếm tệp

Để tìm kiếm một tệp nào đó trên hệ thống linux, có các công cụ sau: find, locate.

4.5.1 Câu lệnh find

Câu lệnh find cho phép duyệt các tệp có các tiêu chí phù hợp trong một thư mục và các thư mục con, đồng thời thực hiện các thao tác định nghĩa trước trên các tệp đó. Bảng mô tả một số tiêu chí để có thể tìm kiếm tệp dùng lệnh find.

```
[root@localhost trunghq]# find / -name 'passwd'
/etc/passwd
/etc/pam.d/passwd
```

```
/usr/bin/passwd
/usr/share/doc/nss_ldap-202/pam.d/passwd
/usr/share/doc/pam_krb5-1.60/krb5afs-pam.d/passwd
/usr/share/doc/pam_krb5-1.60/pam.d/passwd
Hoặc
[root@localhost trunghq]# find / -name 'passw*'
/etc/passwd
/etc/pam.d/passwd
/etc/passwd-
/etc/passwd.lock
/usr/bin/passwd
/usr/share/doc/nss_ldap-202/pam.d/passwd
/usr/share/doc/pam_krb5-1.60/krb5afs-pam.d/passwd
/usr/share/doc/pam_krb5-1.60/pam.d/passwd
/usr/share/doc/samba-2.2.7a/docs/pam_smbpass/samples/password-mature
/usr/share/doc/samba-2.2.7a/docs/pam_smbpass/samples/password-migration
/usr/share/doc/samba-2.2.7a/docs/pam_smbpass/samples/password-sync
/usr/share/pixmaps/password.png
/usr/share/man/man1/passwd.1.gz
/usr/share/man/man5/passwd.5.gz
/usr/share/awk/passwd.awk
```

Một ví dụ về tìm kiếm nhiều tiêu chí

```
[root@localhost trunghq]# find / -name 'passwd' -or -size +10000k
/proc/kcore
/var/lib/rpm/Packages
/var/log/lastlog
/etc/passwd
/etc/pam.d/passwd
/root/test2.img
/usr/bin/passwd
/usr/lib/locale/locale-archive
/usr/share/doc/nss_ldap-202/pam.d/passwd
/usr/share/doc/pam_krb5-1.60/krb5afs-pam.d/passwd
/usr/share/doc/pam_krb5-1.60/pam.d/passwd
/usr/share/comps/i386/hdlist2
/test.img
/test2.img
```

Thực hiện các câu lệnh trên kết quả tìm kiếm

```
[trunghq@localhost trunghq]$ sudo time find / -size +10000k -exec ls -l {} \;
-r--------    1 root     root     536875008 Apr 19 03:37 /proc/kcore
-rw-r--r--    1 rpm      rpm      11350016 Apr  2 19:52 /var/lib/rpm/Packages
-r--------    1 root     root     19136220 Apr 19 03:23 /var/log/lastlog
-rw-r--r--    1 root     root     51200000 Apr 14 07:05 /root/test2.img
-rw-r--r--    1 root     root     30301680 Apr  2 19:33 /usr/lib/locale/locale-a
rchive
-rw-r--r--    1 root     root     19066344 Mar 14  2003 /usr/share/comps/i386/hd
list2
-rw-r--r--    1 root     root     512000000 Apr 14 08:24 /test.img
-rw-r--r--    1 root     root     51200000 Apr 14 07:07 /test2.img
0.22user 0.54system 0:00.77elapsed 98%CPU (0avgtext+0avgdata 0maxresident)k
0inputs+0outputs (1503major+361minor)pagefaults 0swaps
```

Câu lệnh find duyệt trên cả cây thư mục, nên tốc độ tìm kiếm thường chậm. Để có thể tìm kiếm nhanh hơn vị trí của một tệp, câu lệnh locate được sử dụng:

```
[trunghq@localhost trunghq]$ time locate gpasswd
/usr/bin/gpasswd
/usr/share/man/man1/gpasswd.1.gz
/usr/share/man/pt_BR/man1/gpasswd.1.gz
/usr/share/man/fr/man1/gpasswd.1.gz
/usr/share/man/hu/man1/gpasswd.1.gz
/usr/share/man/it/man1/gpasswd.1.gz
/usr/share/man/ja/man1/gpasswd.1.gz
/usr/share/man/pl/man1/gpasswd.1.gz

real    0m0.031s
user    0m0.010s
sys     0m0.030s
```

Câu lệnh locate sử dụng CSDL để đánh chỉ số các tệp, do đó quá trình tệp nhanh hơn nhiều. Tuy nhiên trước khi sử dụng locate cần khởi tạo CSDL và cập nhật thường xuyên bằng lệnh updatedb. Câu lệnh slocate thực hiện chức năng của locate một cách bảo mật, không cho các NSD khác có thể tìm kiếm trong CSDL chỉ số của các tệp.

4.6 Bài tập

Bài tập 4.1 *Trong thư mục nhà, hãy tạo ra các thư mục /d1/d2/d3. Đổi tên d2 thành d22. Xóa thư mục d3.*

Bài tập 4.2 *Viết lệnh copy thư mục /etc/ về thư mục /root/backup/etc. Tạo liên kết đến thư mục này đặt tên là etc-backup đặt trong thư mục nhà của root.*

Bài tập 4.3 *Viết câu lệnh để có thể hiển thị số thư mục con của /etc/.*

Bài tập 4.4 *Trong thư mục nhà, tạo thư mục /temp/d1 và /temp/d2. Trong thư mục d1 tạo tệp t1. Viết các câu lệnh tạo các liên kết vật lý và biểu tượng p-link và s-link trong thư mục d2 trỏ đến t1 sử dụng đường dẫn tương đối. Dịch chuyển t1 đến d2. Xác định trạng thái của các liên kết*

Bài tập 4.5 *Trong thư mục nhà, tạo thư mục /temp/d1 và /temp/d2. Trong thư mục d1 tạo tệp t1. Viết các câu lệnh tạo các liên kết vật lý và biểu tượng p-link và s-link trong thư mục d2 trỏ đến t1 sử dụng đường dẫn tuyệt đối. Dịch chuyển t1 đến d2. Xác định trạng thái của các liên kết*

Bài tập 4.6 *Viết câu lệnh find tìm các tệp có đuôi .wri và có kích thước nhỏ hơn 10M.*

Bài tập 4.7 *Viết câu lệnh locate tìm vị trí của tệp cp.*

Chương 5

Tài khoản và quyền

5.1 Khái niệm tài khoản và nhóm tài khoản

5.1.1 Tài khoản

Để có thể sử dụng hệ thống, NSD cần có một tài khoản. Tài khoản được đặc trưng bởi một cặp tên đăng nhập (user name, login) và mật khẩu (password). Mỗi tài khoản có một mã số người sử dụng (user id) mà hệ thống sử dụng để xác định tài khoản thực hiện các thao tác sau khi đăng nhập. Trong hệ điều hành Linux có 3 loại tài khoản: tài khoản của người quản trị hệ thống (root) có toàn quyền thực hiện các thao tác trên hệ thống, tài khoản thông thường chỉ có toàn quyền trên một thư mục dành riêng gọi là thư mục nhà, trên các thư mục khác chỉ có quyền hạn chế. Loại tài khoản thứ 2 này được chia làm 2 loại, một loại chỉ dùng để chạy các ứng dụng hệ thống (daemon), và một loại dành cho NSD thông thường. Với tài khoản của người sử dụng thông thường, ngoài tên, mật khẩu và mã số tài khoản, Linux còn cung cấp thêm thư mục nhà, shell mặc định cho NSD. Các thông tin này thường được đặt ở tệp /etc/passwd.

Tệp /etc/passwd là một tệp văn bản, mỗi dòng tương ứng với một tài khoản. Trong dòng có nhiều trường thông tin phân cách nhau bằng dấu :, lần lượt là tên tài khoản, mật khẩu, mã số tài khoản, mã số nhóm mặc định của tài khoản, Mô tả ngắn về tài khoản, thư mục nhà và shell,

```
[trunghq@localhost trunghq]$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/etc/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
ntp:x:38:38::/etc/ntp:/sbin/nologin
trunghq:x:500:500::/home/trunghq:/bin/bash
```

Trường mật khẩu trong mỗi hàng thường có giá trị x. Giá trị này chỉ ra rằng mật khẩu không được lưu ở trong tệp /etc/passwd mà được mã hóa và lưu trong tệp /etc/shadow. Trường thứ 3 và thứ 4 lần lượt lưu trữ mã số của tài khoản và mã số của nhóm tài khoản mặc định. Linux phân biệt NSD chỉ thông qua mã số NSD, do đó nếu tồn tại 2 dòng có cùng mã só NSD, hệ

thống sẽ báo lỗi. Nếu có 2 dòng có cùng mã số tài khoản, khi đăng nhập vào hệ thống tài khoản đứng trước sẽ được sử dụng.

Tệp shadow chứa mật khẩu và các thông tin có liên quan đến mật khẩu. Tương tự như /etc/passwd, /etc/shadow gồm nhiều dòng, phân tách nhau bằng dấu :.

```
User:Pwd:Last pwd change :Minimum:Maximum:Warn:Inactive :Expire
```

User:	Tên đăng nhập
Password:	Mật khẩu
Last password change :	Số ngày từ 01/01/1970 đến ngày thay đổi mật khẩu cuối cùng
Minimum:	Số ngày tối thiểu giữa 2 lần thay đổi mật khẩu.
Maximum:	Số ngày tối đa mật khẩu có giá trị.
Warn :	Số ngày đến khi mật khẩu bị hết hạn.
Inactive :	Số ngày đến khi mật khẩu bị vô hiệu hóa vì quá hạn.
Expire :	Số ngày từ 01/01/1970 đến khi mật khẩu bị vô hiệu

5.1.2 Nhóm tài khoản

Để có thể quản trị các tài khoản NSD một cách có hiệu quả hơn, NSD thường được gộp vào thành nhóm người sử dụng. Việc gộp thành các nhóm cho phép giảm bớt số lượng các thao tác quản trị khi thực hiện phân quyền theo nhóm chứ không phân theo NSD. Mặc định, tất cả NSD đều nằm trong một nhóm. Khi mới được tạo ra, NSD nằm trong nhóm có tên trùng với tên đăng nhập, có mã số trùng với mã số của NSD.

Thông tin về các nhóm được lưu trữ trong tệp /etc/group. Tệp /etc/group bao gồm nhiều dòng, mỗi dòng chia thành các trường bằng dấu :. Trường đầu tiên là tên nhóm, trường thứ 2 là mật khẩu nhóm và trường thứ 3 là mã số của nhóm, trường cuối cùng là danh sách NSD thuộc nhóm, phân tách bằng dấu ,. Giá trị x trong trường mật khẩu chỉ ra là nhóm sử dụng mật khẩu lưu trữ trong tệp /etc/gpasswd. Cấu trúc của tệp /etc/gpasswd tương tự như tệp /etc/passwd.

5.2 Các thao tác quản lý tài khoản

Các thao tác liên quan đến quản lý tài khoản bao gồm: thêm, xóa, thay đổi thông tin về tài khoản NSD, thay đổi mật khẩu NSD. Các thao tác quản lý nhóm NSD liên quan đến thêm, xóa, thay đổi thông tin, thêm thành viên của nhóm NSD. Ngoài ra còn một số thao tác bổ sung như thay đổi NSD và nhóm sử dụng hiện tại, hiển thị các tài khoản đang kết nối, hiển thị các nhóm mà người sử dụng là thành viên, thay đổi mật khẩu nhóm.

Chức năng	Câu lệnh
Thêm NSD	useradd
Xóa NSD	userdel
Thay đổi thông tin NSD	usermod
Thay đổi mật khẩu NSD	passwd
Thêm nhóm NSD	groupadd
Xóa nhóm NSD	groupdel
Thay đổi thông tin nhóm NSD	groupmod
Thêm thành viên	gpasswd -a
Thay đổi NSD hiện tại	su
Thay đổi nhóm NSD hiện tại	sg, newgrp
Hiển thị các tài khoản đang đăng nhập	users
Hiển thị các nhóm của NSD	groups
Thay đổi mật khẩu nhóm	gpasswd

Bảng 5.1: Các lệnh quản lý nhóm và NSD

Để thực hiện các thao tác này, có thể không cần sử dụng bất cứ câu lệnh đặc biệt nào, nà chỉ cần thay đổi nội dung của các tệp /etc/passwd, /etc/shadow, /etc/group và /etc/gpasswd. Để cho thuận tiện, các phần mềm đã được phát triển để thực hiện các thay đổi trong các tệp nói trên một cách tự động.

Trừ các thao tác thay đổi thông tin và mật khẩu của chính nhóm hoặc NSD, các thao tác khác đều đòi hỏi có quyền của người quản trị root. Bảng 5.2 chỉ ra một số câu lệnh thực hiện các chức năng nói trên.

Ví dụ

5.2.1 Quyền quản trị root

Để thực hiện các thao tác quản trị hệ thống cần có quyền của quản trị hệ thống (root). Để có quyền này, NSD có 2 cách:

- Đăng nhập bằng tài khoản root hoặc sử dụng lệnh su. Cả 2 trường hợp đều cần có mật khẩu của root.
- Đăng nhập bằng tài khoản thông thường và sử dụng lệnh sudo để thực hiện các thao tác cần quyền quản trị root.

Sử dụng tài khoản root có lợi thế là thuận tiện, khi gõ các câu lệnh không cần phải gõ thêm câu lệnh sudo. Tuy nhiên, chính điểm thuận tiện này đôi khi lại tạo điều kiện cho NSD thực hiện các thao tác quản trị một cách không

chủ ý, dẫn đến các sai sót có hậu quả lớn đến hệ thống. Một điểm hạn chế khác của việc dùng tài khoản root là các quản trị viên phải chia sẻ tài khoản root, như vậy rất khó qui trách nhiệm và giám sát hoạt động của các quản trị viên.

Sử dụng lệnh sudo không cần phải sử dụng mật khẩu root mà chỉ cần sử dụng mật khẩu của chính NSD. Như vậy khi thực hiện lệnh sudo, NSD đã chắc chắn muốn thực hiện một thao tác với quyền của quản trị hệ thống. Từng NSD quản trị viên đăng nhập bằng tài khoản và mật khẩu của mình, sử dụng sudo với mật khẩu của mình, do đó có thể dễ dàng giám sát các thao tác của các quản trị viên. Hiển nhiên mọi quản trị viên khi đó đều có thể thay đổi mật khẩu của các thành viên khác hoặc của root, nhưng khi đó các thao tác này sẽ bị ghi lại. Trong các hệ thống sử dụng sudo, thông thường tài khoản root không được kích hoạt (mật khẩu rất khó). NSD quản trị viên có quyền thực hiện lệnh sudo nếu tên đăng nhập hoặc ID của quản trị viên có trong /etc/sudoers.

5.3 Khái niệm quyền truy cập và chủ sở hữu

Các tài nguyên trong hệ thống Linux được biểu diễn bằng các tệp. Việc phân quyền các tài nguyên này cũng được tiến hành dựa trên các tệp. Phân quyền cần chỉ ra ai được làm cái gì trên đối tượng đang được xem xét. Hệ thống tệp Linux có 2 loại tệp cơ bản: Thư mục và tệp thông thường.

5.3.1 Các quyền

Trên một tệp hoặc một thư mục có thể có 3 loại quyền: r- đọc, w-ghi và x-thực hiện. Còn một loại quyền thứ 4, quyền đặc biệt sẽ được xem xét sau.

Trên một tệp, quyền r có nghĩa là được đọc nội dung tệp. Quyền w là quyền được thay đổi nội dung tệp. Quyền x là quyền có thể được thực hiện tệp như một lệnh nhị phân hoặc một kịch bản shell. Có thể thấy quyền x chỉ được thực hiện khi có quyền đọc. Các ví dụ minh họa về các quyền của tệp được minh họa trong Hình 5.3.2.

Trên một thư mục, quyền r là quyền được đọc nội dung của thư mục (liệt kê các tệp nằm trong thư mục), quyền w là quyền thay đổi danh sách này, còn quyền x tương ứng với việc chọn thư mục là thư mục làm việc hiện tại. Điểm cần chú ý là để thực hiện quyền r và quyền w với thư mục, cần có quyền x Các ví dụ minh họa về các quyền của thư mục được minh họa trong Hình 5.3.2.

5.3.2 Các đối tượng sử dụng

Trên một tệp hoặc thư mục phân biệt 3 lớp NSD khác nhau. Chủ sở hữu (owner) là NSD đã tạo ra tệp và có quyền thay đổi các quyền trên tệp. Nhóm chủ sở hữu là nhóm đã tạo ra tệp. Tất cả các đối tượng còn lại được ký hiệu là khác (other). Chỉ có chủ sở hữu của tệp hoặc root có quyền thay đổi quyền trên các tệp. Với mỗi nhóm NSD có thể định nghĩa cả 3 quyền đã trình bày ở trên. Như vậy với mỗi một tệp sẽ có 9 loại quyền khác nhau.

Khi liệt kê các tệp bằng câu lệnh ls -la, các quyền này được biểu diễn bằng 9 chữ cái ngay sau chữ cái biểu diễn kiểu của tệp. 3 chữ cái đầu tiên biểu diễn các quyền của chủ sở hữu, 3 chữ cái tiếp theo biểu diễn các quyền của nhóm sở hữu, 3 chữ cái cuối cùng biểu diễn các quyền của những NSD khác. Mỗi bộ 3 chữ rwx biểu diễn 3 quyền đọc ghi thực hiện đầy đủ. Quyền thiếu được thay thế bởi dấu -. Ví dụ r-x biểu diễn quyền đọc và thực hiện.

Để có thể truy cập vào các tệp trong một thư mục, cần có quyền truy cập vào thư mục, đồng thời phải có quyền thực hiện trên thư mục chứa tệp. Kể cả khi thư mục không có quyền ghi hay đọc, tệp vẫn có thể được đọc, ghi, nhưng không thể bị xóa đi. Một điểm mâu thuẫn là khi thư mục không có quyền đọc, sẽ không thể liệt kê được danh mục các thư mục con và tệp trong thư mục, nhưng lại có thể đọc được bản ghi inode tương ứng của tệp để từ đó đọc được nội dung của tệp. Quyền đọc của thư mục trên thực tế chỉ là quyền liệt kê nội dung. Quyền ghi của một thư mục tương ứng với việc tạo ra tệp mới và xóa tệp cũ. Ở đây cũng có một điểm mâu thuẫn là không cần có quyền đọc, các thư mục con và tệp của thư mục vẫn có thể được xóa đi và tạo thêm ra.

5.3.3 Các quyền đặc biệt

Mặc dù với câu lệnh ls -la mỗi nhóm NSD chỉ hiển thị 3 quyền nhưng trên thực tế Linux lưu trữ thêm một bít cho mỗi nhóm NSD để biểu diễn các quyền đặc biệt. Các quyền này không có ý nghĩa với nhóm NSD khác. Cụ thể có 3 loại quyền, ứng với thư mục và tệp có các ý nghĩa như sau.

SUID bít Là bít thứ 4 trong nhóm bít dành cho NSD. Chỉ dành cho tệp, không dành cho thư mục. Nếu bít này bằng 1, điều này có nghĩa là NSD thuộc nhóm sở hữu và những NSD khác, nếu có quyền được thực hiện tệp, sẽ được thực hiện tệp với quyền của chủ sở hữu tệp. Ví dụ khi thực hiện lệnh passwd từ tài khoản của 1 NSD, lệnh này sẽ được thực hiện với quyền chủ sở hữu của tệp /usr/sbin/passwd là root, do đó có thể thay đổi nội dung tệp /etc/shadow. Ví dụ về SUID bít được mô tả trong Hình 5.3.3

```
#Truy cập tệp không có quyền
----rw-r--    1 trunghq  trunghq         0 Apr 19 17:22 test
[trunghq@localhost trunghq]$ echo "Thi nghiem so 1" >test
-bash: test: Permission denied
[trunghq@localhost trunghq]$ cat test
cat: test: Permission denied
[trunghq@localhost trunghq]$ ./test
-bash: ./test: Permission denied

#Truy cập tệp chỉ có quyền ghi
--w-rw-r--    1 trunghq  trunghq         0 Apr 19 17:22 test
[trunghq@localhost trunghq]$ echo "Thi nghiem so 1" >test
[trunghq@localhost trunghq]$ cat test
cat: test: Permission denied

#Chỉ có quyền đọc
-r--rw-r--    1 trunghq  trunghq         9 Apr 19 17:24 test
[trunghq@localhost trunghq]$ cat test
Thi nghiem so 1
[trunghq@localhost trunghq]$ echo "Thi nghiem so 1" >test
-bash: test: Permission denied
[trunghq@localhost trunghq]$ ./test
-bash: ./test: Permission denied

#Có quyền thực hiện, không có quyền đọc
---xrw-r--    1 trunghq  trunghq         9 Apr 19 17:24 test
[trunghq@localhost trunghq]$ ./test
./test: ./test: Permission denied
[trunghq@localhost trunghq]$ cat test
cat: test: Permission denied
[trunghq@localhost trunghq]$ echo ls > test
-bash: test: Permission denied

#Có quyền đọc và thực hiện
-r-xrw-r--    1 trunghq  trunghq         9 Apr 19 17:24 test
[trunghq@localhost trunghq]$ ./test
./test: line 1: Thi: command not found
```

Hình 5.3.1: Các quyền của tệp

```
#Không có quyền thực hiện, có quyền ghi và đọc
drw-rwxr-x    2 trunghq  trunghq      4096 Apr 19 18:47 dtest
[trunghq@localhost trunghq]$ cd dtest
-bash: cd: dtest: Permission denied
[trunghq@localhost trunghq]$ ls dtest/
ls: dtest/test: Permission denied
[trunghq@localhost trunghq]$ cat dtest/test
cat: dtest/test: Permission denied
[trunghq@localhost trunghq]$ touch dtest/test2
touch: creating ‘dtest/test2’: Permission denied
[trunghq@localhost trunghq]$ rm dtest/test
rm: cannot lstat ‘dtest/test’: Permission denied

#Có quyền thực hiện
d--x------    2 trunghq  trunghq      4096 Apr 19 18:47 dtest
[trunghq@localhost trunghq]$ cd dtest
[trunghq@localhost dtest]$ rm test
rm: cannot remove ‘test’: Permission denied
[trunghq@localhost dtest]$ touch test2
touch: creating ‘test2’: Permission denied
[trunghq@localhost dtest]$ echo "Thi nghiem so 2" >test
[trunghq@localhost dtest]$ cat test
Thi nghiem so 2
#Có quyền thực hiện, không có quyền đọc
d-wx------    2 trunghq  trunghq      4096 Apr 19 19:11 dtest
[trunghq@localhost trunghq]$ cd dtest
[trunghq@localhost dtest]$ rm test
[trunghq@localhost dtest]$ echo "Thi nghiem so 2" >test
[trunghq@localhost dtest]$ touch test2
[trunghq@localhost dtest]$ ls -l
ls: .: Permission denied
[trunghq@localhost dtest]$ cd ..
[trunghq@localhost trunghq]$ ls -l dtest
ls: dtest: Permission denied
```

Hình 5.3.2: Các quyền trên thư mục

```
[trunghq@localhost trunghq]$ ls -la /bin/cp
-rwxr-xr-x    1 root     root        47732 Feb 19  2003 /bin/cp
[trunghq@localhost trunghq]$ cp proc.c /
cp: cannot create regular file ‘/proc.c’: Permission denied
[trunghq@localhost trunghq]$ ls -la /bin/cp
-rwsr-xr-x    1 root     root        47732 Feb 19  2003 /bin/cp
[trunghq@localhost trunghq]$ cp proc.c /
[trunghq@localhost trunghq]$ ls -la /proc
proc    proc.c
[trunghq@localhost trunghq]$ ls -la /proc.c
-rw-rw-r--    1 root     trunghq        62 Apr 20 00:21 /proc.c
```

Hình 5.3.3: Ví dụ về SUID bít

SGID bít Là bít thứ 4 trong nhóm bít dành cho nhóm sở hữu. Khi được đặt với tệp, bít này có ý nghĩa là tệp sẽ được thực hiện với quyền của nhóm sở hữu tệp, không phải là quyền của người thực hiện. Với thư mục, khi bít này được đặt bằng 1, tất cả các tệp và thư mục con tạo ra trong thư mục này sẽ có nhóm chủ sở hữu là nhóm chủ sở hữu của thư mục. Ví dụ về SGID bít với thư mục được trình bày ở Hình 5.3.4. Ví dụ về SGID bít với tệp xin dành cho bạn đọc.

Sticky bit -t Là bít thứ 4 trong nhóm bít dành cho các NSD khác (other). Chỉ có ý nghĩa đối với thư mục. Khi bít này được đặt bằng 1, chỉ có chủ sở hữu của tệp trong thư mục mới có thể xóa được. Những NSD khác có thể sao chép, đọc, thay đổi nội dung nhưng không xóa được tệp. Ví dụ về sticky bít xem Hình 5.3.5

5.4 Quản lý quyền truy cập và chủ sở hữu

5.4.1 Thay đổi chủ sở hữu

Chủ sở hữu của tệp và thư mục được thay đổi bằng 2 câu lệnh chgrp và chown. Chỉ có người quản trị (root) mới có quyền thực hiện được 2 câu lệnh này. Cả 2 câu lệnh đều có tùy biến -R cho phép có thể thay đổi các tệp và thư mục con trong một thư mục (Hình 5.4.1).

```
[trunghq@localhost trunghq]$ ls -l
total 8
drwsrwsrwx    3 trunghq  g1           4096 Apr 20 00:51 dtest
-rw-rw-r--    1 trunghq  trunghq        62 Apr 20 00:17 proc.c
[trunghq@localhost trunghq]$ cd dtest
[trunghq@localhost dtest]$ mkdir d2
[trunghq@localhost dtest]$ ls -l
total 12
drwxrwxr-x    2 trunghq  trunghq      4096 Apr 20 00:51 d1
drwxrwsr-x    2 trunghq  g1           4096 Apr 20 00:53 d2
-rw-rw-r--    1 trunghq  trunghq        16 Apr 19 19:11 test
-rw-rw-r--    1 trunghq  trunghq         0 Apr 19 19:11 test2
```

Hình 5.3.4: Ví dụ về SGID bít

```
[trunghq@localhost dtest]$ ls -la /tmp/
total 8
drwxrwxrwt    2 root     root         4096 Apr 19 16:40 .
drwxr-xr-x   23 root     root         4096 Apr 20 00:21 ..
[temp@localhost root]$ echo "Ghi vao tep /tmp/xyz" >/tmp/xyz
[temp@localhost root]$ rm /tmp/xyz
rm: cannot remove ‘/tmp/xyz’: Operation not permitted
[temp@localhost root]$ whoami
temp
```

Hình 5.3.5: Sticky bit

```
[root@localhost temp]# mkdir example
[root@localhost temp]# ls -l
total 4
drwxr-xr-x    2 root     root         4096 Apr 20 01:06 example
[root@localhost temp]# chown trunghq example
[root@localhost temp]# ls
example
[root@localhost temp]# ls -l
total 4
drwxr-xr-x    2 trunghq  root         4096 Apr 20 01:06 example
[root@localhost temp]# chgrp  g1  example
[root@localhost temp]# ls -l
total 4
drwxr-xr-x    2 trunghq  g1           4096 Apr 20 01:06 example
[root@localhost temp]# mkdir example1
[root@localhost temp]# ls -l
total 8
drwxr-xr-x    2 trunghq  g1           4096 Apr 20 01:06 example
drwxr-xr-x    2 root     root         4096 Apr 20 01:07 example1
[root@localhost temp]# chown trunghq:g1 example1
[root@localhost temp]# ls -l
total 8
drwxr-xr-x    2 trunghq  g1           4096 Apr 20 01:06 example
drwxr-xr-x    2 trunghq  g1           4096 Apr 20 01:07 example1
```

Hình 5.4.6: Thay đổi sở hữu

5.4.2 Thay đổi quyền

Quyền của thư mục và tệp được thay đổi bởi câu lệnh chmod. Câu lệnh chmod có dạng:

```
chmod quyền <tên thư mục, tệp>
```

Quyền sử dụng trong câu lệnh có thể được biểu diễn bằng 2 cách:

Bằng cú pháp ugoa+-=rwxts Quyền được biểu diễn bằng một xâu có 3 phần. Phần đầu tiên có thể là một hoặc nhiều chữ cái tạo từ u (user), g(group) và o (other). Trường hợp muốn biểu diễn tất cả các đối tượng dùng a(all). Phần cuối cùng là các quyền. Phần thứ 2 là các toán tử. Toán tử = chỉ ra thao tác đặt đúng quyền của các đối tượng bằng quyền khai báo. Toán tử - chỉ ra quyền mới nhận được bằng quyền cũ và bỏ bớt đi các quyền được khai báo. Toán tử + biểu diễn quyền mới nhận được bằng quyền cũ và thêm các quyền được khai báo.

Bằng cú pháp octan Mỗi nhóm quyền cho một đối tượng (3 bít) được biểu diễn bằng một số octan từ 0 đến 7. 3 loại đối tượng tạo thành một số octan có 3 chữ số (ugo). Các bít đặc biệt được tập trung vào thành chữ số đầu tiên (nếu có 4 chữ số). Ví dụ Hình 5.4.7

5.4.3 umask

Để đảm bảo an ninh cho hệ thống, khi các tệp và thư mục được tạo ra lần đầu tiên, phải đặt một số quyền giới hạn cho tệp. Tập hợp các quyền ban đầu này gọi là quyền mặc định. Ví dụ với tệp, quyền mặc định hay được sử dụng là 644, với thư mục là 755. Như vậy để biểu diễn quyền mặc định của hệ thống cần 2 biến, một biến cho thư mục và một biến cho tệp. Để đơn giản hóa, thay vì dùng 2 biến, Linux chỉ dùng một biến chung gọi là umask. Quyền mặc định của tệp là 0666 trừ đi umask, còn của thư mục là 0777 trừ đi umask. Trong trường hợp trên umask sẽ là 022. Chú ý là các quyền đặc biệt bị hạn chế sử dụng do tính chất không an toàn của các quyền này, do đó luôn luôn không có mặt không phụ thuộc vào giá trị của umask. Ví dụ xin xem trong Hình 5.4.8

5.5 Bài tập

Bài tập 5.1 *Viết các câu lệnh tìm kiếm các tệp có bít suid trên cả hệ thống, kiểm tra có phải là tệp passwd, nếu không bỏ bít suid đi.*

```
[trunghq@localhost trunghq]$ chmod ug=r dtest
[trunghq@localhost trunghq]$ ls -l
total 8
dr--r--rwx    4 trunghq  g1           4096 Apr 20 00:53 dtest
-rw-rw-r--    1 trunghq  trunghq        62 Apr 20 00:17 proc.c
[trunghq@localhost trunghq]$ chmod 777 dtest
[trunghq@localhost trunghq]$ ls -l
total 8
drwxrwxrwx    4 trunghq  g1           4096 Apr 20 00:53 dtest
-rw-rw-r--    1 trunghq  trunghq        62 Apr 20 00:17 proc.c
[trunghq@localhost trunghq]$ chmod 4777 dtest
[trunghq@localhost trunghq]$ ls -l
total 8
drwsrwxrwx    4 trunghq  g1           4096 Apr 20 00:53 dtest
-rw-rw-r--    1 trunghq  trunghq        62 Apr 20 00:17 proc.c
[trunghq@localhost trunghq]$ chmod 2777 dtest
[trunghq@localhost trunghq]$ ls -l
total 8
drwxrwsrwx    4 trunghq  g1           4096 Apr 20 00:53 dtest
-rw-rw-r--    1 trunghq  trunghq        62 Apr 20 00:17 proc.c
[trunghq@localhost trunghq]$ chmod 1777 dtest
[trunghq@localhost trunghq]$ ls -l
total 8
drwxrwxrwt    4 trunghq  g1           4096 Apr 20 00:53 dtest
-rw-rw-r--    1 trunghq  trunghq        62 Apr 20 00:17 proc.c
```

Hình 5.4.7: Thay đổi quyền của tệp

```
[trunghq@localhost trunghq]$ umask
0002
[trunghq@localhost trunghq]$ touch tepmoi
[trunghq@localhost trunghq]$ mkdir thumucmoi
[trunghq@localhost trunghq]$ ls -l
total 12
-rw-rw-r--    1 trunghq  trunghq         0 Apr 20 01:26 tepmoi
drwxrwxr-x    2 trunghq  trunghq      4096 Apr 20 01:27 thumucmoi
[trunghq@localhost trunghq]$ umask 0022
[trunghq@localhost trunghq]$ touch tepmoi1
[trunghq@localhost trunghq]$ mkdir thumucmoi1
[trunghq@localhost trunghq]$ ls -l
total 16
drwxrwxrwt    4 trunghq  g1           4096 Apr 20 00:53 dtest
-rw-rw-r--    1 trunghq  trunghq        62 Apr 20 00:17 proc.c
-rw-rw-r--    1 trunghq  trunghq         0 Apr 20 01:26 tepmoi
-rw-r--r--    1 trunghq  trunghq         0 Apr 20 01:28 tepmoi1
drwxrwxr-x    2 trunghq  trunghq      4096 Apr 20 01:27 thumucmoi
drwxr-xr-x    2 trunghq  trunghq      4096 Apr 20 01:28 thumucmoi1
```

Hình 5.4.8: Ví dụ về umask

Bài tập 5.2 *Thực hiện việc copy và đặt chủ sở hữu thành www-data, quyền thư mục 755 và tệp 644 cho Joomla trong thư mục /var/www/*

Chương 6

Quản lý tiến trình và tác vụ

6.1 Tiến trình

6.1.1 Khái niệm

Tiến trình là một chương trình đang được thực hiện. Để xây dựng một chương trình, NSD viết mã nguồn, dịch ra mã nhị phân. Khi tệp chứa mã nhị phân được gọi, mã nhị phân sẽ được tải vào bộ nhớ. Câu lệnh đầu tiên trong khối lệnh sẽ được đăng ký để hệ điều hành sẽ thực hiện.

Trong các hệ điều hành đơn nhiệm, cùng một lúc chỉ có một tiến trình được thực hiện, do đó sau khi được tải vào bộ nhớ, điều khiển sẽ được chuyển về cho câu lệnh đầu tiên của tiến trình.

Trong các hệ điều hành đa nhiệm, có nhiều tiến trình cùng được thực hiện. Các tiến trình này sẽ được đăng ký vào trạng thái thực hiện. Hệ điều hành căn cứ vào đăng ký và mức độ ưu tiên của tiến trình để thực hiện. Mỗi tiến trình luôn luôn được gọi bởi một tiến trình khác. Tập hợp các tiến trình tạo ra một cây các tiến trình.

Một tiến trình khi được thực hiện sẽ có quyền để tác động vào các tài nguyên của hệ thống giống với quyền của người gọi thực hiện tiến trình, trừ trường hợp có bít phân quyền đặc biệt bằng 1.

6.1.2 Thuộc tính

Linux quản lý tiến trình bằng cách gán cho mỗi tiến trình một tham số. Các tiến trình có một chủ sở hữu và nhóm chủ sở hữu. Các tiến trình cũng có mức độ ưu tiên để khi cần thiết có thể được phân phối tài nguyên phù hợp với mức độ ưu tiên. Một tiến trình ở trong Linux có thể có các thuộc tính sau:

- Một định danh (pid)
- Một tiến trình cha (ppid)
- Người sở hữu (uid) và nhóm (gid)
- Câu lệnh kích hoạt tiến trình
- Một đầu vào chuẩn (stdin), một đầu ra chuẩn (stdout), một kênh báo lỗi chuẩn (stderr)
- Thời gian sử dụng CPU (CPU time) và mức độ ưu tiên
- Thư mục hoạt động hiện tại của tiến trình
- Bảng các tham chiếu đến các file được tiến trình sử dụng.

6.1.3 Phân loại

Các tiến trình có thể được chia làm 2 loại.

Tiến trình hệ thống Một số tiến trình được khởi động bởi hệ thống, quản lý các tài nguyên của hệ thống, tương tác chủ yếu với các ứng dụng khác, không có giao diện với NSD. Các tiến trình này gọi là các tiến trình hệ thống. Các tiến trình hệ thống thường được thực hiện trên quyền của các tài khoản hệ thống hoặc tài khoản root.

Tiến trình NSD Các tiến trình của NSD được NSD khởi động bằng câu lệnh. Thông thường các tiến trình này tương tác với NSD thông qua giao diện dòng lệnh hoặc giao diện đồ họa. Các tiến trình này được thực hiện với quyền của NSD đã khởi động tiến trình, trừ các trường hợp các lệnh có bít đặc biệt bằng 1.

6.1.4 Các trạng thái của tiến trình

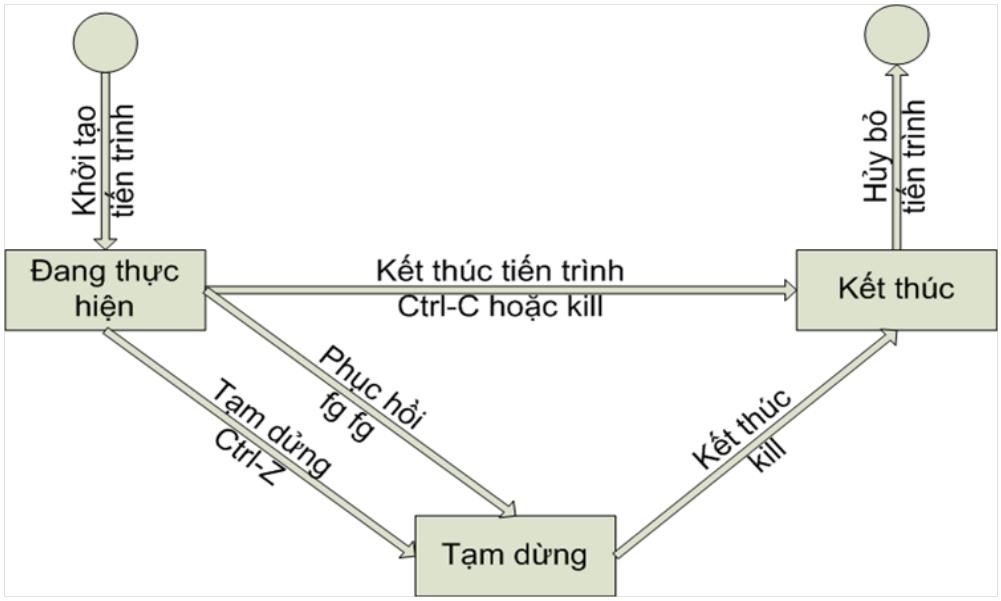
Khi được kích hoạt, nội dung của tệp nhị phân chứa lệnh được tải vào trong bộ nhớ. Câu lệnh đầu tiên của chương trình được xếp vào hàng đợi để hệ thống thực hiện. Tiến trình đã chuyển từ trang thái không được kích hoạt sang trạng thái đang được thực hiện. Trường hợp được khởi động từ giao diện dòng lệnh bởi NSD, nếu NSD ấn phím Ctrl-C, tiến trình sẽ kết thúc, nội dung tệp thực hiện được loại ra khỏi bộ nhớ. Tiến trình ở trạng thái kết thúc. Có thể thấy trạng thái kết thúc và trạng thái không kích hoạt là giống nhau, nên thường được gọi chung là trạng thái kết thúc.

Trong quá trình thực hiện, nếu tiến trình vẫn chiếm console và NSD chọn tổ hợp Ctrl-Z, tiến trình chuyển sang trạng thái tạm dừng. Trong trạng thái tạm dừng, tiến trình vẫn còn ở trong bộ nhwos, tuy nhiên không được đăng ký để hệ thống thực hiện. Tất cả các giá trị trong bộ nhớ của tiến trình vẫn được bảo toán, tuy nhiên quá trình tính toán của tiến trình không được thực hiện. Hình**??** mô tả các trạng thái của tiến trình.

6.1.5 Các thao tác với tiến trình

Để quản lý các tiến trình, NSD cần thực hiện các thao tác: Liệt kê các tiến trình đang được thực hiện trong hệ thống, thay đổi mức độ ưu tiên của các tiến trình, thay đổi trạng thái của các tiến trình.

Câu lệnh ps Câu lệnh ps hiển thị các tiến trình đang được thực hiện. Mặc định câu lệnh này chỉ hiển thị các tiến trình do NSD khởi động. Cần thêm



Hình 6.1.1: Trạng thái của tiến trình

tùy biến -aux để hiển thị các tiến trình khác với đầy đủ thông tin hơn về tiến trình.

```
$ ps
 PID TTY          TIME CMD
2803 pts/1    00:00:00 bash
2965 pts/1    00:00:00 ps
$ ps aux
USER   PID  %CPU  %MEM  VSZ   RSS  TTY  STAT START TIME COMMAND
root     1   0.1   0.1  1104  460   ?    S   15:26 0:03 init[3]
...
trunghq  951   0.0   0.3  1728  996  pts/0 S   16:09 0:00 bash
trunghq  953   0.0   1.9  6860 4916  pts/0 S   16:09 0:00 emacs
trunghq  966   0.0   0.3  2704 1000  pts/0 R   16:23 0:00 ps aux
...
```

Câu lệnh pstree Tiến trình luôn luôn được kích hoạt bởi một tiến trình khác (tiến trình cha). Với mối quan hệ cha con, tập hợp các tiến trình trên hệ thống tạo ra một cây tiến trình. Cây tiến trình này được hiển thị bởi lệnh pstree và có nút gốc là tiến trình init.

```
init-+-apache2---10*[apache2]
     |-atd
     |-console-kit-dae---63*[{console-kit-dae}]
     |-cron
     |-dbus-daemon
     |-dd
     |-6*[getty]
     |-mysqld_safe-+-logger
     |             `-mysqld---14*[{mysqld}]
     |-rsyslogd---3*[{rsyslogd}]
     |-sshd-+-sshd---sshd---bash---pstree
     |      `-2*[sshd---sshd]
     |-udevd---2*[udevd]
     `-upstart-udev-br
```

Hình 6.1.2: Cây tiến trình hiển thị bằng câu lệnh pstree

.

Câu lệnh kill dùng để gửi các tín hiệu đến cho một tiến trình. Các tín hiệu quan trọng thường sử dụng là:

- SIGTERM (15): tín hiệu yêu cầu tiến trình nhận hoàn tất các công việc đang thực hiện và chuyển sang trạng thái kết thúc.
- SIGKILL (9): tín hiệu yêu cầu tiến trình chuyển sang trạng thái kết thúc ngay lập tức.

Lệnh killall dùng để kết thúc tất cả các tiến trình của một câu lệnh thông qua việc truyền tên của câu lệnh dưới dạng một tham số.

Quyền hủy tiến trình thuộc về người sở hữu tiến trình và root.

Câu lệnh top được sử dụng để hiển thị và điều khiển các tiến trình thông qua một giao diện tương tác. Các phím tắt thường dùng là:

- k: gửi tín hiệu SIGTERM đến cho tiến trình.
- r: thay đổi mức độ ưu tiên của tiến trình.

Câu lệnh nice

```
$ nice [-n Value] [Command [Arguments ...]]
```

cho phép thay đổi mức độ ưu tiên của tiến trình. Trong hệ điều hành Linux, các tiến trình được thực hiện song song, phân chia thời gian của CPU dựa vào mức độ ưu tiên. Việc phân chia cụ thể hoàn toàn do nhân hệ điều hành thực hiện. NSD có thể cung cấp mức ưu tiên bằng giá trị nice từ -19 đến 10. Giá trị nice có thể được thay đổi ngay khi khởi tạo tiến trình bằng câu lệnh nice:

```
nice +13 pico myfile.txt
```

thực hiện lệnh pico với mức độ ưu tiên +13, hoặc được thay đổi khi tiến trình đang thực hiện với câu lệnh renice:

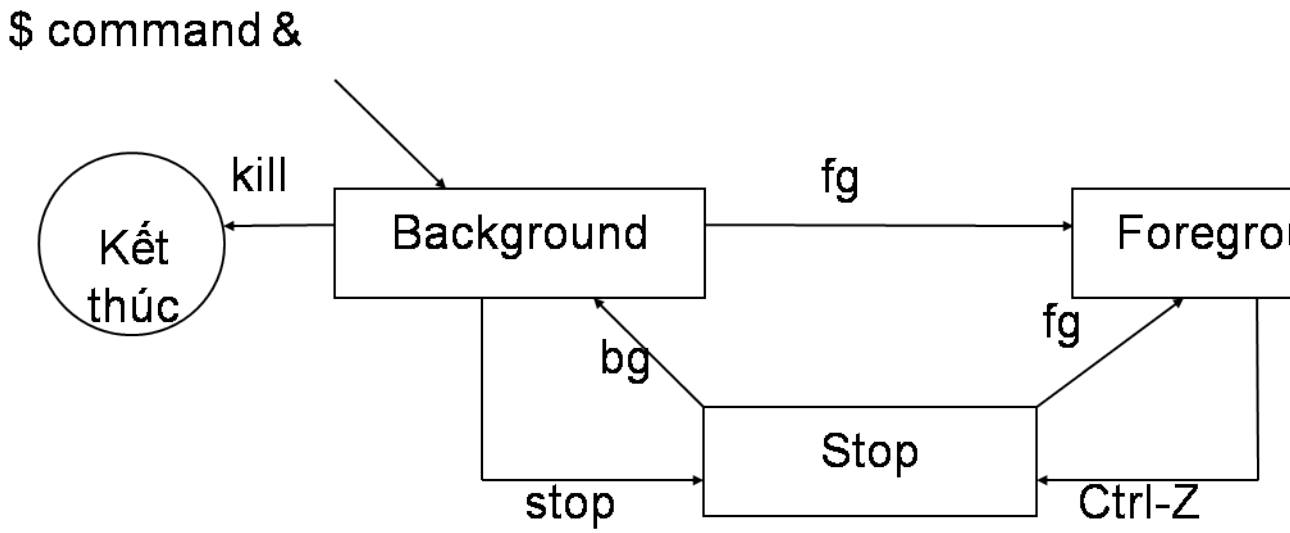
6.2 Quản lý tác vụ

6.2.1 Khái niệm tác vụ

Các tiến trình có thể được gọi bằng một tiến trình khác. Trường hợp được NSD kích hoạt trực tiếp, tiến trình này chính là console (ttyX). Trong trường hợp này, tiến trình được gọi là một công việc-tác vụ của NSD. Mỗi tác vụ đều có một đầu vào chuẩn và đầu ra chuẩn để nhận đầu vào và hiển thị đầu ra. Mặc định đầu vào và đầu ra chuẩn này là consoleKhi tác vụ được khởi tạo từ console, luôn luôn chỉ có 1 tiến trình được sở hữu console. Tiến trình này có thể là chính console hoặc một trong các tác vụ được khởi tạo từ console. Tác vụ sở hữu console được gọi là thực hiện ở chế độ foreground (bề mặt), các tác vụ khác thực hiện ở chế độ nền (background).

Quá trình chạy ở chế độ hiện sẽ tiến hành theo những bước như sau:

- Thực hiện quá trình fork nhân bản tiến trình cha (trong trường hợp thực thi các lệnh, đó sẽ là tiến trình shell)
- Thực hiện quá trình wait, đưa tiến trình cha vào trạng thái ngủ (sleep).
- Thực hiện quá trình exec, thực thi tiến trình con.
- Sau khi tiến trình con thực thi xong, một tín hiệu đánh thức sẽ được gửi đến tiến trình cha.
- Do quá trình chạy như trên => trong quá trình thực hiện tiến trình con, người sử dụng không thể tương tác với tiến trình cha.
- Quá trình chạy ở chế độ ngầm cho phép thực thi tiến trình cha và tiến trình con một cách độc lập.



Hình 6.2.3: Các trạng thái và các thao tác trên tác vụ

6.2.2 Các thao tác trên tác vụ

Ví dụ:

```
$ emacs &
[1] 756
$ stop 756
# or $ stop %1
$ bg 756
# or $ bg %1
$ kill 756
# or $ kill %1
```

6.3 Cơ chế đường ống

6.3.1 Đầu vào và đầu ra chuẩn của tiến trình

Các tiến trình có một đầu vào chuẩn (stdin) và 2 đầu ra chuẩn (stdout). Mặc định, đầu vào chuẩn là bàn phím dùng để nhập thông tin vào tiến trình, đầu ra chuẩn thứ nhất để hiển thị thông tin, đầu ra chuẩn thứ 2 để báo lỗi. Các đầu vào và đầu ra được ký hiệu bằng 0,1,2.

Thay vì sử dụng đầu vào và đầu ra mặc định, có thể thực hiện chuyển hướng đầu vào và đầu ra.

- Chuyển hướng đầu vào chuẩn (<)

```
$ tee < test.txt
```

- Chuyển hướng đầu ra chuẩn (>, »)

```
$ ls > /dev/lp
$ ls >> test.txt
```

- Chuyển hướng kênh báo lỗi

```
$ rm prog.c 2> /dev/null
$ gcc prog.c 2>> erreur.txt
```

6.3.2 Thực hiện song song các câu lệnh

Có thể gọi nhiều câu lệnh từ một dòng lệnh. Các câu lệnh này có thể được thực hiện theo 3 cách khác nhau.

Thực hiện không phụ thuộc lẫn nhau Câu lệnh cm1 được thực hiện, sau đó câu lệnh cmd2 được thực hiện.

```
cmd 1;cmd2
```

Thực hiện phụ thuộc lẫn nhau, tất cả cần kết thúc thành công Câu lệnh cm1 được thực hiện, sau đó câu lệnh cmd2 được thực hiện nếu câu lệnh cm1 được thực hiện thành công.

```
cmd 1 && cmd2
```

Thực hiện theo cơ chế đường ống Câu lệnh cm1 được thực hiện, sau đó câu lệnh cmd2 được thực hiện. Câu lệnh cmd2 nhận đầu ra của cmd1 làm đầu vào chuẩn

```
cmd1 | cmd2
cat /etc/passwd | grep trunghq
```

Câu lệnh tee Câu lệnh tee cho phép copy đầu ra chuẩn thành 2 đầu ra khác nhau:

```
ls -l | tee test | more
```

Chương 7

Cấu hình hệ thống

Trong chương này bạn đọc sẽ làm quen với các bước cấu hình cần thiết để một máy tính có thể kết nối với mạng nội bộ hoặc Internet

7.1 Các thông tin cần cấu hình

Để một máy tính kết nối với mạng, nó cần phải có các thông tin sau:

- Cấu hình card mạng: card mạng cần được "bật" lên nếu đang ở trạng thái "tắt"
- Địa chỉ IP/Netmask: địa chỉ IP để định danh máy tính trên mạng, giúp các gói tin đến được đúng đích cần đến.
- Gateway: địa chỉ để máy tính kết nối ra Internet
- Cấu hình tên miền: gồm tên miền của máy (nếu máy dùng để chạy dịch vụ) và địa chỉ của máy chủ tên miền để máy tính phân giải được tên miền thành địa chỉ IP khi kết nối ra ngoài.

7.2 Kiểm tra cấu hình

Trước khi tiến hành cấu hình, bạn đọc có thể kiểm tra cấu hình hiện tại hoặc kiểm tra kết nối mạng bằng các lệnh cơ bản sau:

7.2.1 Kiểm tra cấu hình mạng hiện tại của máy

Sử dụng các lệnh sau để kiểm tra cấu hình mạng hiện tại của máy:

ifconfig Kiểm tra cấu hình hiện tại của các card mạng gắn trên máy, thông tin trả về bao gồm cả thông tin về cấu hình của mạng loopback. Nếu người dùng muốn xem thông tin cụ thể của card mạng nào thì có thể gõ <ifconfig TenCardMang>. Hình ví dụ minh họa

cat /etc/resolve.conf Xem thông tin về cấu hình địa chỉ máy chủ DNS. Hình ví dụ mình họa

route Xem thông tin về bảng định tuyến của máy tính, thông tin được liệt kê ra như hình dưới đây. <hình minh họa + giải thích>

hostname Xem thông tin về tên máy tính. Hình ví dụ minh họa Bằng các lệnh trên người sử dụng sẽ có đầy đủ thông tin về cấu hình hiện tại trước khi có quyết định xem cần cấu hình lại thông tin trên máy mình hay tiến hành bước tiếp theo là kiểm tra kết nối mạng hiện tại (xem mạng hoạt động hay bị lỗi)

7.2.2 Kiểm tra kết nối mạng

Để kiểm tra kết nối mạng người sử dụng thường trải qua các bước như sau:

- Kiểm tra dây cáp mạng xem đã cắm vào chưa? Đèn card mạng có sáng không (nháy xanh)
- Dùng lệnh <ping loopback> hoặc <ping 127.x.x.x> để kiểm tra xem cấu hình TCP/IP của mạng đã đúng chưa
- Dùng lệnh <ping DiaChiIPHienTai> để kiểm tra xem card mạng có hoạt động không
- Dùng lệnh <ping IPGateway> để kiểm tra xem đường kết nối đến cổng ra ngoài mạng có thông hay không
- Dùng lệnh <ping IPDNSServer> để kiểm tra xem có thể kết nối đến DNS server không
- Dùng lệnh <ping DiaChiWebBenNgoai> để kiểm tra kết nối ra bên ngoài có thông hay không
- Dùng lệnh <traceroute DiaChiWebBenNgoai> để kiểm tra tuyến đường gói tin đi đến đích

Ví dụ: chèn hình ví dụ vào Sau khi kiểm tra, nếu thấy cần thay đổi lại cấu hình hệ thống của máy tính bạn đọc có thể lựa chọn một trong hai phương pháp: cấu hình bằng lệnh hoặc sửa file cấu hình hệ thống. Hai mục sau đây sẽ hướng dẫn bạn đọc chi tiết về hai phương pháp này.

7.3 Cấu hình bằng câu lệnh

Để cấu hình card mạng bằng câu lệnh bạn đọc thực hiện các bước sau:

Bước 1: Cấu hình địa chỉ IP và netmask bằng lệnh sau: <ifconfig TenCardMang DiaChiIPMoi netmask NetmaskMoi> ví dụ

Bước 2: Cấu hình địa chỉ Default Gateway bằng lệnh sau: <route add default gw DiaChiIPGateway> ví dụ

Bước 3: Bật tắt card mạng để thay đổi có hiệu lực bằng lệnh:

- Tắt card mạng: <ifconfig TenCardMang down>. Ví dụ:
- Bật card mạng: <ifconfig TenCardMang up>. Ví dụ:

7.4 Sửa file cấu hình hệ thống

Như đã đề cập ở các chương trên, làm việc trên linux là làm việc với các tập tin và thư mục, mọi thông tin hoặc cấu hình hệ thống đều nằm trong các file. Các lệnh cấu hình cũng chỉ là một cách sửa lại nội dung file hệ thống một cách nhanh chóng. Người sử dụng hoàn toàn có thể mở các file này ra bằng các trình soạn thảo văn bản như: vi, vim, emacs... để sửa nội dung một cách "trực quan" hơn. Mục này tác giả sẽ giới thiệu các file cấu hình hệ thống cơ bản, nội dung bên trong nó và cách thức sửa đổi thông tin trong các file này. Trước khi vào từng mục cụ thể bạn đọc nên nhớ kỹ một điều, trong hệ thống file của linux các thông tin nằm bên phải của dấu "thăng" là thông tin ghi chú, các trình biên dịch lệnh bỏ qua các nội dung này. Lợi dụng đặc điểm đó bạn đọc có thể tạm thời vô hiệu hóa các lệnh hoặc thông tin không mong muốn bằng cách đặt trước nó một dấu "thăng" mà không cần thiết phải xóa bỏ, điều này rất có ích nếu cần khôi phục các lệnh này một cách nhanh chóng ở lần cấu hình sau. Ngoài ra, bạn đọc có thể dùng phương án này để ghi chú lại các thay đổi để có thể nhớ lại được ở lần truy cập sau.

7.4.1 File /etc/network/interfaces

Tập tin này chứa cấu hình của card mạng như ví dụ dưới đây Như ta bạn đọc thấy, tất các các cấu hình cần thực hiện ở phần trên như địa chỉ IP, netmask, gateway đều lưu trong file này. Bạn đọc sửa lại thông tin trong file này rồi tắt/bật lại card mạng để kích hoạt cấu hình mới.

7.4.2 File /etc/init.d/network

Ngoài cách tắt bật card mạng như ở phần trên bạn đọc có thể thực hiện việc bật, tắt, khởi động lại dịch vụ mạng bằng cách truyền tham số vào tập tin </etc/init.d/network> như sau:

- </etc/init.d/network stop>: Tắt dịch vụ mạng

- </etc/init.d/network start>: Bật dịch vụ mạng
- </etc/init.d/network restart>: Khởi động lại dịch vụ mạng - Bạn đọc có thể dùng lệnh này để cấu hình thay đổi có hiệu lực thay vì tắt rồi bật lại card mạng hay dịch vụ mạng

7.4.3 /etc/resolve.conf

Tệp tin này lưu thông tin về địa chỉ của máy chủ DNS, bạn đọc có thể thêm nhiều địa chỉ DNS khác nhau để máy tính dùng khi cần thiết. Thứ tự ưu tiên khi truy vấn là theo thứ tự từ trên xuống dưới. Khi máy tính truy vấn DNS đầu tiên không có được thông tin cần thiết nó sẽ truy vấn đến DNS thứ hai, ba, bốn...đến khi tìm được thông tin hoặc hết địa chỉ lưu trong file này mới dừng lại. Hình ví dụ

7.4.4 File /etc/hosts

Tệp tin này lưu thông tin ánh xạ địa chỉ IP của máy sang tên miền của máy

Chương 8

Quản lý phần mềm

8.1 Nguyên tắc quản lý phần mềm

8.1.1 Các thành phần của phần mềm trên hệ thống

Khi một phần mềm được cài đặt, trên hệ thống sẽ tồn tại các thành phần sau:

Mã thực hiện Thành phần này chứa các mã có thể thực hiện được của phần mềm. Nếu là phần mềm thông thường chạy trực tiếp trên hệ điều hành, thành phần này chứa các mã nhị phân của phần mềm. Nếu là các phần mềm dạng khác, đây có thể là các mã phù hợp với nền tảng (vd Java bytecode, shell scripts).

Các thư viện phần mềm Trong quá trình xây dựng phần mềm, có một số thành phần, chức năng sử dụng các thành phần của các phần mềm khác (hàm, thủ tục, thư viện, ảnh, ...) Khi được cài đặt trên hệ thống, các thành phần này trở thành độc lập với phần mềm, có thể được sử dụng bởi các phần mềm khác

Các tệp cấu hình Để có thể chạy trên một hệ thống và đáp ứng yêu cầu cụ thể của NSD, phần mềm cần được tùy biến. Giá trị của các tùy biến này cần được lưu lại, độc lập với thư viện phần mềm và mã thực hiện. Trong một số trường hợp, các tùy biến này cho phép có thể tìm kiếm các thư viện phần mềm cần thiết.

Dữ liệu tạm thời Trong quá trình hoạt động của chương trình, các dữ liệu tạm thời được tạo ra. Các dữ liệu này chỉ có ý nghĩa trong một phiên làm việc.

Các thành phần nói trên liên kết với nhau, đảm bảo cho phần mềm hoạt động và cung cấp các tính năng cho NSD. Khi hệ thống và NSD thay đổi nhu cầu sử dụng phần mềm, cần thực hiện một số thao tác để tác động tới các thành phần trên của phần mềm.

8.1.2 Các thao tác có thể thực hiện với các phần mềm trên hệ thống

Cài đặt phần mềm Khi phần mềm chưa tồn tại trên hệ thống, để có thể sử dụng phần mềm cần thực hiện thao tác cài đặt. Việc cài đặt phần mềm cần dựa trên một tập hợp các tệp cần thiết cho phần mềm, cụ thể là các thành phần mã thực hiện, thư viện phần mềm, các tệp cấu hình mặc định. Mã thực hiện thường được đóng gói thành các tệp lưu trữ, bên trong có chứa

các thông tin, thậm chí các kịch bản để có thể copy các tệp thực hiện vào các vị trí cần thiết. Các thư viện phần mềm cũng được đóng gói trong các tệp lưu trữ, sử dụng để đảm bảo hoạt động của phần mềm khi chưa có các thư viện cần thiết trên hệ thống. Thành phần này không bắt buộc phải có khi cài đặt. Sau khi các thành phần trên đã được copy vào các vị trí cần thiết, cần có một cấu hình ban đầu để phần mềm có thể hoạt động. Cấu hình ban đầu này được đóng gói kèm theo các thành phần trên.

Gỡ bỏ phần mềm Khi không có nhu cầu sử dụng phần mềm, NSD có thể gỡ bỏ phần mềm ra khỏi hệ thống. Không giống như gỡ bỏ mã thực hiện, việc gỡ bỏ các thư viện phần mềm có thể gây ảnh hưởng tới các phần mềm khác. Tuy nhiên nếu không gỡ bỏ thì các thư viện phần mềm này sẽ tồn tại trên hệ thống mãi mãi. Như vậy nếu không có công cụ để lấy thông tin về các phần mềm khác, bài toán gỡ một phần mềm có sử dụng các thư viện phần mềm chung với các phần mềm khác là không thực hiện được. Các tệp cấu hình có thể được lưu trữ lại cho trường hợp tái sử dụng phần mềm.

Cấu hình lại phần mềm Khi cài đặt phần mềm, một cấu hình mặc định được đưa vào hệ thống. Cấu hình này thường là một cấu hình đã được nhà sản xuất phần mềm kiểm tra và thử nghiệm kỹ lưỡng, nên có tính ổn định cao. Sau một thời gian quản trị thay đổi cấu hình, có thể xảy ra trường hợp phần mềm được cấu hình chưa đúng dẫn đến hoạt động không ổn định. Thao tác cấu hình lại phần mềm cho phép thiết lập lại các cấu hình mặc định để quản trị viên có thể bắt đầu từ đầu.

Tra cứu thông tin về phần mềm Để đảm bảo các phần mềm trên hệ thống hoạt động tốt, không ảnh hưởng đến các phần mềm khác, quản trị viên cần biết các thành phần của phần mềm nằm ở trên những tệp nào. Qua đó có thể thực hiện các thao tác bổ sung để hoàn thiện việc cài đặt, gỡ bỏ và cấu hình lại phần mềm.

8.1.3 Cách thức quản lý phần mềm

Các phần mềm sử dụng các thư viện phần mềm chung, có các chức năng chung do đó phụ thuộc lẫn nhau. Như vậy khi thực hiện các thao tác trên phần mềm chắc chắn sẽ ảnh hưởng đến các phần mềm khác. Do đó cần quản lý sự ràng buộc giữa các phần mềm, đảm bảo các thành phần của phần mềm luôn luôn sẵn sàng. Có một số cách tiếp cận để quản lý phần mềm:

Độc lập Các phần mềm tự quản lý việc thực hiện các thao tác trên phần mềm. Tự kiểm tra các thư viện phần mềm, các tệp cấu hình thiếu/thừa,

nếu cần thì sẽ cài đặt. Nhược điểm của phương pháp này là người tạo ra các phần mềm sẽ phải quan tâm đến nhiều vấn đề không liên quan gì đến chức năng chính của phần mềm.

Sử dụng kịch bản Để cho thuận tiện, các thao tác kiểm tra cài đặt ở trên được tập hợp lại thành các script, sử dụng chung bởi các nhà phát triển phần mềm. Như vậy sẽ giảm bớt khối lượng công việc cần thực hiện để tạo các module kiểm tra và cài đặt nói trên. Tuy nhiên, các kịch bản này không có đủ khả năng để có thông tin về các phần mềm khác cần thiết cho hoặc dựa vào phần mềm đang xem xét.

CSDL về phần mềm Để thực hiện được các thao tác kiểm tra và cài đặt, cần có một CSDL chung, lưu trữ các thông tin về các phần mềm cài đặt trên hệ thống, đặc biệt là ràng buộc giữa các phần mềm. Kịch bản kiểm tra, cài đặt và gỡ bỏ sẽ được chuẩn hóa cho tất cả các phần mềm, nhà phát triển sẽ không cần phải quan tâm đến nữa.

Công cụ quản lý chung Một bước tiếp theo là quản lý tất cả các phần mềm bằng công cụ quản lý chung, có CSDL phần mềm, có đầy đủ các công cụ cho phép kiểm tra, cấu hình, cài đặt và gỡ bỏ các phần mềm.

Trong mục này đã đề cập đến các khái niệm cơ bản nhất về quản lý phần mềm trong hệ thống. Các mục tiếp theo sẽ đề cập đến việc thực hiện, sử dụng các cơ chế quản lý phần mềm trong hệ điều hành Linux.

8.2 Cài đặt phần mềm từ mã nguồn

Ngoài việc cung cấp mã thực hiện để cài đặt trên hệ thống, phần mềm còn có thể được cài đặt sử dụng mã nguồn. Việc cài đặt phần mềm từ mã nguồn có nhiều lợi thế. Khi dịch phần mềm từ mã nguồn, quản trị viên có thể kiểm soát hoàn toàn (vì có mã nguồn) các chức năng của phần mềm, đảm bảo không có chức năng nào không nằm trong dự kiến được cài đặt vào hệ thống. Việc sử dụng các thư viện của các phần mềm khác cũng có thể được tùy biến (sử dụng thư viện động, tĩnh hoặc tích hợp vào mã thực hiện). Mặt khác, quá trình dịch được thực hiện tại chỗ cho phép trình dịch có thể tối ưu mã thực hiện theo các cấu hình cụ thể của hệ thống. Nhược điểm của phương pháp này là luôn đòi hỏi có trình dịch phù hợp trên hệ thống (một số hệ thống không cài đặt trình dịch vì lý do bảo mật). Quá trình dịch mã nguồn hoàn toàn do lập trình viên và người cài đặt thực hiện, sẽ rất khó khăn cho việc quản lý các phần mềm trên hệ thống nói chung. Quá trình dịch, cài đặt, gỡ bỏ phần mềm không có thông tin về các phần mềm khác trên hệ thống,

nên không thực hiện được các thao tác trên nhiều phần mềm. Thông thường phương pháp này được sử dụng trên các hệ thống có phần mềm ít thay đổi, đòi hỏi hiệu năng cao một cách đặc biệt hoặc các hệ thống thử nghiệm dành cho các nhà phát triển phần mềm.

Mã nguồn có thể được tải về một cách trực tiếp, có thể được cài đặt thông qua các gói mã nguồn, có thể được tải về thông qua các phần mềm quản lý phiên bản như CVS, SVN.

Quá trình dịch mã nguồn với các phần mềm tương đối phức tạp đòi hỏi thực hiện các câu lệnh dịch, copy, tham chiếu tới các tệp khác nhau trong cây con thư mục của mã nguồn. Các nhà lập trình C sử dụng một công cụ để có thể ghi lại và thực hiện các thao tác này lặp đi lặp lại trong quá trình phát triển gọi là automake. Automake sử dụng tệp makefile để ghi lại các thao tác cần thực hiện để có thể dịch mã nguồn ra mã thực hiện, sau đó dùng lệnh make để kích hoạt kịch bản này với các tùy biến khác nhau. Hiển nhiên trong quá trình phát triển kịch bản dịch này sẽ được sử dụng nhiều lần. Makefile còn là công cụ để nhà phát triển có thể hướng dẫn quản trị viên cài đặt phần mềm từ mã nguồn. Makefile được sử dụng chủ yếu với mã nguồn C, tuy nhiên có thể được sử dụng với một công cụ lập trình bất kỳ.

Các thao tác cài đặt, gỡ bỏ, cấu hình lại và cấu hình mặc định phần mềm thực tế cũng là các thao tác tương tự như các thao tác bổ trợ thực hiện trong quá trình dịch mã nguồn thành mã thực hiện. Chính vì vậy, các thao tác này được tích hợp luôn vào kịch bản dịch mã nguồn, được kích hoạt riêng biệt hoặc cùng với quá trình dịch mã nguồn tùy theo các tùy biến của câu lệnh make. Các tùy biến này thông thường là install, uninsstal, clean, reconfigure, ...

Mặc dù Makefile cho phép tùy biến quá trình dịch mã nguồn với rất nhiều các thao tác phụ trợ, tuy nhiên việc tạo ra một Makefile phù hợp với các hệ thống khác nhau, với các trình dịch khác nhau của cùng một ngôn ngữ, với sự có mặt hay không có mặt của các thư viện phần mềm cần thiết trên hệ thống. Trường hợp quá trình dịch với Makefile bị lỗi, cần chỉnh sửa lại Makefile để phù hợp với hệ thống hiện tại. Để hạn chế sự can thiệp của các quản trị viên vào Makefile, quá trình này được thực hiện trước khi dịch mã nguồn. Di kèm theo mã nguồn, có một công cụ (configure, xconfig) cho phép:

- Kiểm tra cấu hình hệ thống có phù hợp
- Thay đổi các dòng của Makefile cho phù hợp với hệ thống

Nếu quá trình cấu hình không hoàn thành, điều này chứng tỏ trên hệ thống còn thiếu một số thành phần cơ bản để có thể dịch mã nguồn. Trường hợp này quản trị viên cần trở lại bước trước để bổ sung thêm các thành phần phần mềm còn thiếu.

Như vậy có thể thấy quá trình cài đặt phần mềm được thực hiện qua các bước:

- Chuẩn bị mã nguồn (tải mã nguồn từ Internet, copy từ các thiết bị lưu trữ, ...)
- Chuẩn bị môi trường để dịch mã nguồn (Chuẩn bị các phần cứng và phần mềm cần thiết)
- Chuẩn bị kịch bản dịch mã nguồn (Thực hiện lệnh cấu hình tự động hoặc copy các Makefile phù hợp)
- Dịch mã nguồn (make all)
- Cài đặt mã thu được sau khi dịch mã nguồn vào hệ thống (make install)
- Cấu hình phần mềm theo các cấu hình mặc định (make config)

Quá trình gỡ bỏ phần mềm được thực hiện bằng lệnh make uninstall và make clean.

Hình 8.2.1 trình bày một ví dụ về cài đặt phần mềm từ mã nguồn.

8.3 Cài đặt phần mềm bằng các công cụ quản lý gói

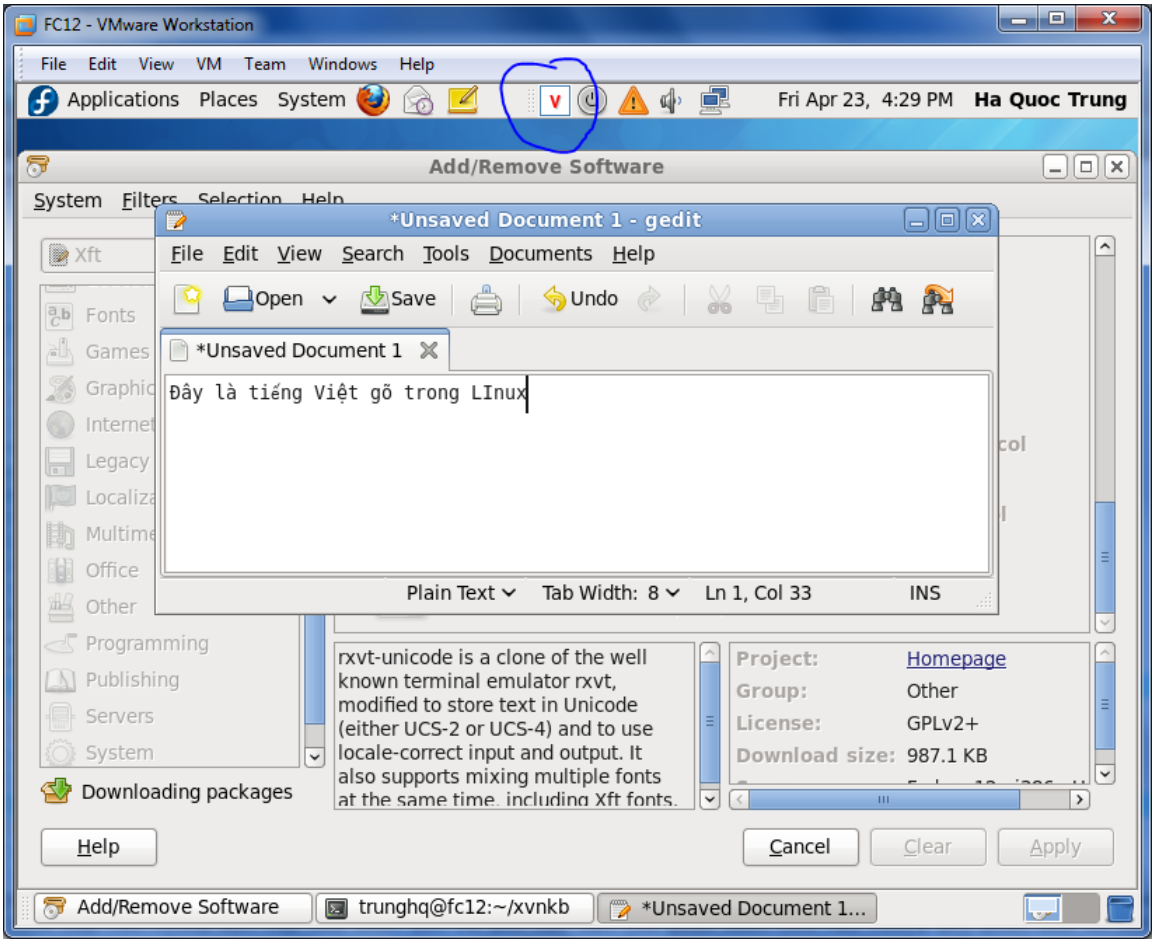
Để đơn giản hóa việc cài đặt các phần mềm, các thành phần cần thiết cho việc cài đặt một phần mềm thường được tích hợp vào trong một tệp duy nhất gọi là gói cài đặt của phần mềm. Như vậy trong gói phần mềm có thể có mã cài đặt, các cấu hình cơ bản và các kịch bản cài đặt, cấu hình lại, gỡ bỏ phần mềm. Để thuận tiện cho các nhà phát triển phần mềm, các chuẩn đóng gói phần mềm được cung cấp kèm theo các công cụ đóng gói phần mềm và chuẩn về cơ sở dữ liệu phần mềm trên hệ thống. Các công cụ đóng gói phần mềm cho phép nhà phát triển có thể dễ dàng tích hợp các thành phần cần thiết vào gói phần mềm, cung cấp các thông tin về ràng buộc phần mềm để thực hiện các thao tác trên nhiều phần mềm. Chuẩn về CSDL cho phép quản lý các ràng buộc về phần mềm này. Trong các bản phân phối của hệ điều hành Linux, có 2 chuẩn được sử dụng rộng rãi. Một chuẩn của Redhat, thường gọi là Redhat Package Manager (RPM) và Debian Package (dpkg). Các ví dụ minh họa trong mục này sẽ được thực hiện dựa trên rpm. Các ví dụ tương tự trên Debian Package không có nhiều khác biệt, xin dành cho bạn đọc thực hiện.

Với các công cụ quản lý phần mềm, các chức năng có thể chia thành 2 nhóm: nhóm quản lý gói phần mềm và gói quản lý phần mềm.

```
wget http://xvnkb.sourceforge.net/xvnkb-0.2.9a.tar.gz
# Hoặc
$ cvs -d ':pserver:anonymous@xvnkb.cvs.sf.net:/cvsroot/xvnkb' login
$ cvs -z3 -d ':pserver:anonymous@xvnkb.cvs.sf.net:/cvsroot/xvnkb' checkout xvnkb
[trunghq@fc12 xvnkb]$ ./configure
Configuration for xvnkb 0.2.10 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... no
Checking xft-config... no
Checking Xft... no
Compile options:
  Enable XFT: no
  Enable spell checking: yes
  Enable extended keystroke: no
  Enable ABC liked Telex keystroke: no
done.

Type "make" to compile
[trunghq@fc12 xvnkb]$ make
[root@fc12 xvnkb]# make install
make[1]: Entering directory `/home/trunghq/xvnkb/tools'
make[1]: Nothing to be done for `all'.
make[1]: Leaving directory `/home/trunghq/xvnkb/tools'
Copy xvnkb  =>  /usr/local/bin ... ok
Copy xvnkb.so.0.2.10  =>  /usr/local/lib ... ok
Copy xvnkb_localeconf.sh  =>  /usr/local/bin ... ok
Initialize xvnkb core ... done
#Khởi động xvnkb
xvnkb
#Sau khi cấu hinh, đã có thể gõ tiếng Việt
```

Hình 8.2.1: Ví dụ về cài đặt và gỡ bỏ phần mềm từ mã nguồn



Hình 8.2.2: Xvnkb được thực hiện trong Linux

8.3.1 Gói phần mềm

Các gói rpm được đặt tên theo nguyên tắc

`name-version-release.architecture.RPM`

, trong đó name là tên ứng dụng hoặc tên gói khi nó được cài đặt trên hệ thống, version là phiên bản của phần mềm chứa trong gói, release là phiên bản phụ của phần mềm. Trường cuối cùng architecture trong tên gói tệp chính là kiểu kiến trúc, định danh phần cứng để thực hiện phần mềm.

Có hai loại gói RPM: RPM nhị phân và RPM nguồn. Một gói RPM nhị phân chứa mã thực hiện được biên dịch riêng cho một kiến trúc cụ thể. Phần lớn những gói RPM nhị phân chứa đựng các ứng dụng hoàn chỉnh, số còn lại cung cấp các thư viện để chia sẻ cho nhiều ứng dụng có thể dùng chung.

Các gói RPM nguồn (thông thường tên tệp có phần mở rộng là .src.RPM. Chứa mã nguồn của chương trình và các scripts cần thiết để dịch ra các mã thực hiện. Đây là một cách khác để phân phối mã nguồn phần mềm.

Quá trình tạo ra một gói rpm được thực hiện bởi lệnh rpmbuild. Đầu vào của lệnh này là tập các tệp cần được đưa vào trong gói, một tệp cấu hình gói.

Sau khi đã được phân phối tới hệ thống của NSD, có thể thực hiện các thao tác sau trên một gói phần mềm

- Tra cứu thông tin gói: liệt kê các thông tin về phần mềm, các tệp chứa trong gói.
- Trích xuất một hoặc nhiều tệp từ gói.
- Kiểm tra gói: thử kiểm tra tính phù hợp của gói với hệ thống hiện tại.
- Cài đặt gói: Cài đặt gói và thực hiện các cấu hình cơ bản.

Tất cả các thao tác trên đều được thực hiện bởi lệnh rpm.

8.3.2 Các thao tác trên phần mềm

Phần mềm sau khi đã được cài đặt vào hệ thống không phụ thuộc vào gói phần mềm. Các thao tác có thể thực hiện trên phần mềm là:

- Tra cứu thông tin về phần mềm.
- Cấu hình lại phần mềm.
- Gỡ bỏ phần mềm.

Các thao tác này cũng được thực hiện bởi rpm. Các thao tác chi tiết với rpm xin dành cho bạn đọc.

8.4 Quản lý các kho phần mềm

8.4.1 Công cụ quản lý phần mềm bậc cao

Trong các ví dụ ở mục trước, trong quá trình cài đặt các gói phần mềm, rpm luôn luôn kiểm tra liệu các gói cần thiết đã được cài đặt chưa. Nếu không có, NSD bắt buộc phải cài đặt các gói phần mềm cần thiết rồi mới thực hiện được việc cài đặt gói đang xem xét. Trong những trường hợp các gói phụ thuộc lẫn nhau tạo ra vòng, giải pháp duy nhất là cài đặt tất cả các gói cùng một lúc. Trong quá trình cài đặt các gói như vậy, tất cả các thao tác tải gói, cài gói đều do NSD thực hiện.

Để cho việc cài đặt trở nên dễ dàng, hầu hết các bản phân phối đều cung cấp các giải pháp quản lý phần mềm bậc cao. Đây là các phần mềm thực hiện các thao tác quản lý phần mềm thông qua rpm hoặc dpkg, tuy nhiên tự động hóa một phần các công việc liên quan đến tải gói và cài đặt gói. Một điểm khác biệt của các phần mềm này so với rpm và dpkg là chúng cho phép quản lý cùng một lúc nhiều phần mềm. Các bản phân phối dựa trên rpm thường dùng dpkg hoặc urpmi. Các bản phân phối dựa trên debian sử dụng apt.

8.4.2 Các kho phần mềm

Để có thể cài đặt các phần mềm một cách tự động, các công cụ nói trên cần có khả năng tải được các gói phần mềm về hệ thống để rồi sau đó cài đặt. Như vậy các phần mềm này cần có một danh sách các kho phần mềm, khi có nhu cầu sẽ tải các phần mềm cần thiết về. Để tải các gói phần mềm, có thể sử dụng giao thức http, ftp, truy cập tệp trực tiếp hoặc thậm chí P2P. Các kho phần mềm thông thường chứa một hoặc nhiều kho phần mềm của các bản phân phối khác nhau. Để cho quá trình tìm kiếm các gói được nhanh chóng, thông tin về các tệp chứa trong kho phần mềm được tập hợp tạo ra một tệp tiêu đề. Các công cụ cài đặt sẽ dựa vào các tiêu đề này để tìm kiếm tệp cần thiết. Hình **??** mô tả các cấu hình kho dữ liệu cơ bản của yum. Các nhà phân phối duy trì trên các server các phiên bản repo phù hợp với các bản phân phối của mình. Với mỗi bản phân phối, có các repo con cho các phần mềm đã được dịch, các phần mềm để thử nghiệm và các phần mềm còn đang được phát triển. Các công cụ bậc cao còn hỗ trợ việc tải các tệp từ các kho phần mềm bằng các giao thức khác khác nhau.

8.4.3 Các công cụ tương tác

Trên cơ sở các công cụ bậc cao, các bản phân phối cung cấp các công cụ với giao diện tương tác để có thể quản lý các phần mềm. Ví dụ yumex, aptitude, synaptic, ... Việc sử dụng các công cụ này thuận tiện hơn, tuy nhiên về bản chất không có gì khác so với việc sử dụng công cụ bậc cao.

8.5 Bài tập

Bài tập 8.1 *Cài đặt phần mềm xvnkb từ mã nguồn. Nêu các lỗi có thể xảy ra và cách giải quyết.*

Bài tập 8.2 *Cài đặt phần mềm mc sử dụng lệnh rpm. Nêu các lỗi xảy ra và cách giải quyết.*

Bài tập 8.3 *Cài đặt phần mềm kile sử dụng câu lệnh yum.*

Bài tập 8.4 *Sử dụng công cụ đồ họa để gỡ x-org khỏi hệ thống*

Bài tập 8.5 *Xây dựng kho dữ liệu phần mềm cho bản FC mới nhất. Cấu hình yum để sử dụng kho phần mềm này*

Bài tập 8.6 *Xây dựng kho dữ liệu phần mềm cho bản Ubuntu mới nhất. Cấu hình apt để sử dụng kho phần mềm này*

Chương 9

Khởi động và kết thúc

9.1 Khởi động phần cứng

9.1.1 Tổng quan về quá trình khởi động

Để có thể có được môi trường cho phép NSD có thể thực hiện các ứng dụng, truy cập vào các tài nguyên của máy tính, cần có một quá trình khởi động các thiết bị vật lý, nạp các phần mềm cần thiết để quản lý các thiết bị, các dịch vụ hệ thống đảm bảo cho các chương trình ứng dụng có thể thực hiện. Quá trình này gọi là quá trình khởi động của máy tính. Tùy theo nhu cầu của NSD, máy tính có thể thực hiện các quá trình khởi động khác nhau. Kết quả của quá trình thực hiện sẽ là các môi trường khác nhau phù hợp với nhu cầu của NSD.

Để có thể đạt được mục đích trên, với các hệ thống máy tính cần thực hiện các bước khởi động sau;

- Khởi động hệ thống vật lý.
- Khởi động hệ điều hành.
- Khởi động các dịch vụ hệ thống.
- Khởi tạo môi trường cho NSD.

9.1.2 Khởi động máy tính

Khi có tín hiệu bật máy, chương trình POST (power on self test) được thực hiện. Chương trình kiểm tra các thiết bị phần cứng cơ bản của hệ thống. Nếu có lỗi máy tính sẽ báo cho NSD, hoặc tắt máy nếu không thể khắc phục. Nếu không hệ thống sẽ thực hiện khởi động theo chế độ mặc định, tìm kiếm một thiết bị lưu trữ để có thể tải nhân HĐH về. Có thể có các khả năng: khởi động từ ổ cứng, ổ di động, ổ mạng, ổ mềm, ... Giai đoạn này khi thực hiện trên các máy tính thông thường thì thời gian tương đối ngắn và không có các thao tác phức tạp. Khi thực hiện trên các máy tính có cấu hình phần cứng phức tạp hơn thì đây chính là giai đoạn để khởi tạo các cấu hình phần cứng của hệ thống. Ví dụ các ổ RAID, các ổ SCSI đều được cấu hình tại thời điểm này.

Để có thể tải và khởi tạo hệ điều hành, các hệ thống thường sử dụng sector đầu tiên trên các thiết bị lưu trữ (Master Boot Record). Trong Master Boot Record chứa một chương trình con sẽ khởi động toàn bộ quá trình tải nhân hệ điều hành. Trường hợp mặc định, chương trình con này sẽ tìm một phân vùng (phân vùng tích cực) của ổ đĩa và khởi động chương trình con nằm trên sector đầu tiên của phân vùng này. Chi tiết hơn về MBR và First Sector có thể xem trong Chương 10

9.2 Khởi động hệ điều hành

Để khởi động hệ điều hành, cần tải nhân hệ điều hành vào hệ thống và sau đó chuyển điều khiển cho nhân hệ điều hành. Giải pháp đầu tiên là cho phép chương trình chứa trong MBR có thể tải trực tiếp hệ điều hành trên thiết bị lưu trữ. Tuy nhiên, với kích thước hạn chế của Boot Program trong MBR (446 byte), rất khó để có thể tải các hệ điều hành bằng các thao tác bậc thấp (khi hệ điều hành chưa được tải và thực hiện, sẽ không có bất cứ công cụ bổ sung nào để đọc thiết bị lưu trữ). Thực tế thường sử dụng phương pháp khởi động 2 bước. Trong bước thứ nhất sẽ tải các phần mềm nhỏ, nằm ở các vị trí dễ đọc, cố định trên thiết bị lưu trữ, có chức năng hỗ trợ truy cập các ổ đĩa. Bước thứ hai, với sự hỗ trợ của các phần mềm nói trên, hệ thống sẽ đọc và tải nhân hệ điều hành.

Với giải pháp thứ 2, chương trình khởi động nằm trong MBR cần phù hợp với các phần mềm sẽ tải trong bước 1. Các chương trình khởi động phổ biến nhất hiện tại là: MS Boot Record, Lilo boor recors, Grub boot record. Trong trường hợp khi MBR và các chương trình khởi động bước 1 không tương thích, có thể cài đặt lại MBR cho phù hợp với hệ thống.

Linux cung cấp 2 công cụ dùng để khởi động hệ điều hành Linux. LILO (Linux Loader) là công cụ cho phép NSD có thể định nghĩa các kịch bản khởi động, sau đó ghi các kịch bản này vào một thư mục đặc biệt (/boot) là thư mục có thể được truy cập dễ dàng từ MBR boot program. Căn cứ vào các dữ liệu này, boot program sẽ khởi động phần còn lại của Lilo và tải nhân Linux vào bộ nhớ. GRUB (Grand Unified Booter) cho phép NSD có thể khai báo trực tiếp các tham số của quá trình khởi động vào thư mục boot, có thể có một giao diện dòng lệnh để thực hiện quá trình khởi động theo nhu cầu của mình). Xu hướng hiện nay GRUB đang thay thế dần LILO.

9.2.1 LILO

Là một chương trình nhỏ dùng để tải nhân ĐH. LILO có thể được cài ở MBR của ổ đĩa, hoặc ở sector đầu tiên của phân vùng, thay thế cho Boot Program để tải nhân hệ điều hành vào trong bộ nhớ. LILO không có chức năng xác thực và bảo mật. LILO sử dụng thư mục /boot để lưu trữ các tệp có liên quan. Các tệp .map lưu trữ vị trí (vật lý) của các tệp mà LILO cần. Trước khi ghi MBR của LILO lên MBR, LILO sao lưu một bản của MBR cũ trong tệp /boot/boot.xxx

Các tệp này được tạo ra sử dụng câu lệnh lilo (/sbin/lilo). Câu lệnh lilo đọc cấu hình mặc định trong /etc/lilo.conf, sau đó ghi vào thư mục boot để phục vụ cho quá trình boot. Lệnh lilo có các option thường sử dụng là:

- lilo -t: kiểm tra liệu /etc/lilo.conf có đúng cú pháp hay không. Thao tác này tránh cho việc ghi các tham số không chuẩn xác cho quá trình khởi động.
- lilo -v hoặc không có tham số: Căn cứ vào nội dung của /etc/lilo.conf thực hiện việc tạo ra các tệp .map trong thư mục boot và ghi LILO MBR vào ổ cứng hoặc phân vùng.

LILO có thể được cấu hình để khởi động HĐH Linux cũng như các hệ điều hành khác. Trường hợp hệ điều hành Linux, có thể truyền một số tham số ban đầu để khởi động nhân hệ điều hành. Hình **??** mô tả một tệp /etc/lilo.còn điển hình. Trong ví dụ này, LILO được cấu hình để hiển thị menu lựa chọn các tùy biến khởi động trong vòng 50s. Quá thời gian đó, LILO sẽ lựa chọn mặc định là cấu hình có tên linux để khởi động. LILO cho phép khai báo nhiều cấu hình khởi động (tối đa 16). Cần khai báo thư mục gốc nằm ở đâu, vị trí của map file và boot sector. Với các cấu hình linux, cần khai báo các tệp chứa hệ điều hành (image và inited), nhãn của cấu hình khởi động và một số tham số của nhân hệ điều hành. Với các hệ điều hành khác, cần khai báo nhãn và phân vùng sẽ khởi động. Để sử dụng lilo, cần thực hiện các thao tác sau:

- Cài đặt lilo.
- Cấu hình /etc/lilo.conf.
- Kiểm tra cấu hình bằng câu lệnh lilo -t.
- Cập nhật cấu hình khởi động bằng câu lệnh lilo -v.

Trường hợp đang sử dụng một chương trình khởi động khác, muốn sử dụng lại lilo, câu lệnh lilo -v sẽ thực hiện thao tác này. Trong quá trình khởi động bằng LILO, nếu có lỗi xảy ra, LILO sẽ thông báo bằng cách hiển thị từng phần của LILO. Cụ thể:

- Không hiển thị gì: Không khởi động được LILO.
- L: Đã khởi động được LILO boot sector, không tìm thấy phần còn lại của LILO.
- LI: Đã tìm thấy phần 2 của LILO nhưng không thực hiện được.
- LIL: Phần 2 của LILO đã được thực hiện, không đọc được map hoặc map không đúng.
- LILO: Hoàn thành quá trình khởi động của LILO.

```
prompt
#Show the lilo menu
timeout=50
#Time to wait for user input before enter default boot
default=linux
#default boot configuration
boot=/dev/sda
#Boot from first scsi disk
map=/boot/map
#boot map, don't touch it

install=/boot/boot.b
#boot sector content
message=/boot/message
linear
image=/boot/vmlinuz-2.4.20-8smp
label=linux
initrd=/boot/initrd-2.4.20-8smp.img
#boot image
read-only
append="root=LABEL=/"
image=/boot/vmlinuz-2.4.20-8
label=linux-up
initrd=/boot/initrd-2.4.20-8.img
read-only
append="root=LABEL=/"
#windows
other = /dev/sda1
label = winxp
```

Hình 9.2.1: Nội dung tệp /etc/lilo.conf

```
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE:  You have a /boot partition.  This means that
#          all kernel and initrd paths are relative to /boot/, eg.
#          root (hd0,0)
#          kernel /vmlinuz-version ro root=/dev/sda2
#          initrd /initrd-version.img
#boot=/dev/sda
default=0
timeout=10
splashimage=(hd0,0)/grub/splash.xpm.gz
title Red Hat Linux (2.4.20-8smp)
root (hd0,0)
kernel /vmlinuz-2.4.20-8smp ro root=LABEL=/
initrd /initrd-2.4.20-8smp.img
title Red Hat Linux-up (2.4.20-8)
root (hd0,0)
kernel /vmlinuz-2.4.20-8 ro root=LABEL=/
initrd /initrd-2.4.20-8.img
title Windows
      rootnoverify (hd0,4)
      chainloader +1
```

Hình 9.2.2: Tệp /boot/grub.conf

9.2.2 GRUB

GRUB (GRand Unified Bootloader) bổ sung thêm nhiều tính năng cho quá trình khởi động. NSD có thể khai báo mật khẩu để khởi động các cấu hình phần mềm, có thể có một shell để tùy biến các quá trình khởi động. Để có thể sử dụng GRUB, cần cài đặt phần mềm GRUB. Câu lệnh grub-instal cài đặt các tệp cần thiết (nếu chưa có) và cài đặt GRUB boot sector. Sau khi thực hiện lần đầu, NSD sẽ không cần thực hiện lại grub install, mà chỉ cần thay đổi trực tiếp vào tệp /boot/grub.conf là đã có tác động trực tiếp đến quá trình khởi động. Hình 9.2.2 mô tả một cấu hình của GRUB có chức năng tương tự như LILO trong Hình 9.2.1. Có thể thấy một vài điểm khác biệt như các phân vùng được đánh dấu bằng cú phapr (hdX,Y), X là số thứ tự của ổ đĩa (từ 0), Y là số thứ tự của phân vùng.

\includegraphics[scale=1]{boot-kernel.png}

Hình 9.3.3: Quá trình khởi động nhân

9.3 Khởi động hệ thống

Sau khi các chương trình khởi động hoàn thành nhiệm vụ, nhân của HĐH được tải vào hệ thống và được thực hiện. Mô hình đơn giản của quá trình thực hiện được mô tả trong Hình **??**. Sau khi khởi động nhân, quyền kiểm soát hệ thống thuộc vệ tiến trình init. Tiến trình này sẽ tiếp tục thực hiện các công việc:

- Tải các mô đun phần mềm bổ sung cho việc quản lý phần cứng của hệ thống.
- Kích hoạt các dịch vụ hệ thống.
- Khởi động các giao diện cho NSD.

Các thao tác này được thực hiện bởi tiến trình init. Tiến trình này là gốc của các tiến trình trong hệ thống. Phụ thuộc vào nhu cầu sử dụng, có thể có các tập hợp phần cứng, phần mềm và giao diện khác nhau của NSD có thể được kích hoạt. Linux tập hợp các phần mềm và phần cứng này thành các mức thực hiện. Linux định nghĩa 7 mức thực hiện từ 0-6. Mức 0 sử dụng khi cần tắt hệ thống. Mức 6 sử dụng khi bật hệ thống. Mức 3 và 5 là các mức thực hiện bình thường của hệ thống, với nhiều NSD và kết nối mạng, trong đó mức 5 có giao diện đồ họa. Mức 2 là mức có nhiều NSD có thể đăng nhập vào hệ thống nhưng không có kết nối mạng. Mức 1 là mức thường dùng khi bảo dưỡng hệ thống, chỉ có 1 NSD kết nối và không có kết nối mạng. Câu lệnh runlevel hiển thị mức thực hiện hiện tại và mức thực hiện trước khi chuyển sang mức thực hiện hiện tại. Câu lệnh init cho phép kích hoạt một mức thực hiện.

Các mức thực hiện được các hệ thống triển khai khác nhau. Mức 3 với một số bản phân phối được định nghĩa có giao diện đồ họa (debian, ubuntu), trong khi với Redhat, FC lại không có.

Init sử dụng tệp cấu hình là /etc/inittab. Một ví dụ về tệp initab điển hình được mô tả trong Hình **??**. Tệp inittab gồm nhiều dòng, mỗi dòng gồm nhiều trường phân cách nhau bằng dấu :. Các trường có ý nghĩa như sau:

- Nhãn: trường này chỉ ra bối cảnh mà dòng lệnh sẽ được thực hiện, hoặc chỉ đơn giản là đánh dấu dòng.

```
# Default runlevel. The runlevels used by RHS are:
#   0 - halt (Do NOT set initdefault to this)
#   1 - Single user mode
#   2 - Multiuser, without NFS (The same as 3, if you do not have networking)
#   3 - Full multiuser mode
#   4 - unused
#   5 - X11
#   6 - reboot (Do NOT set initdefault to this)
#
##Syntax: id:runlevels:action:process
id:5:initdefault:
#Default runlevel. Don't change it to 0 or 6
# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit
l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6

# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
# When our UPS tells us power has failed, assume we have a few minutes
# of power left.  Schedule a shutdown for 2 minutes from now.
# This does, of course, assume you have powerd installed and your
# UPS connected and working correctly.
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"
# If power was restored before the shutdown kicked in, cancel it.
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"
# Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
# Run xdm in runlevel 5
x:5:respawn:/etc/X11/prefdm -nodaemon
```

Hình 9.3.4: Một tệp /etc/inittab điển hình

respawn	Tiến trình được khởi động lại sau khi kết thúc
wait	Tiến trình chỉ khởi động một lần khi kích hoạt mức thực hiện. Init sẽ chờ cho tiến trình kết thúc.
once	Tiến trình sẽ thực hiện một lần
boot	Tiến trình chỉ thực hiện khi khởi động. Mức thực hiện bị bỏ qua.
bootwait	Tiến trình chỉ thực hiện khi khởi động, init chờ tiến trình kết thúc mới thực hiện các thao tác khác
off	Không làm gì cả khác
ondemand	Chỉ thực hiện khi các mức thực hiện a,b,c được kích hoạt. khác
initdefault	Mức thực hiện mặc định của hệ thống
sysinit	Chỉ thực hiện khi hệ thống khởi động, trước khi boot và bootwait được thực hiện
powerwait	Thực hiện khi mất điện nguồn.
powerokwait	Thực hiện khi có điện (lại)
powerfailnow	UPS sắp hết điện
ctrlaltdel	init nhận được SIGNINT= Ctrl-Alt-Del được ấn
kbrequest	Tiến trình khai báo được thực hiện khi nhận được một phím từ bàn phím

Bảng 9.1: Các giá trị có thể của trường cách thức thực hiện

- Mức thực hiện: chỉ ra các mức thực hiện mà dòng sẽ được thực hiện. Mức thực hiện có thể nhận giá trị từ 1 đến 6 hoặc tập hợp của các giá trị đó.
- Cách thức thực hiện: cách thực thực hiện câu lệnh. Bảng 9.1 mô tả các giá trị có thể của trường này.
- Câu lệnh khởi động tiến trình. Trường này bị bỏ qua với một số giá trị của trường cách thức thực hiện.

Tiến trình gốc init đươc khởi động trong 2 trường hợp. Trường hợp thứ nhất khi khởi động hệ thống. Trường hợp thứ 2 khi câu lệnh init được thực hiện. Trong trường hợp thứ nhất, init sẽ đọc inittab và khởi động hệ thống theo mức thực hiện được khai báo trong dòng chứa initdefault. Ở Hình **??**, init tiếp tục thực hiện dòng si::sysinit:/etc/rc.d/rc.sysinit là dòng chỉ thực hiện khi hệ thống khởi động. rc.sysinit là một kịch bản shell, cho phép kiểm tra xem liệu các mô đun quản lý phần cứng đã được tải vào bộ nhớ chưa. Nếu chưa kịch bản này sẽ kích hoạt các phần mềm tương ứng.

```
[trunghq@localhost trunghq]$ ls /etc/rc3.d
K05saslauthd  K50snmpd      S10network   S24pcmcia      S80sendmail
S97rhnsd   K20nfs        K50snmptrapd  S12syslog    S25netfs
S85gpm       S99local       K24irda       K74ntpd       S13portmap
S26apmd        S90crond     K35smb        S05kudzu      S14nfslock
S28autofs      S90xfs       K35winbind    S08iptables   S17keytable
S55sshd        S95anacron   K45named      S09isdn       S20random
S56rawdevices  S95atd
[trunghq@localhost trunghq]$ ls -l /etc/rc3.d/S90
S90crond  S90xfs
[trunghq@localhost trunghq]$ ls -l /etc/rc3.d/S90crond
lrwxrwxrwx    1 root     root           15 Apr  2 19:36
/etc/rc3.d/S90crond -> ../init.d/crond
```

Hình 9.3.5: Nội dung thư mục /etc/rc3.d/

Phân đoạn tiếp theo ở trong tệp inittab là khai báo các tiến trình sẽ được khởi động ở mức thực hiện. Thực chất chỉ có một kịch bản /etc/rc.d/rc với tham số là mức thực hiện tương ứng. Kịch bản /etc/rc.d/rc thực hiện các công việc sau:

- Kiểm tra mức thực hiện và đặt các tham số môi trường có liên quan
- Kết thúc các tiến trình không thực hiện trong mức thực hiện mới
- Khởi động các tiến trình trong mức thực hiện mới.

Các kịch bản khởi động và kết thúc các tiến trình nằm trong thư mục /etc/rcX, trong đó X là mức thực hiện. Trong mỗi thư mục, có các kịch bản để khởi động và kết thúc các tiến trình. Các kịch bản này được ký hiệu là AXXY, trong đó A có thể nhận giá trị S hoặc K thay cho Start hoặc Kill, biểu diễn kịch bản khởi động hoặc kết thúc của tiến trình. XX là số thứ tự, cho biết tiến trình nào sẽ khởi động trước tiến trình nào. A là tên của tiến trình. Các kịch bản này đều là các liên kết tới các kịch bản thực sự ở trong thư mục /etc/init.d. Ví dụ về nội dung thư mục /etc/rc3.d/được mô tả trong Hình 9.3.5

Mỗi mức thực hiện sẽ có một số các tiến trình (dịch vụ) hệ thống được khởi động. Các tiến trình còn lại không được khởi động trong mức thực hiện, cần được kết thúc khi chuyển sang mức thực hiện này. Khi khởi động và khi kết thúc các tiến trình cần tuân theo trật tự nhất định, vì các tiến trình còn phụ thuộc lẫn nhau. Tất cả những yêu cầu này được thực hiện bằng cách:

- Các tiến trình hệ thống được khởi động và kết thúc bằng các kịch bản nằm trong thư mục /etc/init.d
- Các thư mục /etc/rcX tương ứng với mức thực hiện X.
- Trong mỗi thư mục, có các K và S với các số thứ tự quyết định trình tự thực hiện.

Trở lại với tệp /etc/inittab, sau các dòng khai báo cách thức khởi động các tiến trình hệ thống là các khai báo các tác vụ sẽ thực hiện trong một số trường hợp thông thường. Sau đó là phần khởi tạo các giao diện NSD. Có thể thấy mặc định hệ thống sẽ khởi động 6 console (tty1-6) và một giao diện đồ họa.

Như vậy việc phân biệt các mức thực hiện trên thực tế chỉ là phân biệt tập hợp các tiến trình sẽ được khởi động trong mức thực hiện, phụ thuộc vào nội dung tệp /etc/inittab và các thư mục /etc/rc.d/. NSD hoàn toàn có thể thay đổi theo nhu cầu của mình. Các nhà phân phối cũng đóng gói các tệp này khác nhau. Trong các bản phân phối dựa trên Debian, mức 3,4,5 là các mức thực hiện có đồ họa. Trong bản phân phối dựa trên Redhat, mức 3 là mức không có đồ họa.

Các dòng trong /etc/inittab có thể được coi là chuỗi các thao tác sẽ thực hiện khi khởi động một mức thực hiện nào đó, đồng thời cũng có thể coi là các thao tác được thực hiện đáp ứng các sự kiện xảy ra trong hệ thống. Vai trò kép này khiến cho việc quản lý tệp /etc/inittab cũng như các tệp khác gặp nhiều khó khăn.

Trong các bản phân phối mới, init đang được thay thế bởi Upstart, trong đó các dòng của inittab sẽ được tách ra thành các sự kiện, lưu trữ trong thư mục /etc/event.d/. Các liên hệ giữa các sự kiện không được biểu diễn bằng thứ tự, mà biểu diễn bằng quan hệ xảy ra trước/sau với từng sự kiện.

9.4 Các dịch vụ

Các tiến trình hệ thống còn được gọi là các dịch vụ. Với các dịch vụ mạng, có thể chia làm 2 loại: dịch vụ đơn lẻ và dịch vụ khởi động bằng tcpwrapper.

9.4.1 Dịch vụ đơn lẻ

Các dịch vụ đơn lẻ được biểu diễn bằng các kịch bản nằm trong thư mục /etc/init.d/ trong đó thực hiện các thao tác gọi đến các tệp thực hiện của dịch vụ. Các tệp nằm trong thư mục này được liên kết đến bởi các liên kết S và K trong thư mục /etc/rcX.d/. Trên các dịch vụ này, có thể thực hiện các thao tác sau:

- Khởi động dịch vụ.
- Kết thúc dịch vụ.
- Đặt dịch vụ khởi động trong một mức thực hiện nào đó
- Tắt dịch vụ trong một mức thực hiện nào đó.

2 thao tác đầu tiên có thể thực hiện theo 2 cách: gọi trực tiếp kịch bản từ thư mục /etc/init.d/ với các tham số start, stop, reload, restart, ... hoặc thực hiện bằng câu lệnh service <tên dịch vụ> <câu lệnh>. 2 thao tác sau có thể được thực hiện bằng cách tạo các liên kết tương ứng trong các thư mục /etc/rcX.d//. Tuy nhiên việc xác định trật tự ưu tiên gặp nhiều khó khăn, do đó việc cài đặt các dịch vụ đơn lẻ thường được thực hiện với câu lệnh chkconfig. Câu lệnh này cho phép:

- Hiển thị các dịch vụ khởi động ở các mức thực hiện
- Hiển thị các mức thực hiện có khởi động một dịch vụ
- Bật dịch vụ trong mức thực hiện
- Tắt dịch vụ trong mức thực hiện
- Thêm một dịch vụ
- Xóa một dịch vụ

Hình 9.4.1 trình bày các câu lệnh thực hiện các thao tác trên

9.4.2 Superdaemon

Để có thể cung cấp một dịch vụ mạng, trên máy chủ phải có một tiến trình luôn luôn chạy, các tiến trình khác sẽ kết nối vào tiến trình đó. Nếu trên hệ thống có nhiều dịch vụ cùng thực hiện, hệ thống sẽ tiêu tốn tài nguyên để các tiến trình phục vụ cho các dịch vụ đó luôn luôn thực hiện, kể cả khi không xử lý bất cứ yêu cầu nào từ các tiến trình khách. Để giải quyết vấn đề này, Linux sử dụng một tiến trình thường xuyên lắng nghe tất cả các yêu cầu từ phía tiến trình khác, phân tích các yêu cầu. Nếu yêu cầu gửi đến tiến trình đã được kích hoạt, yêu cầu sẽ được chuyển đến cho tiến trình. Nếu không, tiến trình sẽ được kích hoạt rồi yêu cầu mới được chuyển. Sau một khoảng thời gian xác định, tiến trình không có yêu cầu để xử lý sẽ tự kết thúc.

Tiến trình làm nhiệm vụ phân phối nói trên gọi là superdaemon (siêu server) hay còn gọi là tcpwrapper. Phần mềm tương ứng là inetd và xinetd.

```
[root@localhost root]# chkconfig --list crond
crond           0:off   1:off   2:on    3:on    4:on    5:on    6:off
[root@localhost root]# chkconfig --level 235 crond off
[root@localhost root]# chkconfig --list crond
crond           0:off   1:off   2:off   3:off   4:on    5:off   6:off
[root@localhost root]# mv /etc/rc4.d/S
S05kudzu       S13portmap     S25netfs       S80sendmail    S95atd
S08iptables    S14nfslock     S26apmd        S85gpm         S97rhnsd
S09isdn        S17keytable    S28autofs      S90crond       S99local
S10network     S20random      S55sshd        S90xfs
S12syslog      S24pcmcia      S56rawdevices  S95anacron
[root@localhost root]# mv /etc/rc4.d/S90
S90crond  S90xfs
[root@localhost root]# mv /etc/rc4.d/S90crond /etc/rc4.d/S95.d
[root@localhost root]# ls /etc/rc4.d/S9*
/etc/rc4.d/S90xfs      /etc/rc4.d/S95atd  /etc/rc4.d/S97rhnsd
/etc/rc4.d/S95anacron  /etc/rc4.d/S95.d   /etc/rc4.d/S99local
[root@localhost root]# chkconfig --level 4 crond reset
[root@localhost root]# chkconfig --list crond
crond           0:off   1:off   2:off   3:off   4:on    5:off   6:off
```

Hình 9.4.6: Các thao tác trên các dịch vụ đơn lẻ

```
#echo   stream  tcp     nowait  root    internal
#echo   dgram   udp     wait    root    internal

ftp     stream  tcp     nowait  root    /usr/sbin/tcpd  in.ftpd -l -a
#pop-3   stream  tcp     nowait  root    /usr/sbin/tcpd ipop3d
#swat      stream  tcp     nowait.400      root /usr/sbin/swat swat
```

Hình 9.4.7: Ví dụ về cấu hình inetd

Inetd sử dụng tệp /etc/inetd.conf để khai báo các dịch vụ được quản lý bởi inetd. Xinetd sử dụng /etc/xinetd.conf để khai báo các cấu hình chính của superdaemon và /etc/xinetd.d để khai báo các dịch vụ bị quản lý. Hình **??** và Hình **??** mo tả một vài ví dụ về cấu hình các dịch vụ.

Tên và cổng của các dịch vụ cần được thống nhất giữa tiến trình khác và tiến trình chủ. Linux thực hiện bằng cách sử dụng chung một tệp /etc/services phân phối khi cài đặt. Hình 9.4.2 mô tả các dịch vụ truyền thống.

Ngoài mục đích ban đầu, cơ chế superdaemon còn cho phép kiểm soát chung tất cả các yêu cầu đến và đi trên máy tính. Tệp /etc/host.allow liệt kê các máy được quyền kết nối vào superdaemon. Tệp /etc/host.deny liệt kê các máy tính bị cấm kết nối.

9.5 Bài tập

Bài tập 9.1 *Hệ thống với LILO bị hỏng MBR nên không khởi động được. Sử dụng CD rescue, làm thế nào để phục hồi lại MBR của LILO?*

Bài tập 9.2 *Căn cứ vào cấu hình được nêu trong Hình 9.2.2 hãy viết các lệnh để khởi động Linux từ GRUB shell*

Bài tập 9.3 *Cấu hình hệ thống để khởi động không có giao diện đồ họa, với 3 tty.*

Bài tập 9.4 *Chương trình ssh không kết nối được với server sshd. Phân biệt các khả năng lỗi sau đây:*

- *Không có kết nối mạng*
- *Có kết nối mạng, sshd chưa được khởi động*
- *Có kết nối mạng, sshd đã khởi động nhưng máy tính client nằm trong /etc/host.deny.*

```
[root@localhost root]# cat /etc/xinetd.conf
#
# Simple configuration file for xinetd
#
# Some defaults, and include /etc/xinetd.d/
defaults
{
        instances               = 60
        log_type                = SYSLOG authpriv
        log_on_success          = HOST PID
        log_on_failure          = HOST
        cps                     = 25 30
}

includedir /etc/xinetd.d
[root@localhost root]# cat /etc/xinetd.d/
chargen      daytime-udp  rsync        sgi_fam      time-udp
chargen-udp  echo         servers      sshd
daytime      echo-udp     services     time

[root@localhost root]# cat /etc/xinetd.d/sshd
# default: off
# description: The rsync server is a good addition to an ftp server, as it \
#       allows crc checksumming etc.
service ssh
{
        disable = no
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/sbin/sshd
        server_args     = -i
        log_on_failure  += USERID
}
```

Hình 9.4.8: Ví dụ cấu hình xinetd

```
systat          11/udp          users
daytime         13/tcp
daytime         13/udp
qotd            17/tcp          quote
qotd            17/udp          quote
msp             18/tcp                          # message se
nd protocol
msp             18/udp                          # message se
nd protocol
chargen         19/tcp          ttytst source
chargen         19/udp          ttytst source
ftp-data        20/tcp
ftp-data        20/udp
# 21 is registered to ftp, but also used by fsp
ftp             21/tcp
ftp             21/udp          fsp fspd
ssh             22/tcp                          # SSH Remote
 Login Protocol
ssh             22/udp                          # SSH Remote
 Login Protocol
telnet          23/tcp
telnet          23/udp
```

Hình 9.4.9: Một phần của tệp /etc/services

Chương 10

Thiết bị lưu trữ

10.1 Các khái niệm cơ bản

10.1.1 Các loại đĩa vật lý trong Linux

Linux hỗ trợ hầu hết các loại thiết bị lưu trữ có mặt trên thị trường. Các thiết bị lưu trữ luôn luôn được biểu diễn bằng các tệp đặt trong thư mục /dev/. Tùy theo loại kết nối của thiết bị lưu trữ với hệ thống, Linux đặt tên theo qui tắc riêng. Một số loại thiết bị lưu trữ mà hệ điều hành Linux hỗ trợ là:

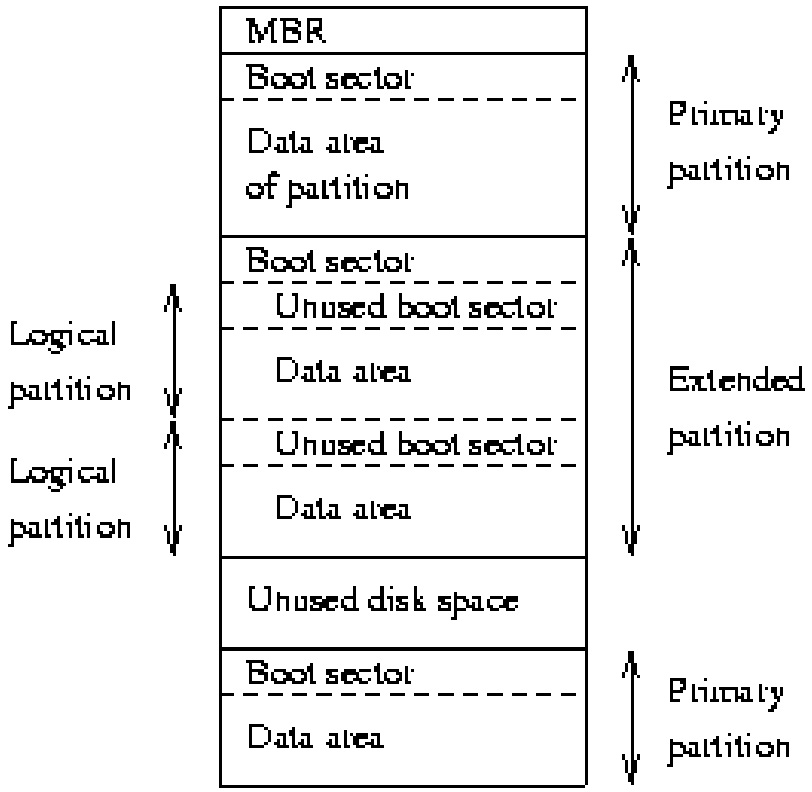
Đĩa cứng giao tiếp theo chuẩn ATA-IDE IDE (Integrated Drive Electronics) là chuẩn giao tiếp xuất hiện từ những năm 80, sử dụng cáp có 40 chân. Mỗi máy tính có thể có 2 bộ điều khiển IDE (primary và secondary). Mỗi bộ điều khiển kiểm soát một cáp IDE. Trên mỗi cáp IDE có thể cắm 2 thiết bị chủ (master) và tớ (slave). Như vậy trên một máy tính có thể kết nối tối đa 4 thiết bị IDE. Linux biểu diễn các thiết bị này bằng các tệp /dev/hda, /dev/hdb, /dev/hdc, /dev/hdd theo thứ tự. Thông thường CD-ROM được kết nối với /dev/hdc. I.

SCSI SCSI (Small Computer System Interface) sử dụng cáp trục để kết nối các thiết bị. Mỗi thiết bị sẽ có một SCSI ID. Trừ thiết bị điều khiển, trên mỗi một cáp SCSI có thể cắm được 7 hoặc 15 thiết bị. Khi hệ thống khởi động, bộ điều khiển scan cáp trục để thiết lập các ID cho các thiết bị, do đó có nhiều trường hợp thứ tự các thiết bị thay đổi trong các lần khởi động khác nhau. Linux biểu diễn các thiết bị kết nối chuẩn SCSI bằng các tệp /dev/sdX, trong đó X là một chữ cái từ a-z biểu diễn thiết bị.

USB và IEEE-1394 Linux cũng như một số HĐH khác không có hỗ trợ trực tiếp cho các thiết bị lưu trữ có các kết nối này. Thông thường, các kết nối này được giả lập như các kết nối SCSI. Như vậy, khi cắm thêm một thiết bị USB vào hệ thống, thiết bị này sẽ đươc coi như một thiết bị SCSI, có khả năng sẽ làm thay đổi thứ tự của các thiết bị SCSI.

RAID Redundant Array of Independent Disks (RAID) là cách thức sử dụng nhiều thiết bị lưu trữ để cho hiệu năng cao và độ tin cậy cao hơn. Linux ký hiệu các thiết bị RAID bằng /dev/mdX, trong đó X là số thứ tự của thiết bị.

LVM Logical Volume Manager (LVM) là cách thức kết hợp các không gian ổ cứng thành một không gian ổ cứng duy nhất. Linux biểu diễn các ổ đĩa Logic này bằng /dev/LogVol00.



Hình 10.1.1: Các phân vùng trên ổ đĩa

10.1.2 Phân vùng

Các ổ đĩa được chia thành các phân vùng (partitions). Thông tin về các phân vùng được nằm trong phần còn lại của MBR. Với cấu trúc như vậy, một ổ đĩa chỉ có thể chia tối đa thành 4 phân vùng. Các phân vùng này gọi là phân vùng chính (primary). Để có thể có nhiều hơn 4 phân vùng trên một ổ đĩa cứng, Boot sector của một trong các phân vùng nói trên sẽ được sử dụng để phân chia phân vùng này thành tối đa 04 phân vùng khác. Phân vùng primary bị chia nhỏ gọi là phân vùng mở rộng (Extended Partition), còn các phân vùng mới được tạo ra gọi là phân vùng logic. Nếu như các phân vùng chính có thể được đọc bởi BIOS, thì các phân vùng logic chỉ có thể được đọc bởi hệ điều hành. Mỗi phân vùng thường được đánh dấu để định dạng theo một loại hệ thống tệp cụ thể, được đánh dấu là tích cực (active) hoặc không tích cực. Một phân vùng tích cực sẽ được chương trình MBR mặc định của ổ đĩa gọi trong quá trình khởi động. Hình **??**.

Linux biểu diễn các phân vùng sử dụng tên tệp giống như với ổ đĩa, bổ sung thêm số thứ tự phân vùng trong ổ đĩa. Các phân vùng chính (primary)

kể cả phân vùng mở rộng được đánh số từ 1-4. Các phân vùng logic được đánh số bắt đầu từ 5.

GRUB sử dụng một cách đánh tên ổ đĩa và phân vùng khác (hdX,Y), trong đó X là số thứ tự của ổ cứng tính từ 0 và Y là số thứ tự của phân vùng tính từ 0.

10.1.3 Hệ thống tệp

Sau khi các ổ đĩa đã được chia thành các phân vùng, cần định dạng lại các phân vùng để có thể tổ chức khoảng không gian ổ cứng trên các phân vùng thành tệp, thư mục và cây thư mục. Một cách thức tổ chức khoảng không gian trên phân vùng như vậy gọi là hệ thống tệp. Có thể liệt kê một vài hệ thống tệp phổ biến:

FAT Hệ thống tệp đơn giản của Microsoft. Không có tính năng bảo mật.

NTFS Hệ thống tệp của Windows NT. Có bảo mật và kiểm soát truy cập.

EXT2 Hệ thống tệp của Linux, có kiểm soát truy cập và xác thực

EXT3 Hệ thống tệp của Linux, có nhật ký, có kiểm soát truy cập và xác thực.

EXT4 Hệ thống tệp của Linux, có nhật ký, mã hóa, có kiểm soát truy cập và xác thực.

Hình 10.1.2 Trong các hệ thống Linux đơn giản, không đòi hỏi độ tin cậy và bảo mật cao, có thể lựa chọn EXT2. Với các hệ thống đòi hỏi bảo mật, độ tin cậy cao nên sử dụng các định dạng EXT3 và EXT4. Hệ thống Linux mặc định sử dụng một phân vùng được định dạng riêng để làm bộ nhớ ảo. Định dạng này gọi là định dạng SWAP.

10.2 Quản lý đĩa và phân vùng

10.2.1 Thao tác trên phân vùng

Các thao tác cơ bản được thực hiện trên phân vùng là: hiển thị thông tin về các phân vùng, xóa phân vùng, thay đổi cấu hình của phân vùng, tạo các phân vùng mới,ghi các thay đổi vào MBR và nhiều chức năng khác. Ngoài ra, sau khi phân vùng được tạo ra và đặt các tham số phù hợp, cần định dạng phân vùng.

Các công cụ thường được sử dụng để thực hiện các thao tác trên trong Linux là pdisk, fdisk, gparted, sfdisk. Với giao diện text, công cụ fdisk là

```
 0  Empty           1c  Hidden Win95 FA 70  DiskSecure Mult bb  Boot Wizard hid
 1  FAT12           1e  Hidden Win95 FA 75  PC/IX           be  Solaris boot
 2  XENIX root      24  NEC DOS         80  Old Minix       c1  DRDOS/sec (FAT-
 3  XENIX usr       39  Plan 9          81  Minix / old Lin c4  DRDOS/sec (FAT-
 4  FAT16 <32M      3c  PartitionMagic  82  Linux swap      c6  DRDOS/sec (FAT-
 5  Extended        40  Venix 80286     83  Linux           c7  Syrinx
 6  FAT16           41  PPC PReP Boot   84  OS/2 hidden C:  da  Non-FS data
 7  HPFS/NTFS       42  SFS             85  Linux extended  db  CP/M / CTOS / .
 8  AIX             4d  QNX4.x          86  NTFS volume set de  Dell Utility
 9  AIX bootable    4e  QNX4.x 2nd part 87  NTFS volume set df  BootIt
 a  OS/2 Boot Manag 4f  QNX4.x 3rd part 8e  Linux LVM       e1  DOS access
 b  Win95 FAT32     50  OnTrack DM      93  Amoeba          e3  DOS R/O
 c  Win95 FAT32 (LB 51  OnTrack DM6 Aux 94  Amoeba BBT      e4  SpeedStor
 e  Win95 FAT16 (LB 52  CP/M            9f  BSD/OS          eb  BeOS fs
 f  Win95 Ext'd (LB 53  OnTrack DM6 Aux a0  IBM Thinkpad hi ee  EFI GPT
10  OPUS            54  OnTrackDM6      a5  FreeBSD         ef  EFI (FAT-12/16/
11  Hidden FAT12    55  EZ-Drive        a6  OpenBSD         f0  Linux/PA-RISC b
12  Compaq diagnost 56  Golden Bow      a7  NeXTSTEP        f1  SpeedStor
14  Hidden FAT16 <3 5c  Priam Edisk     a8  Darwin UFS      f4  SpeedStor
16  Hidden FAT16    61  SpeedStor       a9  NetBSD          f2  DOS secondary
17  Hidden HPFS/NTF 63  GNU HURD or Sys ab  Darwin boot     fd  Linux raid auto
18  AST SmartSleep  64  Novell Netware  b7  BSDI fs         fe  LANstep
1b  Hidden Win95 FA 65  Novell Netware  b8  BSDI swap       ff  BBT
```

Hình 10.1.2: Các định dạng hệ thống tệp

công cụ thường được sử dụng. Fdisk cho phép NSD có thể quản lý các phân vùng theo cách tương tác với chương trình, hoặc thực hiện trực tiếp bằng câu lệnh. Hình 10.2.3 mô tả giao diện của fdisk. Để tạo ra một phân vùng mới với fdisk cần chọn n, sau đó chọn block đầu tiên của phân vùng và kích thước phân vùng, cuối cùng là kiểu của phân vùng. Có thể thực hiện các thao tác xóa, thêm phân vùng nhiều lần trước khi ghi các thay đổi lên MBR của ổ cứng. Chỉ khi ghi vào MBR, các thay đổi mới có hiệu lực.

10.2.2 Thao tác trên hệ thống tệp

Sau khi các phân vùng đã được tạo ra, cần định dạng phân vùng theo một trong các hệ thống tệp. Câu lệnh mkfs cho phép thực hiện thao tác này. Tùy biến -t cho phép khai báo kiểu hệ thống tệp cần định dạng. Câu lệnh mkfs sẽ gọi các lệnh mk2fs, mkfs.ext2, mkfs.ext3, mkfs.ext4, mkfs.vfat, mkfs.ntfs, ... Định dạng ext3 thực chất là định dạng ext2 thêm một nhật ký của hệ thống tệp. Nhật ký này cho phép các thao tác kiểm tra và phục hồi lỗi có thể được thực hiện với tốc độ nhanh hơn. Hình 10.2.4 mô tả một ví dụ định dạng phân vùng, đầu tiên là theo định dạng ext2.

Chú ý là với định dạng tệp Linux, 5% không gian phân vùng được dự trữ. Khi 95% không gian bộ nhớ được dùng, NSD thông thường không được dùng tiếp không gian trống, chỉ có superuser (root) được quyền truy cập vào phân vùng. Phân vùng sẽ được sử dụng 38 lần hoặc 180 ngày, sau đó hệ thống sẽ tự động kiểm tra tính toàn vẹn của phần vùng.

Để thay đổi các tham số của phân vùng ext sau khi định dạng, có thể sử dụng câu lệnh tune2fs. Câu lệnh tune2fs có thể thay đổi kích thước block, số lượng inodes, giới hạn số lần sử dụng trước khi kiểm tra tính toàn vẹn của đĩa. Khi thao tác với tune2fs cần tiến hành sao lưu dữ liệu chứa trên phân vùng, tránh trường hợp mất mát dữ liệu. Các tham số có thể là:

- -b kích thước block
- -i số lượng byte cho 1 inode
- -c Số lần mount
- -j Có nhật ký?
- -m dự trữ
- -r số block dự trữ
- -g, -u nhóm và NSD được dùng dự trữ

```
[root@localhost root]# fdisk /dev/sda
The number of cylinders for this disk is set to 1044.
There is nothing wrong with that, but this is larger than 10
24,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LI
LO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): m
Command action
   a   toggle a bootable flag
   b   edit bsd disklabel
   c   toggle the dos compatibility flag
   d   delete a partition
   l   list known partition types
   m   print this menu
   n   add a new partition
   o   create a new empty DOS partition table
   p   print the partition table
   q   quit without saving changes
   s   create a new empty Sun disklabel
   t   change a partition's system id
   u   change display/entry units
   v   verify the partition table
   w   write table to disk and exit
   x   extra functionality (experts only)
Command (m for help): p

Disk /dev/sda: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/sda1   *         1        13    104391   83  Linux
/dev/sda2            14       914   7237282+  83  Linux
/dev/sda3           915      1044   1044225   82  Linux swap
```

Hình 10.2.3: Các câu lệnh cơ bản của fdisk

```
[root@localhost root]# mkfs -t ext2 /dev/sdb2
mke2fs 1.32 (09-Nov-2002)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
51200 inodes, 204784 blocks
10239 blocks (5.00%) reserved for the super user
First data block=1
25 block groups
8192 blocks per group, 8192 fragments per group
2048 inodes per group
Superblock backups stored on blocks:
        8193, 24577, 40961, 57345, 73729
Writing inode tables: done
Writing superblocks and filesystem accounting information: done
This filesystem will be automatically checked every 38 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

Hình 10.2.4: Định dạng hệ thống tệp

Để có thể sử dụng phân vùng đã được định dạng, cần gắn phân vùng này vào cây thư mục của hệ thống. Chú ý là mặc dù phân vùng được biểu diễn bởi một tệp, tuy nhiên, các thao tác đọc và ghi trên tệp này thuần túy là các thao tác đọc và ghi các khối dữ liệu thô trên phân vùng, không liên quan gì để tổ chức của hệ thống tệp trên phân vùng. Câu lệnh cần sử dụng là mount, với 2 tham số cơ bản là hệ thống tệp cần sử dụng và thư mục để gắn hệ thống tệp này vào. Ngoài ra cần chỉ rõ kiểu của hệ thống tệp trong trường hợp không phải là ext (-t), và có thể cần thêm một số các thông số khác (-o) để có thể sử dụng được hệ thống tệp. Với câu lệnh mount, không những có thể sử dụng được các phân vùng trên ổ đĩa vật lý, mà còn có thể sử dụng được các tệp thông thường như là các ổ đĩa ảo (Hình 10.2.5

10.2.3 Quản lý bộ nhớ ảo

Việc định dạng các phân vùng bộ nhớ ảo có một vài điểm khác biệt so với các định dạng khác. Để định dạng một phân vùng thành swap sử dụng câu lệnh mkswap:

```
[root@localhost root]# mkswap /dev/sdb2
Setting up swapspace version 1, size = 209694 kB
```

```
[root@localhost root]# mount /dev/sdb2 /temp
[root@localhost root]# mount
/dev/sda2 on / type ext3 (rw)
none on /proc type proc (rw)
usbdevfs on /proc/bus/usb type usbdevfs (rw)
/dev/sda1 on /boot type ext3 (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
none on /dev/shm type tmpfs (rw)
/test.img on /temp type ext2 (rw,loop=/dev/loop0,usrquota,gr
/dev/sdb2 on /temp type ext2 (rw)
[root@localhost root]# cat /etc/mtab
/dev/sda2 / ext3 rw 0 0
none /proc proc rw 0 0
usbdevfs /proc/bus/usb usbdevfs rw 0 0
/dev/sda1 /boot ext3 rw 0 0
none /dev/pts devpts rw,gid=5,mode=620 0 0
none /dev/shm tmpfs rw 0 0
/test.img /temp ext2 rw,loop=/dev/loop0,usrquota,grpquota 0
/dev/sdb2 /temp ext2 rw 0 0
[root@localhost root]# umount /temp
```

Hình 10.2.5: Sử dụng câu lệnh mount và tác động lên tệp mtab

Rồi sau đó dùng câu lệnh swapon để hệ thống sử dụng:

```
[root@localhost root]# swapon /dev/sdb2
[root@localhost root]# swapon -s
Filename                        Type            Size    Used
/dev/sda3                       partition       1044216 0
/dev/sdb2                       partition       204776  0
```

Có thể cấu hình để sử dụng một tệp như là swap file:

```
[root@localhost root]# mkswap /test.img
Setting up swapspace version 1, size = 511995 kB
[root@localhost root]# swapon /test.img
[root@localhost root]# swapon -s
Filename                        Type            Size    Used
/dev/sda3                       partition       1044216 0
/dev/sdb2                       partition       204776  0
/test.img                       file            499992  0
```

Với các phân vùng được sử dụng thường xuyên, có thể khai báo để hệ thống tự động mount khi khởi động. Tệp /etc/fstab được sử dụng vào mục đích này. Tệp /etc/fstab có nhiều dòng, mỗi dòng có nhiều trường có ý nghĩa là:

```
[root@localhost root]# cat /etc/fstab
LABEL=/ /                       ext3    defaults        1 1
LABEL=/boot     /boot                   ext3    defaults        1 2
none            /dev/pts                devpts  gid=5,mode=620  0 0
none            /proc                   proc    defaults        0 0
none            /dev/shm                tmpfs   defaults        0 0
/dev/sda3       swap                    swap    defaults        0 0
/dev/cdrom      /mnt/cdrom              udf,iso9660 noauto,owner,kudzu,ro 0 0
/dev/fd0        /mnt/floppy             auto    noauto,owner,kudzu 0 0
/test.img       /temp/                  ext2 loop,rw,usrquota,grpquota,d
```

Trường đầu tiên của mỗi dòng là tệp phân vùng. Có một số trường hợp sử dụng trực tiếp các cơ chế của nhân. Trường thứ 2 chỉ ra thư mục để gắn tệp. Trường thứ 3 chỉ ra kiểu hệ thống tệp. Sau đó là tùy biến để mount và cuối cùng là các trường liên quan đến sao lưu và kiểm tra lỗi.

10.3 Quản lý hạn ngạch

Khi một phân vùng được sử dụng chung bởi nhiều NSD, cần có một chính sách để có thể hạn chế trường hợp phân vùng bị đầy, không NSD nào có thể

truy cập được. Chính sách này cũng cần đảm bảo việc sử dụng phân vùng thuận tiện cho NSD, giảm thiểu nhu cầu liên hệ với quản trị viên để "qiải quyết". Công cụ sử dụng để cài đặt chính sách trên là hạn ngạch.

10.3.1 Khái niệm hạn ngạch

Hạn ngạch là một tập hợp các hạn chế về tài nguyên không gian ổ đĩa với NSD. Linux thực hiện việc quản lý hạn ngạch theo phân vùng-hệ thống tệp. Linux phân biệt hạn ngạch cho NSD và hạn ngạch cho nhóm NSD. Linux giới hạn NSD về số lượng các tệp có thể tạo ra (inodes) và dung lượng đĩa có thể sử dụng (blocks). Với mỗi một loại tài nguyên và NSD hoặc nhóm NSD, Linux có 2 giới hạn: giới hạn mềm và giới hạn cứng. Khi NSD vượt quá giới hạn mềm, hệ thống sẽ cảnh báo NSD, tuy nhiên vẫn để NSD có thể sử dụng phân vùng. Trong trường hợp này, NSD cần tận dụng quyền truy cập này để dọn dẹp và thoát khỏi trạng thái cảnh báo Nếu NSD vượt quá mức giới hạn cứng, mặc định NSD sẽ không truy cập được vào phân vùng và cần liên hệ với quản trị viên để có thể truy cập lại và giải phóng không gian đĩa. Nếu sau một thời gian nằm trong trạng thái cảnh báo mà NSD không thực hiện các thao tác dọn dẹp để thoát khỏi tình trạng này, NSD sẽ không truy cập được vào phân vùng. Thời gian này gọi là thời gian ân hạn (grace time) được dùng chung cho tất cả NSD hoặc nhóm NSD Thông thường tất cả các thông tin này được lưu trong một CSDL đặc biệt nằm ở thư mục gốc của phân vùng.

10.3.2 Thiết lập hạn ngạch

Để thiết lập hạn ngạch, cần cài đặt phần mềm quota

```
#yum install quota
```

sau đó kích hoạt chế độ sử dụng hạn ngạch khi sử dụng phân vùng. Các tùy biến usgquota và grpquota cho phép mount một phân vùng với hỗ trợ hạn ngạch:

```
# mount -o usrquota,grpquota /dev/sdb2 /temp
```

rồi tạo ra một CSDL về hạn ngạch Với Linux, CSDL cho NSD là aquota.user, cho nhóm NSD là aquota.group.

```
#touch /temp/aquota.user
#touch /temp/aquota.group
# ls -la /temp
total 17
```

```
drwxr-xr-x    3 root     root         1024 Apr 25 04:20 .
drwxr-xr-x   23 root     root         4096 Apr 24 07:24 ..
-rw-r--r--    1 root     root            0 Apr 25 04:20 aquota.group
-rw-r--r--    1 root     root            0 Apr 25 04:20 aquota.user
drwx------    2 root     root        12288 Apr 25 04:16 lost+found
```

2 tệp mới tạo ra chỉ là 2 tệp rỗng, không chứa CSDl dữ liệu về hạn ngạch. Để khởi tạo CSDL trên 2 tệp này, sử dụng câu lệnh quotacheck. Đây là câu lệnh để kiểm tra tính toàn vẹn của các tệp CSDl quota, tuy nhiên, câu lệnh này có hiệu ứng phụ là khi các tệp này bị lỗi, nó sẽ tự sửa các tệp nếu có thể.

```
[root@localhost root]# quotacheck -ug /temp
quotacheck: WARNING -  Quotafile /temp/aquota.user was probably truncated.
 save quota settings...
quotacheck: WARNING -  Quotafile /temp/aquota.group was probably truncated
t save quota settings...
[root@localhost root]# ls -la /temp
total 29
drwxr-xr-x    3 root     root         1024 Apr 25 04:27 .
drwxr-xr-x   23 root     root         4096 Apr 24 07:24 ..
-rw-r--r--    1 root     root         6144 Apr 25 04:27 aquota.group
-rw-r--r--    1 root     root         6144 Apr 25 04:27 aquota.user
drwx------    2 root     root        12288 Apr 25 04:16 lost+found
```

Động tác cuối cùng là kích hoạt hạn ngạch trên phân vùng:

```
#quotaon /temp
[root@localhost root]# repquota -ug /temp
*** Report for user quotas on device /dev/sdb2
Block grace time: 7days; Inode grace time: 7days
                        Block limits                File limits
User            used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root      --      13       0       0              4     0     0


*** Report for group quotas on device /dev/sdb2
Block grace time: 7days; Inode grace time: 7days
                        Block limits                File limits
Group           used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root      --      13       0       0              4     0     0
```

10.3.3 Sử dụng hạn ngạch

Để xem thông tin về hạn ngạch trên một hệ thống tệp đang sử dụng, dùng câu lệnh repquota. Các tùy biến -u, -g cho phép hiển thị các thông tin về quota của NSD hoặc của nhóm NSD. Tùy biến -a cho phép hiển thị thông tin về tất cả các hệ thống tệp đang sử dụng. Để có thể thay đổi quota của người sử dụng, có thể sử dụng công cụ edquota để gọi một chương trình soạn thảo của hệ thống, cập nhật các thông tin của NSD, cũng có thể sử dụng câu lệnh setquota. Ví dụ

```
[root@localhost root]# setquota -u trunghq BS BH IS IH /temp
```

trong đó các tham số lần lượt là Block soft limit, hard limit, Inode soft and hard limit. Trường hợp NSD chưa sử dụng phân vùng lần nào, các thay đổi sẽ được cập nhật sau khi NSD sử dụng phân vùng thành công lần đầu tiên và các tệp quota được kiểm tra.

10.4 Bài tập

Bài tập 10.1 *Tạo một tệp kích thước 20M. Định dạng và mount tệp vào thư mục /temp trên hệ thống. Cấu hình để tệp được mount tự động.*

Bài tập 10.2 *Cấu hình hệ thống để chỉ sử dụng swapfile.*

Chương 11

Sao lưu

11.1 Quá trình sao lưu

11.1.1 Vì sao phải sao lưu

Các hệ thống máy tính được dùng để xử lý dữ liệu. Dữ liệu trong quá trình xử lý được lưu trữ tạm thời trong bộ nhớ điều hành, trong bộ nhớ bền vững. Các bộ nhớ bền vững (các thiết bị lưu trữ) cho phép dữ liệu có thể được lưu trữ bền vững và lâu dài, thực hiện các thao các có liên quan với nhau trong khoảng thời gian dài. Trong suốt thời gian dữ liệu được lưu trữ, có thể có những tác động làm thay đổi, mất mát dữ liệu được lưu trữ.

Một số hệ thống máy tính cung cấp các dịch vụ đòi hỏi tính sẵn sàng cao. Trường hợp hệ thống máy tính này bị sự cố, cần có một hệ thống khác có thể được kích hoạt để thay thế hệ thông này. Có thể kể ra một số dạng rủi ro dẫn đến tê liệt hệ thống và mất mát dữ liệu:

Tai nạn tự nhiên Các tai nạn tự nhiên như hỏa hoạn, sự cố điện nguồn kéo dài, thường có tác động không chỉ đến dữ liệu và dịch vụ, mà ảnh hưởng đến tất cả các thành phần khác của hệ thống. Tuy nhiên các tai nạn này thường chỉ xảy ra trong một không gian địa lý hẹp.

Lỗi phần cứng Các hệ thống phần cứng sau một thời gian dài hoạt động có thể bị lỗi. Các lỗi này thường dẫn đến việc một thành phần, một hệ thống con nào đó của hệ thống không hoạt động. Dịch vụ do hệ thống cung cấp có thể bị tê liệt, tuy nhiên, khả năng dữ liệu vẫn được bảo tồn là rất cao, nhất là khi sự cố không xảy ra trên thiết bị lưu trữ.

Lỗi phần mềm Các phần mềm có thể chứa các lỗi bên trong. Các lỗi nghiêm trọng có thể làm tê liệt dịch vụ được cung cấp, làm mất mát một số dạng dữ liệu, có thể tạo điều kiện để các loại lỗi khác có thể xảy ra.

Lỗi con người Các lỗi này thường xảy ra dưới 2 dạng: dạng vô ý do lỗi của người chịu trách nhiệm vận hành hệ thống thực hiện không đúng qui trình, thao tác dẫn đến sự cố, dạng cố ý do những người có quyền truy cập vào hệ thống thực hiện các thao tác nhằm gây tác động lên hệ thống. Trong cả hai trường hợp, hậu quả có thể là rất nghiêm trọng, có thể dẫn đến khả năng tê liệt hoàn toàn hệ thống, mất tất cả các dữ liệu.

Chính vì những lý do trên, nên cần tiến hành sao lưu các hệ thống. Sao lưu là thao tác tạo ra một bản sao của hệ thống và lưu trữ an toàn. Khi hệ thống chính bị sự cố, bản sao của hệ thống được sử dụng để phục hồi lại hệ thống.

11.1.2 Các loại sao lưu

Sao lưu có thể được phân loại theo nhiều tiêu chí khác nhau. Nếu theo đối tượng sao lưu, có thể phân loại thành:

- Sao lưu dịch vụ: trên một hệ thống khác, không nhất thiết phải giống hệ thống chính, kích hoạt dịch vụ tương tự như dịch vụ cũ. Nếu vì lý do nào đó, hệ thống cũ không cung cấp được dịch vụ, các yêu cầu thực hiện dịch vụ được chuyển về dịch vụ kích hoạt trên hệ thống mới. Hầu hết các dịch vụ mạng đều thực hiện dạng sao lưu này.

- Sao lưu dữ liệu: sao chép toàn bộ dữ liệu sang vị trí khác. Nếu hệ thống ban đầu bị sự cố, dữ liệu sẽ được copy lại để được tiếp tục sử dụng.

- Sao lưu hệ thống vật lý: thay vì việc sử dụng một hệ thống vật lý, có thể dùng nhiều hệ thống vật lý. Trường hợp có sự cố xảy ra, hệ thống dự trữ sẽ được đưa vào hoạt động thay thế cho hệ thống bị sự cố.

Căn cứ theo vị trí và cách thức sao lưu:

- Sao lưu nóng: khi hệ thống chính bị sự cố, hệ thống sao lưu được đưa vào hoạt động ngay, thay thế cho hệ thống cũ. Sao lưu nóng đòi hỏi có sự cập nhật thường xuyên giữa hệ thống được sao lưu và bản sao.

- Sao lưu nguội: Các dữ liệu cần thiết cho việc phục hồi hệ thống được sao lưu. Khi có sự cố xảy ra, hệ thống tạm dừng hoạt động. Sau khi quá trình phục hồi được hoàn thành, hệ thống lại được đưa vào hoạt động.

Trong thực tế khi sử dụng HĐH Linux, các quản trị viên thường chia ra các loại sao lưu sau đây

- Sao lưu tệp và thư mục.

- Sao lưu phân vùng và ổ đĩa.

- Sao lưu toàn bộ.

- Sao lưu tăng dần.

- Sao lưu một phần.

```
tar -cvpf /archive/full-backup-‘date ’+%d-%B-%Y’‘.tar.gz \
    --directory / --exclude=mnt --exclude=proc .
```

Để có thể điều khiển vị trí của sao lưu, sử dụng câu lệnh mt:

```
mt -f /dev/nst0 rewind
mt -f /dev/nst0 offline
```

Hình 11.2.1: Sao lưu và điểu khiển băng từ

11.2 Sao lưu tệp và thư mục

11.2.1 Sao lưu với câu lệnh tar

Câu lệnh tar cho phép tạo ra một tệp lưu trữ từ các tệp hoặc thư mục đầu vào với khả năng lưu trữ các thông tin về đường dẫn tương đối, tuyệt đối cũng như về quyền sử dụng của các tệp và thư mục. Các thư mục được sao lưu là các thư mục chứa các dữ liệu cần thiết. Câu lệnh tar cho phép loại bỏ không sao lưu các thư mục không cần thiết như /proc, /var, ... Thiết bị lưu trữ thông thường là các băng từ, là dạng các thiết bị ký tự, do đó việc định dạng hệ thống tệp trên các thiết bị này là không khả thi. Các thiết bị này thường được coi như một tệp. Ví dụ về sao lưu thư mục home vào băng từ.

```
tar -zcvpf /archive/full-backup-‘date ’+%d-%B-%Y’‘.tar.gz \
    --directory / --exclude=mnt --exclude=proc .
```

Tùy biến p cho thấy các quyền của các tệp sẽ được bảo lưu khi lưu trữ, tùy biến –directory lưu trữ các tệp theo đường dẫn tuyệt đối. Tùy biến z thực hiện việc nén khi lưu trữ. Thông thường trên băng từ không thực hiện việc nén dữ liệu, để đảm bảo nếu có một vài ký tự bị hỏng, các tệp khác vẫn có thể đọc được. Quá trình sao lưu khi đó được thực hiện bằng câu lệnh:

11.2.2 Phục hồi với câu lệnh tar

Quá trình phục hồi diễn ra ngược với quá trình sao lưu. Tuy nhiên cần rất cẩn thận khi thực hiện quá trình phục hồi. Các tùy biến của quá trình sao lưu và các tùy biến của quá trình phục hồi có thể phối hợp với nhau theo nhiều cách, cho phép phục hồi theo các cách thức khác nhau, có thể rất khó kiểm soát.

Tùy biến t cho phép mô phỏng quá trình giải nén, không tác động đến các dữ liệu đang tồn tại.

Có thể thực hiện câu lệnh tar để giải nén một hay nhiều tệp (không phải tất cả).

Giải pháp an toàn là phục hồi các tệp vào một thư mục khác, sau đó so sánh và cập nhật các tệp vào thư mục cần thiết.

```
tar -tzvf /archive/full-backup-01-04-2010.tar.gz
tar -tzvf /archive/full-backup-01-04-2010.tar.gz trunghq
```

11.3 Sao lưu phân vùng và ổ đĩa

11.3.1 Câu lệnh dd

Câu lệnh dd là câu lệnh cho phép sao chép 2 tệp bất kỳ, kể cả các tệp thiết bị. Do đó, ngoài việc dùng để sao chép các tệp, câu lệnh dd còn có thể được thực hiện để sao chép ổ đĩa, phân vùng vào các tệp và ngược lại. Điểm yếu lớn nhất của việc sao chép bằng dd là thực hiện sao chép ở mức thấp, theo từng block, do đó quá trình sao lưu chậm và không phát hiện được các vùng dữ liệu không sử dụng. Ví dụ

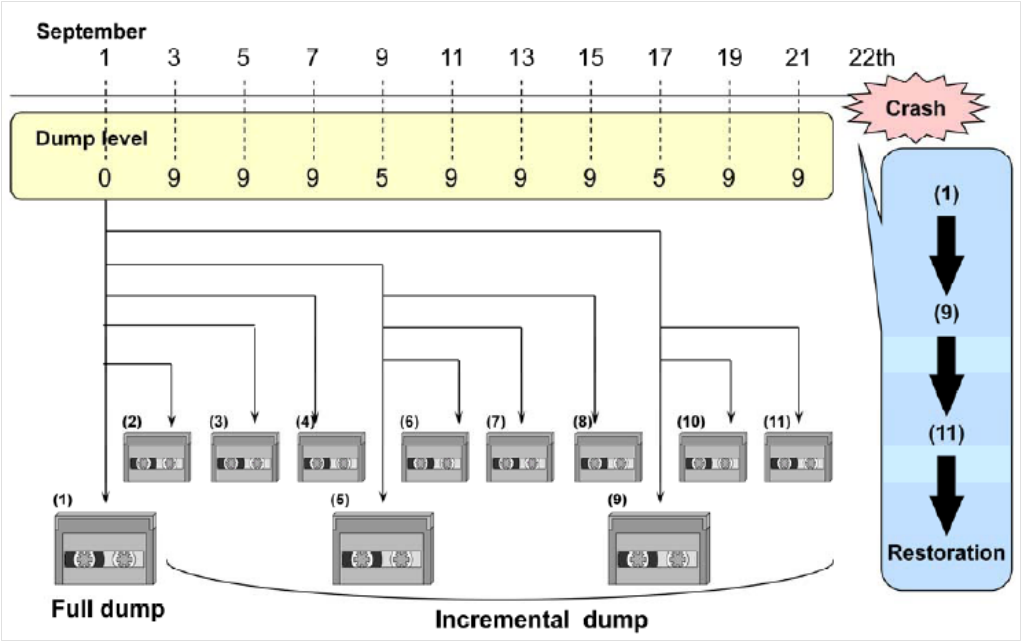
```
dd count=xxx if=/dev/hda of=/dev/hdb2
dd count=xxx if=/dev/hda1 of=/dev/hdb2
dd count=xxx if=/dev/hda of=/dev/hdb
dd count=xxx if=/dev/hda1 of=/dev/hdb1
dd count=xxx if=/dev/hda of=/dev/f1
dd count=xxx if=/dev/f1 of=/dev/hda
```

Khi phục hồi từ dd, dữ liệu trên ổ đĩa/phân vùng đích sẽ bị ghi đè không báo trước, vì vậy cần hết sức cẩn thận.

11.3.2 dump và restore

Câu lệnh dump dùng để sao chép một tệp (thường là một phân vùng) vào một tệp lưu trữ kèm theo lịch sử thực hiện lưu trữ trên tệp hoặc thư mục đó. Câu lệnh restore là câu lệnh cho phép phục hồi lại phân vùng/ổ đĩa từ bản sao lưu mới nhất. Để thực hiện câu lệnh dump, các ổ đĩa và phân vùng cần sao lưu phải ở trong trạng thái rỗi, không bị sử dụng. Để đảm bảo điều này, thông thường hệ thống được chuyển về mức thực hiện 1, các phân vùng được umount và kiểm tra lỗi trước khi thực hiện sao lưu:

```
# init 1
# umount /home
fsck -aV /dev/hda6
```



Hình 11.3.2: Các mức thực hiện dump

Sau đó có thể sử dụng lênh dump để sao lưu

```
(a)# dump 0uf /dev/st0 /dev/hda6
(b)# dump 5uf /dev/st0 /dev/hda6
(c)# dump 9uf /dev/nst0 /dev/hda6
```

11.3.3 Các mức sao lưu

Để có thể xác định sự phụ thuộc của các bản sao, dump sử dụng các mức sao lưu. Khi câu lệnh dump được thực hiện bản sao sẽ được tạo ra dựa trên các thay đổi so với bản sao mới nhất có mức sao lưu nhỏ hơn mức sao lưu đang được thực hiện. Việc đưa vào mức sao lưu cho phép thực hiện sao lưu với tần suất cao hơn mà vẫn tiết kiệm không gian lưu trữ. Trong hình vẽ, sau khi thực hiện sao lưu toàn phần, các bản sao mức 9 của các ngày tiếp theo được thực hiện dựa trên bản sao lưu toàn phần này. Như vậy các bản sao lưu của các ngày tiếp theo chỉ chứa các thay đổi từ ngày thực hiện sao lưu toàn phần. Bản sao của ngày thứ 3 sẽ có kích thước lớn nhất. Sau 3 ngày, thực hiện sao lưu ở mức 5 để bền vững các thay đổi trong 3 ngày trước trong sao lưu mức 5. 3 ngày tiếp theo tiếp tục thực hiện sao lưu mức 9. Ngày thứ 4 tiếp tục thực hiện sao lưu mức 5. Sau 4 lần thực hiện sao lưu mức 5 có thể thực hiện các sao lưu mức thấp hơn hoặc thậm chí là sao lưu toàn phần.

Với cách thức sao lưu như trên, thời gian và khối lượng sao lưu mỗi lần giảm xuống đáng kể, tuy nhiên hệ thống vẫn đảm bảo đồng bộ giữa các dữ liệu sao lưu và dữ liệu thực trên hệ thống.

11.3.4 Phục hồi

Câu lệnh restore cho phép phục hồi từ các bản sao được tạo ra bởi lệnh dump. Câu lệnh cho phép xem nội dung bản sao, phục hồi lên các thư mục khác nhau, phục hồi một số các tệp hoặc thư mục hoặc phục hồi một cách tương tác với NSD.

```
# restore rf /dev/st0
# restore rf /dev/st0
# restore cf /dev/st0 .x/usr00
# restore if /dev/st0
```

Quá trình phục hồi được thực hiện ngược với quá trình sao lưu. Đầu tiên các sao lưu mức thấp được phục hồi, sau đó là các sao lưu mức cao hơn. Tuy nhiên, các thao tác này được thực hiện tự động, câu lệnh phục hồi chỉ cần khai báo tệp lưu trữ là đủ.

```
#  mkfs /dev/hda6
#  fsck -aV /dev/hda6
#  mount  /dev/hda6   /home
# cd /home #  cd /home
#  restore  rf /dev/st0
#  rm restoresymtable
```

11.3.5 Kịch bản sao lưu

Kịch bản sao lưu đóng vai trò rất quan trọng trong việc sao lưu và phục hồi. Kịch bản này cần được xây dựng dựa trên yêu cầu sử dụng của hệ thống. Thông thường, để có bản sao lưu toàn phần cần một thời gian khá dài, do đó sao lưu toàn phần cần được thực hiện trong các ngày nghỉ (cuối tuần). Hàng ngày, thời gian có thể tiến hành sao lưu là tương đối hạn hẹp, do đó chỉ có thể tiến hành sao lưu các thay đổi trong ngày. Phụ thuộc nhu cầu dữ liệu có thể gộp các sao lưu ngày lại vào một sao lưu mức thấp hơn trong tuần.

Tháng Năm 09						
Thứ Hai	Thứ Ba	Thứ Tư	Thứ Năm	Thứ Sáu	Thứ Bảy	Chủ Nhật
Tháng Tư 27	28	29	30	Tháng Năm 1	2	3
Dump mức 4	Dump mức 5	Dump mức 6	Dump mức 7	Dump mức 8	Dump mức 9	Dump mức 0
4	5	6	7	8	9	10
Dump mức 4	Dump mức 5	Dump mức 6	Dump mức 7	Dump mức 8	Dump mức 9	Dump mức 1
11	12	13	14	15	16	17
Dump mức 4	Dump mức 5	Dump mức 6	Dump mức 7	Dump mức 8	Dump mức 9	Dump mức 2
18	19	20	21	22	23	24
Dump mức 4	Dump mức 5	Dump mức 6	Dump mức 7	Dump mức 8	Dump mức 9	Dump mức 3
25	26	27	28	29	30	31
Dump mức 4	Dump mức 5	Dump mức 6	Dump mức 7	Dump mức 8	Dump mức 9	Dump mức 0

Hình 11.3.3: Kịch bản sao lưu

Chương 12

Nhật ký

12.1 Khái niệm ghi nhật ký

Trong quá trình hoạt động của hệ thống, cần thiết phải ghi chép, lưu trữ lại các thông tin về hoạt động của hệ thống. Các thông tin này sẽ giúp ích cho việc giải quyết sự cố, phát hiện các sai sót, các hành động xâm nhập, tác động vào hệ thống. Về lý thuyết, bất cứ một thông tin nào về hoạt động của hệ thống cũng có thể giúp ích cho quá trình phát hiện và khắc phục các sự cố. Thực tế, việc ghi chép theo dõi các hệ thống hoặc là quá sơ sài, không đủ thông tin giúp ích cho quản trị viên, hoặc quá chi tiết, dẫn đến các thông tin không có giá trị.

Các thông tin cần ghi chép có thể được phân loại theo nhiều cách, phụ thuộc vào việc thông tin này được sinh ra từ đâu, được lọc thế nào và được ghi vào vị trí nào. Nếu theo nguồn gốc sản sinh ra thông tin có thể chia ra:

- Thông tin về hoạt động của các thiết bị phần cứng.
- Thông tin về hoạt động của nhân hệ điều hành.
- Thông tin về hoạt động của các dịch vụ hệ thống.
- Thông tin về hoạt động của các ứng dụng.
- Thông tin về các thao tác của NSD.

Căn cứ vào mức độ quan trọng của thông tin, có thể phân loại thành

- Mức thông tin chi tiết: các thông tin chi tiết liên quan đến hoạt động nội bộ của các tiến trình, không phản ánh việc hoạt động của hệ thống có bình thường hay không. Thường các thông tin này phục vụ cho các nhà phát triển chương trình để theo dõi hoạt động của chương trình trong giai đoạn phát triển.
- Mức thông tin dịch vụ: các thông tin liên quan đến hoạt động bình thường của tiến trình, chủ yếu là các hoạt động có ảnh hưởng đến các tiến trình khác.
- Thông tin cảnh báo: Thông tin về hoạt động bất bình thường của các tiến trình. Các hoạt động bất bình thường này chưa làm ảnh hưởng đến các tiến trình khác.
- Thông tin báo lỗi: Thông tin về lỗi xảy ra trong tiến trình, cho biết một hoặc nhiều các chức năng quan trọng của tiến trình không thể được đảm bảo, ảnh hưởng đến một số các tiến trình có liên quan.

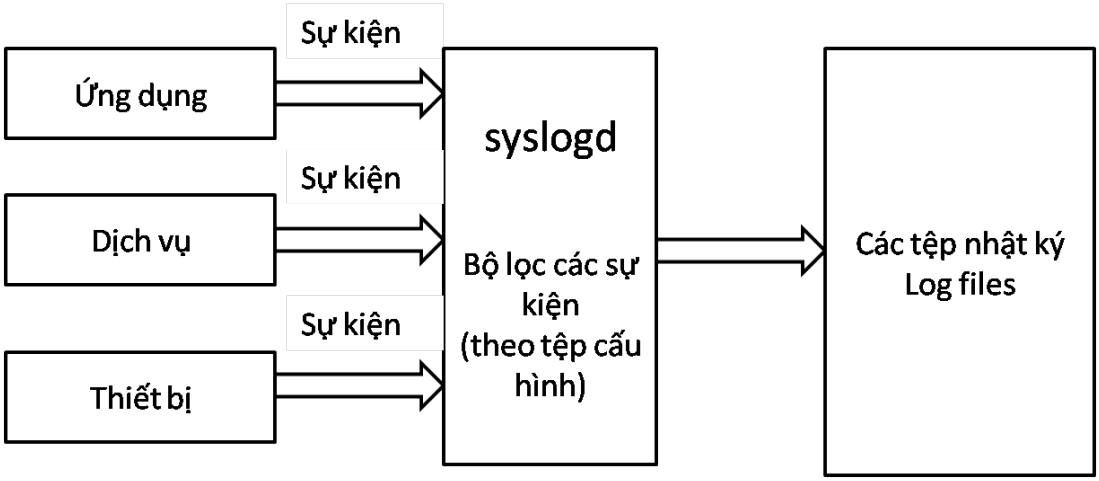
- Thông tin lỗi hệ thống: Lỗi nghiêm trọng trong các tiến trình hệ thống, tất cả các tiến trình trong hệ thống đều bị ảnh hưởng.

Các phần mềm khi được thiết kế đều có tính năng ghi nhật ký riêng, cho phép lọc và lưu trữ các sự kiện xảy ra trong quá trình thực hiện vào một thư mục riêng. Cách thức này thường được sử dụng cho các phần mềm tương đối độc lập, không có nhiều các phần mềm khác phụ thuộc. Trường hợp các phần mềm phụ thuộc lẫn nhau, việc từng phần mềm ghi nhật ký riêng sẽ gây khó khăn cho viêc theo dõi một cách tổng thể các sự kiện xảy ra trong hệ thống. Với các phần mềm hệ thống, thông tin về các sự kiện xảy ra sẽ được chuyển đến một tiến trình phụ trách về việc ghi nhật ký. Tiến trình này sẽ thực hiện việc lọc và ghi nhật ký cho phù hợp. Với cách tiếp cận này, quản trị viên có thể cấu hình hệ thống để cho các nhật ký của các phần mềm có liên quan có thể được ghi vào cùng một chỗ, thuận tiện cho việc phát hiện các kịch bản sự kiện đã xảy ra của các phần mềm phụ thuộc lẫn nhau.

12.2 Cơ chế ghi nhật ký trong Linux

Trong hệ điều hành Linux, nhật ký được ghi chép theo cả 2 cách tiếp cận nói trên. Một số phần mềm độc lập (apache, postfix, ...) thực hiện ghi nhật ký trực tiếp và tự quản lý các tệp nhật ký của mình. Một số phần mềm khác quản lý các sự kiện xảy ra trong phần mềm và gửi các thông báo đến một tiến trình quản lý chung (syslogd). Căn cứ vào cấu hình nhật ký do quản trị viên xác lập, syslogd sẽ tiến hành ghi các thông tin về các sự kiện xảy ra vào các tệp tương ứng. Cơ chế trao đổi thông tin giữa các tiến trình khác với syslogd được mô tả trong Hình 12.2.1. syslogd luôn sẵn sàng chờ các thông báo từ các ứng dụng khác trên cổng 514. Sau khi nhận được thông báo, căn cứ vào cấu hình của mình, syslogd lọc và ghi các thông báo vào các tệp đầu ra phù hợp. Ví dụ về cấu hình bộ lọc được chỉ ra trong Hình 12.2.2. Tệp cấu hình của syslogd gồm nhiều dòng, mỗi dòng gồm 2 trường phân cách nhau bằng dấu cách. Mỗi dòng mô tả một thao tác mà syslogd sẽ thực hiện khi thông báo thỏa mãn các tiêu chí được mô tả trong trường thứ nhất. Cú pháp của trường thứ nhất có dạng xxx.yyy, trong đó xxx là nguồn của thông báo (facility), yyy là mức độ ưu tiên xử lý của thông báo (priority). Linux phân biệt các loại facility sau:

- auth: các thông báo có liên quan đến bảo mật của hệ thống.
- authpriv: các thông báo liên quan đến kiểm soát truy cập.
- deamon: các thông báo từ các tiến trình hệ thống và các chương trình chạy thường trú (daemon).



Hình 12.2.1: Cơ chế hoạt động của syslogd

```
# Sample syslog.conf file that sorts messages by
# mail, kernel, and "other", and broadcasts
# emergencies to all logged-in users

# print most sys. events to tty10 and to the xconsole
# pipe, and emergencies to everyone

kern.warn;*.err;authpriv.none   |/dev/xconsole
*.emerg                         *

# send mail, news (most), and kernel/firewall msgs to
# their respective log files
mail.*                  -/var/log/mail
kern.*                  -/var/log/kernel_n_firewall

# save the rest in one file
*.*;mail.none           -/var/log/messages
```

Hình 12.2.2: Tệp cấu hình của syslogd

- kern: thông báo từ nhân hệ điều hành
- mark: thông báo từ syslogd, dùng để đảm bảo thứ tự thời gian.
- user: thoogn báo đến từ các ứng dụng.
- local7: thông báo do quá trình khởi động sinh ra.
- *: bất cứ thông báo nào.
- none: không có thông báo nào.

Có 7 mức độ ưu tiên khác nhau của các thông báo:

- debug: Các thông báo chỉ dành cho việc gỡ lỗi trogn từng chương trình. Có thể không sử dụng đối với các hệ thống đang vận hành
- info: Các thông tin về hoạt động bình thường của các tiến trình. Phục vụ cho việc phát hiện các bất thường không thường xuyên xảy ra.
- notice: Thông tin về các hoạt động bình thường, tuy nhiên cần đặc biệt chú ý
- warning: Cảnh báo có khả năng xảy ra lỗi. Thông thường đây là các thông báo về các lỗi phần mềm tự xử lý được, tuy nhiên có thể gây ảnh hưởng tới các tiến trình khác.
- err: Có lỗi xảy ra, không xử lý được.
- crit: Có lỗi nghiêm trọng xảy ra, không xử lý được, ảnh hưởng tới nhiều tiến trình khác.
- alert: Có lỗi nghiêm trọng xảy ra, không xử lý được, chắc chắn sẽ ảnh hưởng tới đa số các tiến trình khác. Khuyến nghị các tiến trình nếu có thể kết thúc công việc để đảm bảo an toàn dữ liệu.
- emerg: Có lỗi nghiêm trọng xảy ra, không xử lý được, không đảm bảo hoạt động của hệ thống.

Các mức ưu tiên có thể phối hợp với các toán tử so sánh và logics như =,!,* và none. Toán tử = được hiểu là tất cả các thông báo có cùng hoặc hơn mức ưu tiên mô tả sau toán tử. ! lọc tất cả những thông báo không thỏa mãn biểu thức ưu tiên sau toán tử. * và none tương ứng với tất cả các mức ưu tiên và không mức ưu tiên nào được lựa chọn.

Trường thứ 2 của mỗi dòng chỉ ra thao tác mà syslogd sẽ thực hiện với thông báo. syslogd cho phép khai báo mềm dẻo các thao tác này. Các thao tác này có thể là:

- Ghi vào một tệp.
- Chuyển đến một đường ống (để thực hiện một lệnh chẳng hạn).
- Chuyển đến một tệp thiết bị.
- Chuyển đến một máy tính ở xa.
- Hiển thị thông báo cho NSD.

Ví dụ

```
mail,uucp.notice;uucp.!=alert     /var/log/mail
```

Cần chú ý là các mức ưu tiên chỉ là thỏa thuận giữa các tiến trình với nhau. Việc gửi thông báo với mức độ ưu tiên nào hoàn toàn do người làm ra chương trình chủ động. Đây có thể coi là một cơ chế rất mềm dẻo để phối hợp giữa các tiến trình, nhưng đồng thời cũng là một lỗ hổng về bảo mật. Bằng việc xem xét log, quản trị viên có thể có một cái nhìn tổng quan về các sự kiện xảy ra trong hệ thống. Tuy nhiên, với những quản trị viên chưa quen với hệ thống Linux, hoặc khi có một phần mềm nào đó thay đổi về vị trí khi nhật ký, việc tìm được các thông tin cần thiết gặp nhiều khó khăn.

Kiểm tra việc cấu hình nhật ký Để kiểm tra cấu hình nhật ký theo dòng, có thể sử dụng chương trình logger. Chương trình logger là chương trình cho phép gửi các sự kiện đến syslogd. Sau đó có thể kiểm tra xem các thông tin nhật ký có được lưu đúng vào vị trí dự định hay không.

```
logger -p daemon.warn "Day chi la kiem tra thoi"
```

Để có thể kiểm tra tất cả các mức ưu tiên, có thể viết một kịch bản cho shell

```
#!/bin/bash
#
# Script to generate one log message per
# priority level per facility
#
for i in {auth,authpriv,cron,daemon,kern,lpr,mail,mark,news,syslog,user,uucp,loc
do
  for k in {debug,info,notice,warning,err,crit,
            alert,emerg}
  do
    logger -p $i.$k "Test message, facility $i
                     priority $k"
  done
done
```

Các tệp log mà quản trị viên thường xuyên phải quan tâm là:

- /var/log/message: Các thông báo chung của hệ thống.
- /var/log/auth.log: Nhật ký xác thực.
- /var/log/kern.log: Thông báo từ nhân Linux
- /var/log/cron.log: Thông báo từ việc thực hiện các thao tác tự động.
- /var/log/maillog: Nhật ký của Mail server
- /var/log/qmail/ : Nhật ký của Qmail.
- /var/log/httpd/: Nhật ký của Apache.
- /var/log/lighttpd: Nhật ký của Lighttpd.
- /var/log/boot.log : Các thông báo của quá trình khởi động.
- /var/log/mysqld.log: Nhật ký của MySQL
- /var/log/secure: Nhật ký của quá trình đăng nhập.

12.3 Tối ưu quá trình ghi nhật ký

Các tệp nhật ký sau một thời gian sử dụng sẽ có kích thước tăng lên không ngừng. Nếu không có các biện pháp, trên lý thuyết kích thước của các tệp nhật ký sẽ tăng vô hạn. Rất khó có thể xác định được các thông tin trong một số lượng lớn các tệp nhật ký.

logrotate là phần mềm được xây dựng để làm thuận tiện hơn việc quản trị các hệ thống sản sinh ra nhiều các tệp nhật ký. logrotate cho phép nén, xóa, cập nhật và gửi các tệp nhật ký đi một cách tự động. Mỗi tệp nhật ký sẽ được kiểm tra định kỳ theo tuần, theo tháng và theo kích thước của tệp nhật ký.

Với cấu hình mặc định, logrotate thực hiện các thao tác trên các tệp nhật ký hàng ngày. Các tệp nhật ký sẽ được cập nhật khi các điều kiện về kích thước của tệp được kích hoạt, hoặc khi quản trị viên khai báo các thao tác này một cách tường minh (tùy biến –force).

logrotate có thể hoạt động trên nhiều tệp cấu hình khác nhau. Các tệp này thông thường được liên kết với nhau trong một tệp cấu hình duy nhất. Hình 12.3.3 là ví dụ một tệp cấu hình đơn giản cảu logrotate Các thao tác trên từng tệp nhật ký riêng biệt được ghi trong các tệp cấu hình đặt trong thư mục /etc/logrotate.d/. Ví dụ Các tùy biến trong các tệp cấu hình có ý nghĩa như sau:

```
# see "man logrotate" for details
# rotate log files weekly
weekly
# keep 4 weeks worth of backlogs
rotate 4
# create new (empty) log files after rotating old ones
create
# uncomment this if you want your log files compressed
#compress
# RPM packages drop log rotation information into this directory
include /etc/logrotate.d
# no packages own wtmp -- we'll rotate them here
/var/log/wtmp {
monthly
create 0664 root utmp
rotate 1
}
```

Hình 12.3.3: Ví dụ về tệp cấu hình của logrotate

```
/var/log/httpd/*.log {
 weekly
 rotate 52
 compress
  missingok
  notifempty
  sharedscripts
  postrotate
      /bin/kill -HUP `cat /var/run/httpd.pid 2>/dev/null` 2> /dev/null || true
}
```

Hình 12.3.4: Cấu hình logrotate cho một dịch vụ cụ thể

- weekly : Các tệp nhật ký được thực hiện nếu ngày trong tuần hiện tại nhỏ hơn ngày trong tuần khi thực hiện kiểm tra tệp nhật ký hoặc đã kiểm tra được hơn 01 tuần.
- rotate 52 : Tệp nhật ký được xử lý 52 lần trước khi bị xóa đi hoặc gửi theo email.
- compress : Các tệp lưu trữ cũ của nhật ký được nén (để tiết kiệm không gian đĩa).
- missingok : Nếu tệp nhật ký không có, tiếp tục xử lý các tệp nhật ký tiếp theo. Không thông báo lỗi.
- notifempty : Không xử lý nếu nhật ký rỗng.
- sharedscripts : Các tệp nhật ký cùng thực hiện một kịch bản sẽ chỉ thực hiện kịch bản một lần. Nếu không có nhật ký nào được xử lý, kịch bản dùng chung này sẽ không thực hiện.
- postrotate câu lệnh endscript : Câu lệnh thực hiện sau khi xử lý xong tệp nhật ký.

12.4 Bài tập

Bài tập 12.1 *Đăng nhập vào hệ thống bằng tài khoản người quản trị, xem nội dung tệp /var/log/messages. Câu lệnh nào cho biết các sự kiện mới nhất xảy ra trong hệ thống.*

Bài tập 12.2 *Theo dõi tệp nói trên sử dụng lệnh tail*

Bài tập 12.3 *Mở một console khác, thử đăng nhập bằng mật khẩu đúng và mật khẩu sai, theo dõi sự thay đổi của tệp trên*

Bài tập 12.4 *Căn cứ vào tệp cấu hình của logrotate, giải thích tệp /var/log/messages được xử lý thế nào.*

Chương 13

Tự động hóa công việc

13.1 Khái niệm

Nhiệm vụ của quản trị viên một hệ thống máy tính là đảm bảo cho hệ thống luôn luôn hoạt động bình thường. Để thực hiện nhiệm vụ này, quản trị viên không những phải nắm được các kiến thức về hệ thống, luôn luôn sẵn sàng để có thể khắc phục sự cố, mà còn phải thường xuyên kiểm tra hệ thống máy tính để có thể phát hiện các bất thường, dự đoán các sự cố xảy ra để có các phương án xử lý phù hợp. Với sự giúp đỡ của các công cụ sao lưu, ghi nhật ký, các công việc này của quản trị viên trở nên dễ dàng hơn nhiều. Tuy nhiên, việc thường xuyên định kỳ phải thực hiện công tác sao lưu, dọn dẹp và kiểm tra nhật ký cũng như các thao tác kiểm tra khác trên hệ thống là công việc rất nặng nề về số lượng, ít tính sáng tạo, đòi hỏi tính kiên nhẫn và bền bỉ. Mặt khác, khi thực hiện các thao tác với số lượng lớn, việc có sai sót là không thể tránh khỏi, việc phát hiện ra sai sót là vô cùng khó khăn.

Để giải quyết vấn đề này, cần có cách thức để tự động hóa các công việc của quản trị viên. Có hai phương pháp thực hiện tự động hóa các thao tác. Phương pháp thứ nhất định nghĩa các thao tác sẽ được lặp đi lặp lại theo năm, tháng, tuần, ... Phương pháp thứ 2 định nghĩa các thao tác sẽ được thực hiện một lần trong tương lai. Trong hệ điều hành Linux, phương pháp thứ nhất được thực hiện bằng phần mềm crond, phương pháp thứ 2 được thực hiện bằng phần mềm at.

13.2 Câu lệnh crond

Cron là tiến trình thường trú để đặt lịch thực hiện cho các tác vụ. Mỗi người sử dụng sẽ có một tệp crontab lưu trữ lịch thực hiện các thao tác vào các thời điểm khác nhau. Hệ thống cũng có một tệp crontab hệ thống dùng để đặt lịch cho các thao tác hệ thống như dọn dẹp log, cập nhật CSDL của locate, ...

Để khai báo lịch thực hiện tác vụ, có thể sử dụng các câu lệnh, cũng có thể sử dụng chương trình với giao diện đồ họa như Gnome Scheduled tasks.

Tệp crontab là một tệp có dạng text, chứa danh sách các thao tác sẽ được thực hiện và thời gian thực hiện các thao tác đó. Các thao tác này và thời gian thực hiện sẽ được điều khiển bởi tiến trình crond và được thực hiện ở chế độ nền của hệ thống.

Để khai báo các thao tác tự động bằng câu lệnh, sử dụng câu lệnh crontab -e để thêm các dòng vào trong tệp crontab. Sau khi đã khai báo các tác vụ theo cú pháp của cron, cần lưu lại các thay đổi và thoát ra khỏi chương trình soạn thảo. Các thay đổi sẽ được cron cập nhật. Nếu không ghi lại, sẽ không có thay đổi nào với lịch thực hiện của các tác vụ được áp dụng.

13.2.1 Cú pháp của tệp crontab

Tệp crontab gồm nhiều dòng, mỗi dòng tương ứng với lịch thực hiện của một tác vụ. Mỗi dòng gồm nhiều trường, các trường phân cách nhau bằng dấu cách. Các trường từ 1 đến 5 dùng để khai báo thời gian theo thứ tự *phút, giờ, ngày, tháng, thứ*. Trường thứ 6 dùng để khai báo câu lệnh thực hiện tác vụ. Ví dụ

```
01 11 1 1 1 /usr/bin/somedirectory/somecommand
```

Khai báo tạc vụ /usr/bin/somedirectory/somecommand thực hiện vào 11:01am ngày 01/01 và mọi ngày thứ 2 của tháng 1. Dấu sao (*) sử dụng để mô tả giá trị tùy ý của trường. Ví dụ

```
01 11 * * * /usr/bin/somedirectory/somecommand
```

thực hiện câu lệnh /usr/bin/somedirectory/somecommand vào 11:01am tất cả các ngày. Dấu phẩy (,) có thể được sử dụng để khai báo nhiều thời điểm thực hiện cùng một tác vụ. Dấu trừ (-) khai báo một khoảng các giá trị để thực hiện liên tục câu lệnh. Ví dụ

```
01,31 04,05 1-15 1,6 * /usr/bin/somedirectory/somecommand
```

thực hiện câu lệnh /usr/bin/somedirectory/somecommand vào phút thứ 4h01, 4h31, 5h01, 5h31 từ ngày 1 đến nagfy 15 của các tháng 01 và tháng 6. Câu lệnh thực hiện tác vụ "/usr/bin/somedirectory/somecommand" cần được khai báo bằng đường dẫn tuyệt đối

Có thể khai báo các tác vụ thực hiện một cách định kỳ. Ví dụ

```
*/30 * * * * /usr/bin/somedirectory/somecommand
```

thực hiện câu lệnh mỗi 30 phút thay cho việc khai báo

```
0,30 * * * * /usr/bin/somedirectory/somecommand
```

Để xem danh sách, cập nhật danh sách các tác vụ đã được thực hiện, có thể sử dụng câu lệnh crontab. crontab -l liệt kê danh sách các thao tác sẽ được thực hiện. crontab -e cho phép soạn thảo các lịch thực hiện. crontab -r xóa các tác vụ. crond thực hiện các lịch được ghi trong /var/spool/crontabs, tuy nhiên để đảm bảo những NSD khác nhau không tác động lên lịch của nhau, không nên trực tiếp soạn thảo tệp này.

```
This has the username field, as used by /etc/crontab.
       # /etc/crontab: system-wide crontab
       # Unlike any other crontab you don't have to run the `crontab'
       # command to install the new version when you edit this file.
       # This file also has a username field, that none of the other crontabs do

       SHELL=/bin/sh
       PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

       # m h dom mon dow user  command
       42 6 * * *  root  run-parts --report /etc/cron.daily
       47 6 * * 7  root  run-parts --report /etc/cron.weekly
       52 6 1 * *  root  run-parts --report /etc/cron.monthly
       #
       # Removed invocation of anacron, as this is now handled by a
       # /etc/cron.d file
```

Hình 13.2.1: Thực hiện anacron bằng câu lệnh runparts

Giới hạn truy cập

Crond sử dụng các tệp /etc/crond.allow và /etc/crond.deny để xác định NSD nào được quyền khai báo các thao tác tự động. Chỉ những NSD nào có tên trong tệp /etc/crond.allow và không có tên trong tệp /etc/crond.deny mới có quyền tùy biến crond cho riêng mình. Nếu tệp /etc/cron.allow tồn tại, NSD cần phải được liệt kê trong tệp này để có thể khai báo crontab. Tương tự, nếu tệp /etc/cron.deny tồn tại, NSD cần không có mặt trong tệp này mới sử dụng được crond. Nếu cả 2 tệp này không tồn tại, kết quả phụ thuộc vào bản phân phối.

Các lịch thực hiện nói trên dành cho NSD. Để có thể thực hiện các tác vụ hệ thống, sử dụng câu lệnh

```
sudo crontab -e
```

để khai báo các tác vụ cần thực hiện với quyền của quản trị.

Sử dụng crond, anacron cho phép khai báo các tác vụ hệ thống được thực hiện trong các thư mục /etc/cron.hourly/, /etc/cron.daily/, /etc/cron.weekly/, và /etc/cron.monthly/ lần lượt cho các thao tác thực hiện hàng giờ, hàng ngày, hàng tuần. Tương tự như trong /etc/rcX.d/ trong thư mục này các script được thực hiện theo thứ tự tên của các kịch bản.

13.2.2 Một vài chú ý

Các câu lệnh khởi động bằng crond không có console, do đó bắt buộc phải khai báo tường minh đầu ra chuẩn, nếu không sẽ không được thực hiện. Có thể khởi động các ứng dụng đồ họa, tuy nhiên, cần đưa màn hình đồ họa chuẩn vào trong ACL (xem thêm chương **??**)

Trong các thao tác liên quan đến sao lưu, các tệp sao lưu sử dụng thời gian hiện tại trong tên tệp. Để làm được điều này cần sử dụng câu lệnh date và cú pháp ; : Ví dụ

```
*/30 * * * * echo test > test‘date +"%Y%m%d_%H%M%S"‘
```

tạo ra tên tệp với năm, tháng, ngày, giờ, phút, giây trong tên tệp.

13.3 Câu lệnh at

Câu lệnh at kết hợp với các câu lệnh batch, atq, atrm thực hiện các lệnh được nhập vào từ stdin hoặc tệp và thực hiện tại một thời điểm thời gian sau đó. Câu lệnh atq cho phép xem danh sách các lệnh sẽ được thực hiện, còn câu lệnh atrm cho phép xóa các lệnh đã được đặt để thực hiện. Việc thực hiện các lệnh này do tiến trình atd thực hiện. Như vậy để sử dụng at cần chắc chắn là dịch vụ atd đã được khởi tạo. Ví dụ câu lệnh:

```
at -f echo "It is now $(date +%T) on $(date +%A)" -v 14:00
```

Thực hiện câu lệnh vào lúc 14:00.

Tham số thời gian có thể được định nghĩa theo nhiều cách khác nhau

```
at -f test  10pm tomorrow
at -f test 2:00 tuesday
at -f test 2:00 july 11
at -f test 2:00 next week
```

Để đảm bảo an toàn cho hệ thống, các tệp /etc/at.allow và /etc/at.deny được sử dụng. Cơ chế khai báo NSD và hiệu ứng tương tự như tệp cron.allow và cron.deny.

13.4 Bài tập

Bài tập 13.1 *Sử dụng câu lệnh crontab để thực hiện việc tìm các tệp có suid và guid bít.*

Bài tập 13.2 *Sử dụng câu lệnh at để tắt máy vào 10h tối ngày mai.*

Phần III

Quản trị các dịch vụ mạng trên Linux

Chương 14

Dịch vụ tên miền-DNS

14.1 Khái niệm cơ bản về DNS

14.1.1 Vai trò của DNS

Mỗi máy tính khi tham gia vào mạng được cấp một địa chỉ IP, để kết nối với nhau các máy tính cần biết được địa chỉ IP của nhau. Cũng như vậy người sử dụng khi sử dụng các dịch vụ mạng như web, email, chat... phải nhớ được địa chỉ IP của các máy chủ cung cấp dịch vụ này để ra lệnh cho máy tính của mình kết nối đến. Việc này gây khó khăn cho người sử dụng do phải nhớ rất nhiều con số khô khan và không có tính gợi nhớ. Ngoài ra mỗi máy tính hay máy chủ khi cài đặt còn có một tên máy (hostname). Tên này thường được đặt bằng ngôn ngữ tự nhiên và mang tính gợi nhớ cao. Ví dụ máy tính cá nhân của chúng ta thường đặt luôn theo tên của chúng ta như thanhlx, trunghq, máy chủ cung cấp các dịch vụ thì được đặt tên với yếu tố gợi nhớ đến dịch vụ đó như: mail.hut.edu.vn, hut.edu.vn... Dịch vụ DNS ra đời nhằm kết nối ưu điểm của hai dịch vụ này thành một dịch vụ trung gian duy nhất. DNS làm nhiệm vụ ánh xạ các tên dễ nhớ của máy tính thành địa chỉ IP tương ứng của máy đó. DNS chạy như một dịch vụ trung gian cho tất cả các dịch vụ mạng khác theo mô hình Client/Server. Khi một người dùng muốn kết nối máy tính của mình tới máy khác chỉ cần gõ vào tên máy tính cần kết nối, máy của người sử dụng sẽ tự động liên lạc với máy chủ DNS để hỏi địa chỉ IP tương ứng và thiết lập kênh liên lạc để hai máy trao đổi thông tin.

Phân giải tên miền Dịch vụ DNS hoạt động theo mô hình Client/Server với Server là một máy chủ làm nhiệm vụ quản lý cơ sở dữ liệu tên miền và đưa ra câu trả lời khi nhận được yêu cầu, thường được gọi là Name Server. Client là một trình truy vấn - Resolver - được các máy tính dùng để truy vấn thông tin cần thiết trên Name Server.

Đảm bảo tính sẵn sàng DNS là một dịch vụ phân tán, điều này cho phép dữ liệu được "chia nhỏ" để quản lý bởi các máy chủ DNS cục bộ, các máy chủ này quản lý việc phân giải tên miền nội bộ thuộc phạm vi được cho phép hoặc quản lý một bộ phận tên miền cụ thể nào đó. Điều này khiến cho việc quản trị trở nên dễ dàng hơn đồng thời việc truy vấn dữ liệu được nhanh hơn.

Phân tải

Đảm bảo tính trong suốt Tất cả các hệ điều hành cho máy tính ngày nay đều tích hợp sẵn các DNS Client và việc truy vấn thông tin diễn ra hoàn

toàn trong suốt đối với người sử dụng. Người sử dụng chỉ cần nhớ hoặc tìm ra tên miền mình cần truy cập và gõ vào chương trình, mọi công việc còn lại do DNS Client thực hiện. Thực tế cũng cho thấy phần lớn người sử dụng không hề biết đến dịch vụ này, nó hoàn toàn trong suốt.

Các giải pháp diễn giải tên

14.1.2 Hệ thống tên miền

Không gian tên Một dữ liệu DNS là một cấu trúc hình cây, một tên miền là một nhánh của cây, tuy nhiên khác với các cấu trúc cây khác như cây thư mục hay cây domain, cấu trúc cây của tên miền được viết từ ngọn đến gốc. Mỗi tên miền được kết thúc bằng dấu "." là ký hiệu là gốc của cây (root) - ví dụ <www.hut.edu.vn.>. Tuy nhiên để thuận tiện, người sử dụng không cần thiết phải gõ dấu chấm này mà nó được máy tính tự động thêm vào. Một tên miền đầy đủ bao gồm có các thành phần cách nhau bởi dấu ".": <tên miền . tên miền cấp 1 (hoặc tên miền cấp 2 . tên miền cấp .) . root>. Tên miền đầy đủ được gọi là FQDN (Fully Qualified Domain Name)

Tên miền cấp 1 Còn được gọi là Top-Level Domain (TLD) được phân loại theo:

- Quốc gia (ccTLD - Country Code Top-Level Domain): .vn (Việt Nam), .jp (Japan), .tw (taiwan)...
- Chức năng (gTLD - Generic Top-Level Domain): .com (các tổ chức và công ty thương mại), edu (Các tổ chức giáo dục), .org (các tổ chức phi lợi nhuận)...
- Được các tổ chức tài trợ: .asia (Các tổ chức thuộc asia do DotAsia Organisation tài trợ), .mil (Quân đội Mỹ - do DoD Network Information Center tài trợ)...

hoặc hay được phân loại theo chức năng như: .com (các tổ chức và công ty thương mại)

Tên miền cấp 2 là tên miền kết hợp giữa tên miền cấp 1 theo chứ ăng và tên miền cấp 1 theo quốc gia. Ví dụ: .edu.vn, .com.eu, .org.jp

FQDN - Fully Qualified Domain Name là tên miền đầy đủ các thành phần như đã nói ở trên. Ví dụ: www.hut.edu.vn, www.vnn.vn, www.bbc.com...

14.1.3 Các thành phần của hệ thống tên miền

Client là trình phân giải tên miền Resolver, được tích hợp trên tất cả các hệ điều hành máy tính hiện nay. Resolver chứa các hàm dùng để tạo ra các truy vấn và gửi chúng đến Name Server sau đó chuyển kết quả thu được cho các ứng dụng cần thiết.

Máy chủ toàn cầu Là các Name Server quản lý địa chỉ của các Top-Level Domain. Các máy chủ này đặt rải rác khắp thế giới, địa chỉ của chúng được công khai cho mọi người biết. Hiện tại có 13 máy chủ toàn cầu được đánh nhãn từ A - M và có tên miền là: a.root-servers.net, b.rootservers.net....

Máy chủ quản lý TLD Được tổ chức quản lý tên miền ủy quyền cho các tổ chức quản lý TDL như: ủy ban quản lý tên miền của các nước (ccTLD), Các nhà đăng ký (gTLD)và tài trợ (sTLD). Các tổ chức này có nhiệm vụ quản lý và phân phối tên miền trong phạm vi quản lý của mình cho các tổ chức, cá nhân có nhu cầu.

Máy chủ vận hành các DNS vùng và các tên miền đầy đủ . Các DNS vùng là các máy chủ tên miền được đăng ký bởi các nhà cung cấp dịch vụ, các tổ chức này dựng các máy chủ DNS nhằm cung cấp dịch vụ tốt hơn cho khách hàng đồng thời giảm tải cho các DNS toàn cầu hay DNS, ví dụ DNS của Google (4.4.4.4, 8.8.8.8), DNS của VDC (203.162.0.11), Open DNS (208.67.220.220)... Máy chủ vận hành các FQDN là máy chủ DNS của các cơ quan tổ chức dùng để vận hành tên miền được các cơ quan, tổ chức đó đăng ký, các máy chủ DNS này được đăng ký trên Internet và được dùng để duy trì cơ sở dữ liệu DNS cho các tên miền mà cơ quan, tổ chức đó đăng ký sử dụng

14.1.4 Cơ chế phân giải tên miền

Các máy chủ DNS thường sử dụng hai cơ chế để phân giải tên miền khi có yêu cầu từ phía client đó là:

Phân giải đệ qui Khi Client gửi yêu cầu đệ quy đến Server, máy chủ DNS thỏa thuận với Client có hỗ trợ hay không.

- Nếu không hỗ trợ tìm kiếm đệ quy, máy chủ DNS tìm kiếm thông tin trong cơ sở dữ liệu cục bộ, nếu có thì gửi lại thông tin, nếu thông thì gửi thông báo là không có hoặc báo host đó đang bận.

- Nếu có hỗ trợ tìm kiếm đệ quy, máy chủ DNS tìm kiếm trong cơ sở dữ liệu cục bộ, nếu có kết quả sẽ trả lời client
- Nếu không tìm thấy thông tin client cần trong cơ sở dữ liệu, server sẽ hỏi các máy chủ toàn cầu
- Các máy chủ toàn cầu phân tích và trả lời lại máy chủ DNS địa chỉ của máy chủ TLD tương ứng
- Máy chủ DNS lại tiến hành hỏi máy chủ TLD và cứ tiếp tục hạ cấp như thế đến khi tìm được câu trả lời
- Máy chủ DNS trả kết quả tìm thấy cho Client.

Phân giải không đệ qui Khác với phân giả đệ quy, phân giải không đệ quy diễn ra như sau: Client gửi yêu cầu phân giải không đệ quy (phân giải tương tác) đến máy chủ DNS, máy chủ DNS sẽ thỏa thuận với Client xem có hỗ trợ kiểu phân giải này không.

- Nếu không hỗ trợ tìm kiếm tương tác, máy chủ DNS tìm kiếm thông tin trong cơ sở dữ liệu cục bộ, nếu có thì gửi lại thông tin, nếu thông thì gửi thông báo là không có hoặc báo host đó đang bận.
- Nếu có hỗ trợ tìm kiếm tương tác, máy chủ DNS tìm kiếm trong cơ sở dữ liệu cục bộ, nếu có kết quả sẽ trả lời client
- Nếu không có máy chủ DNS sẽ gửi cho client địa chỉ của máy chủ toàn cầu.
- Client sẽ tự hỏi các máy chủ toàn cầu này về tên miền và cứ thế thực hiện cho đến khi tìm được câu trả lời.

Kết hợp phân giải đệ qui và không đệ qui

14.1.5 Các loại server

Mỗi một tổ chức, cơ quan khi đăng ký một hay nhiều tên miền có thể dựng một hay nhiều máy chủ để quản lý và duy trì cơ sở dữ liệu tên miền cho các hoạt động liên quan đến tên miền đó. Ngoài ra các cơ quan tổ chức này có thể dựng thêm các máy chủ DNS để chạy song song backup cho các DNS kia. Khi đó các máy chủ DNS chính được gọi là Primary Name Server còn các máy chủ DNS làm nhiệm vụ backup được gọi là các Secondary Name Server hay các Slave Server. Khi các máy chủ Primary Name Server bị hỏng

các máy Secondary Name Server sẽ lên thay thế làm nhiệm vụ phân giải tên miền. Hai máy chủ này có một cơ chế để thường xuyên đồng bộ dữ liệu cho nhau được gọi là Zone Tranfer. Ngoài ra các Primary Name Server có thể tạo ra các SubDomain và ủy quyền cho các Name Server khác quản lý các SubDomain này.

14.2 Cài đặt và cấu hình DNS trên Linux

Các thao tác cần thực hiện để có được một máy chủ DNS là:

- Cài đặt phần mềm Bind (Berkeley Internet Name Daemon).
- Cấu hình Bind để tiến trình named có thể hoạt động.
- Cấu hình các dữ liệu của zone mà máy chủ quản lý.
- Nếu là máy chủ chính, cấu hình các zone: localhost và zone phân giải ngược, zone được quản lý và zone phân giải ngược, các máy chủ gốc.
- Nếu là máy chủ phụ cấu hình các zone: localhost và zone phân giải ngược..

14.2.1 Cài đặt phần mềm

Trong phạm vi cuốn sách này tác giải minh họa việc cài đặt DNS bằng phần mềm BIND9 trên nền hệ điều hành Debian. Để cài đặt một phần mềm trên linux bạn đọc có hai cách:

- Cài từ gói phần mềm được build sẵn cho hệ điều hành đang sử dụng (giống windows)
- Tải mã nguồn của phần mềm rồi build lại và cài đặt.

Ở phương án cài đặt thứ 2 yêu cầu bạn đọc phải nắm rất chắc kiến thức về hệ thống để có thể build được mã nguồn phù hợp với cấu hình hệ thống và ý đồ triển khai của mình. Ở đây tác giả giới thiệu với bạn đọc phương án cài từ gói đóng sẵn cho phân mềm. Hệ thống được thiết kế như sau. máy chủ DNS được cấu hình chi thành hai vùng trong và ngoài. Đối với các máy tính nằm trong mạng LAN khi truy vấn thông tin địa chỉ DNS sẽ trả lời địa chỉ nội bộ của các máy chủ. Nếu là các truy vấn từ các máy tính bên ngoài mạng DNS sẽ trả lời địa chỉ public của máy chủ dịch vụ. Tiến hành cài đặt BIND 9:

```
debian:~# apt-get install bind9
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  bind9utils
Suggested packages:
  bind9-doc resolvconf ufw
The following NEW packages will be installed:
  bind9 bind9utils
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 336kB of archives.
After this operation, 1028kB of additional disk space will be used.
.....
```

Đọc đoạn cấu hình trên ta có thể nhận thấy rằng để cài đặt BIND9 cần phải cài 2 gói sau: bind9, bind9utils Sau khi cài đặt xong ta thực hiện lệnh tắt dịch vụ để chuyển sang cấu hình chi tiết BIND9:

```
debian:~# /etc/init.d/bind9 stop
```

14.2.2 Cấu hình daemon

```
controls {
inet 127.0.0.1 allow { localhost; } keys { rndckey; rndc-key; };
};

//include "/etc/named.custom";
include "/etc/rndc.key";

// ACL statement
acl proxy  { 172.16.0.201/32;};
acl localnet { 127.0.0.1; 172.16.0.0/16;192.168.0.0/16; };
acl bogusnets { 0.0.0.0/8; 1.0.0.0/8; 2.0.0.0/8; 224.0.0.0/3; 10.0.0.0/8; };

options {
directory "/var/named";
//allow-transfer { 172.16.0.210;172.16.0.212; };
blackhole { bogusnets; };
recursive-clients 10000;
version "";
forwarders {203.162.0.11;208.67.220.220;};
```

```
};
// Begin Internal

view "internal" {
     match-clients { localnet; };
//allow-transfer { 172.16.0.210; };
recursion yes;

zone "." IN {
type hint;
file "named.ca";
};

zone  "hut.edu.vn" {
type master;
//allow-query {172.16.0.0/24;};
file  "in/hut.edu.vn.zone";
};

zone  "0.0.127.in-addr.arpa" {
type master;
file  "in/0.0.127.in-addr.arpa.zone";
};

zone  "0.168.192.in-addr.arpa" {
type master;
file  "in/4.168.192.in-addr.arpa.zone";
};
zone  "localhost" {
type master;
file  "in/localhost.zone";
};
};  //End Internal

view "global" {
match-clients { any; };
recursion yes;
allow-recursion { 202.47.142.0/24;202.191.56.0/22; };
//allow-transfer { 202.47.142.131/32; };
zone "hut.edu.vn" IN {
type master;
```

```
file "out/hut.edu.vn.zone";

};
};

key rndc-key {
algorithm hmac-md5;
secret "bakhKedjzAaCMUg8PaVfuA==";
};
```

14.2.3 Cấu hình zonefile

Cấu hình zone IN

```
$TTL 86400
@ IN SOA dns.hut.edu.vn.  dns.hut.edu.vn. (
2010042802; Tang khi thay doi thong tin
3H ; refresh
3600 ; retry
604800 ; expire
600 ; ttl 86400
)


IN NS dns.hut.edu.vn.
dns IN A 172.16.0.193
dns2 IN A 202.191.58.2
IN MX 30 mail.hut.edu.vn.
mail IN MX 20 mail.hut.edu.vn.
mail IN A 202.191.57.199
sv IN MX 40 sv.hut.edu.vn.

tvtt IN A 192.168.40.21 ; Dich vu Tu van Truc tuyen cho sinh vien
tuvantructuyen IN CNAME tvtt ; Dịch vu Tu van Truc tuyen cho sinh vien
#Lưu ý cuối file zone phải để một dòng trắng
```

Cấu hình file OUT

```
$TTL 86400
@ IN SOA dns.hut.edu.vn.  hostmaster.hut.edu.vn. (
2010042802 ;Tang khi thay doi thong tin
```

```
3H ; refresh
3600 ; retry
604800 ; expire
600 ; ttl 86400
)

IN NS dns.hut.edu.vn.
IN MX 20 mail.hut.edu.vn.
fit IN NS dns.hut.edu.vn.
mail IN MX 20 mail.hut.edu.vn.

tvtt IN A 202.191.56.196 ; website Tu van truc tuyen 20090520
tuvantructuyen IN CNAME tvtt ; Tu van Truc tuyen cho Sinh vien
#Lưu ý cuối file zone phải để một dòng trắng
```

Sau khi cấu hình xong các file trên bật lại dịch vụ DNS bằng lệnh

14.2.4 Kiểm tra cấu hình

Trước khi chạy chính thức hoặc sau mỗi lần thay đổi thông tin về DNS cần kiểm tra cấu hình dịch vụ bằng lệnh check zone như sau:

```
[root@dns etc]# named-checkzone hut.edu.vn /var/named/in/hut.edu.vn.zone
zone hut.edu.vn/IN: loaded serial 2010042802
OK
debian:~# named-checkzone hut.edu.vn /var/named/out/hut.edu.vn.zone
zone hut.edu.vn/OUT: loaded serial 2010042802
OK
```

Sau khi kiểm tra, khởi chạy lại dịch vụ để thay đổi có hiệu lực

```
debian:~# /etc/init.d/bind9 start (hoặc reload)
```

14.3 Bài tập

Bài tập 14.1 *Cài đặt hệ thống DNS với mô hình Master/Slave.*

Chương 15

Quản trị webserver

15.1 Dịch vụ web, PHP, MySQL, LAMP

15.1.1 HTML-Web page

Một trong những chức năng của mạng máy tính là cung cấp thông tin từ xa cho NSD. Thông tin thường được tổ chức thành các tài liệu. Các tài liệu này được lưu trữ trong các tệp, ví dụ như các tệp văn bản, tệp hình ảnh.... Trong quá trình sử dụng tài liệu, NSD thường xuyên phải tham chiếu tới các tài liệu khác phụ thuộc vào bối cảnh thông tin mà NSD đang truy cập trong tài liệu. Với mạng máy tính, có thể giả định là các tệp này có thể truy cập từ bất cứ vị trí nào. Để thuận tiện hơn cho NSD, cần có một cách thức để đánh dấu những tham chiếu tới tài liệu khác, cho phép khi kích hoạt các dấu đó, NSD sẽ truy cập vào tài liệu được tham chiếu. Tài liệu mà NSD đang truy cập sẽ trở thành một siêu tài liệu, có liên kết đến rất nhiều tài liệu khác nhau. Ngôn ngữ để mô tả một tài liệu như vậy gọi là ngôn ngữ đánh dấu siêu văn bản (Hyper Text Markup Language-HTML). Tài liệu được mô tả bằng ngôn ngữ HTML gọi là trang HTML hoặc web page.Tham chiếu tới một tài liệu được gọi là địa chỉ, hay còn được gọi là địa chỉ tài nguyên tổng quát (Uniform Resouce Locator-URL). Một URL có dạng protocol://domainname/path/file trong đó protocol là giao thức dùng để truy cập tài nguyên, với các tài liệu HTML là http, domainname là tên máy chứa tài liệu, path là đường dẫn tới tài liệu và file là tên của tài liệu.

15.1.2 Trình duyệt-web server

Để hiển thị các tài liệu HTML cần một phần mềm hiểu được ngôn nữ HTML và hiển thị thành dạng mà NSD có thể đọc được. Phần mềm này gọi là trình duyệt (browser). Các tài liệu HTML cần được lưu trữ tại một máy tính trên mạng (thường không trùng với máy tính của trình duyệt). Cần có một phần mềm để có thể cung cấp cho các trình duyệt trên máy tính khác các tài liệu HTML khi cần thiết. Phần mềm này gọi là máy chủ web. Trình duyệt và máy chủ trao đổi thông tin với nhau thông qua cơ chế yêu cầu-trả lời được qui định bởi giao thức truyền siêu văn bản (Hyper Text Transfer Protocol).Trình duyệt được sử dụng rộng rãi đầu tiên là Hot Potatoes. Hiện tại, các trình duyệt phổ biến nhất là Internet Explorer và Mozilla Firefox. Máy chủ web được sử dụng rộng rãi nhất là Apache Webserver và Microsoft Internet Information Server.

15.1.3 Cơ chế hoạt động của dịch vụ web

Hình ?? mô tả hoạt động của dịch vụ web. Các bước để tải một siêu văn bản là:

1. NSD nhập một địa chỉ tài liệu HTML hoặc kích hoạt vào một liên kết tới tài liệu.

2. Trình duyệt xác định máy chủ web, gửi một yêu cầu tới máy chủ chứa URL

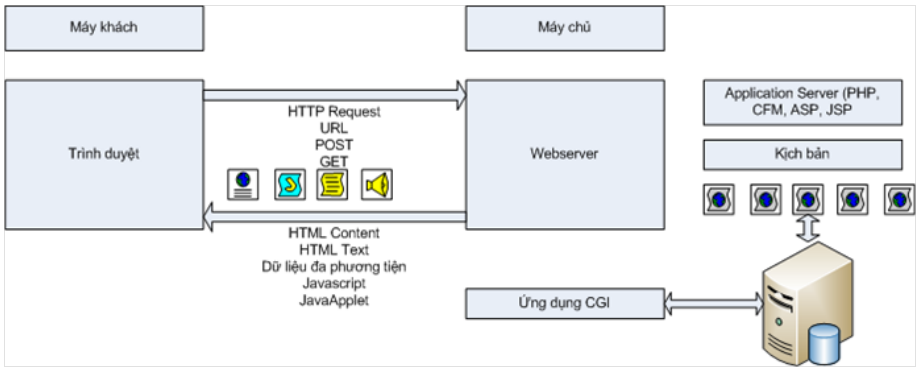
3. Máy chủ xác định vị trí vật lý của tài liệu trên máy chủ dựa vào đường dẫn và tên tài liệu. Nếu tìm thấy trả lại cho trình duyệt. Nếu không trả lại một tài liệu báo lỗi. Thông tin phản hồi được đóng gói trong một gói tin HTTP.

4. Trình duyệt nhận tài liệu HTML và hiển thị tài liệu. Trong quá trình hiển thị tài liệu, nếu cần thêm các tài nguyên khác, lặp lại bước 2.

15.1.4 Server Script và Client Script

Cơ chế trao đổi thông tin như đã trình bày ở trên chỉ cho phép máy chủ cung cấp thông tin đến cho NSD, không cho phép nhận thông tin từ NSD. Ngôn ngữ HTML cung cấp các thẻ để có thể khai báo các ô cho NSD có thể nhập thông tin vào và gửi đến máy chủ bằng cách đóng gói trong yêu cầu. Tại máy chủ các thông tin này được xử lý để đưa ra các thông tin theo yêu cầu của NSD. Đây là cơ chế để có thể cung cấp các nội dung thông tin động bằng các trang web. Điểm còn thiếu là một cơ chế cho phép các chương trình thực hiện trên máy chủ có thể giao tiếp với web server để nhận các thông tin từ phía NSD, xử lý và tùy biến thông tin theo các thông tin nhận được này rồi trả lại cho web server để gửi cho trình duyệt. Chuẩn đầu tiên để giao tiếp giữa các chương trình với máy chủ là chuẩn CGI (Common Gateway Interface). Mỗi yêu cầu của NSD được xử lý bởi một chương trình CGI. Để thuận tiện hơn cho NSD, việc viết các chương trình CGI cho một ứng dụng CGI được gộp lại và thay thế bằng các mô tả tương tác và thao tác trên máy chủ. Các mô tả này được viết bằng ngôn ngữ chuyên dụng và có một trình dịch trên server thực hiện các kịch bản của ngôn ngữ. Trình dịch này gọi là máy chủ ứng dụng (application server), còn đoạn mã mô tả gọi là kịch bản trên máy chủ (server side script).

Các ứng dụng web có thể sử dụng server-side script để thực hiện tương tác giữa NSD và máy chủ web. Tuy nhiên, mức độ tương tác thấp do tốc độ truyền dữ liệu hạn chế. Để có mức độ tương tác lớn hơn, các trình duyệt cần



Hình 15.1.1: Các thành phần của hệ thống web

cho phép thực hiện các chương trình ngay tại trình duyệt. Giải pháp xuất hiện đầu tiên là sử dụng máy ảo Java để thực hiện các Applet. Các applet cho phép NSD tương tác với trình duyệt và máy chủ bằng giao diện như với các chương trình Desktop thông thường. Nhược điểm của chúng là chậm và kích thước lớn. Giải pháp thứ 2 là sử dụng các kịch bản chạy trên trình duyệt như JavaScript hay VB Script. Hầu hết các trình duyệt hiện tại đều hỗ trợ Java Script. Web 2.0 còn cho phép các kịch bản này có thể tương tác trực tiếp với máy chủ, thực hiện song song quá trình tải và hiển thị các trang web, tăng mức độ tương tác giữa NSD và hệ thống. Hình 15.1.1 minh họa các thành phần của hệ thống web.

15.2 Apache web server trên Linux

15.2.1 Cài đặt

Phiên bản phần mềm Apache mới nhất đang được sử dụng là phiên bản 2.2. Trước khi cài đặt, cần đảm bảo là máy tính đã được cấu hình để kết nối mạng. Có thể cài đặt apache từ gói apache2

```
#yum install apache2
hoặc
#apt-get install apache2
```

Khi cài đặt từ gói apache2, chương trình cài đặt sẽ thực hiện các thao tác cấu hình mặc định. Sau khi quá trình cài đặt được hoàn tất, Apache đã được cấu hình mặc định và có thể sử dụng. Từ phiên bản 2.0, máy chủ web được kiểm soát bằng câu lệnh:

```
/etc/init.d/apache2 start|stop|restart|reload|status
```

Trong đó các tùy biến start, stop, restart, reload, status được sử dụng để khởi động, tắt, khởi động lại, hiển thị trạng thái của máy chủ web. Sau khi dịch vụ web được khởi động, có thể kiểm tra bằng câu lệnh

```
#telnet 127.0.0.1 80
hoặc
#lynx 127.0.0.1
```

Trên máy tính khác, cần thay 127.0.0.1 bằng địa chỉ IP tương ứng của máy tính.

15.2.2 Tùy biến

Các tệp cấu hình của Apache web server được đặt trong thư mục /etc/apache2/. Tệp cấu hình chính là tệp /etc/apache2/apache2.conf định nghĩa các cấu hình cơ bản của httpd, tiến trình phục vụ cho dịch vụ web. Các thư mục khác và các tệp cấu hình khác khai báo các cấu hình khác nhau của các web site, web pages, ... do tiến trình httpd quản lý. Cấu hình của apache được tạo bởi các lệnh khai báo (directive), các lệnh khai báo này có thể được tập hợp lại trong một môi trường. Như vậy các cấu hình được khai báo trong tệp /etc/apache2/apache2.conf chỉ cần thay đổi khi quản trị viên muốn thay đổi các tham số của tiến trình httpd như: hiệu năng, số các luồng con, số các tiến trình con.... Các cấu hình khác như ngôn ngữ, chế độ bảo mật đặt trong thư mục /etc/apache2/conf.d Việc khai báo thêm các mô đun bổ trợ của Apache2 được khai báo trong thư mục /etc/apache2/mod-enabled. Các mô đun được cài đặt nhưng chưa được kích hoạt đặt trong thư mục /etc/apache2/mod-available. Một trong các cấu hình quan trọng là

```
ServerRoot "/etc/apache2"
```

chỉ ra thư mục làm việc của tiến trình httpd. Các tệp cấu hình, các dữ liệu của tiến trình sẽ nằm trong thư mục này. Cấu hình Các thao tác cơ bản khác liên quan đến cấu hình web server là: Cấu hình một web site ảo, chia sẻ một tệp, một thư mục qua web, cấu hình kiểm soát truy cập cho tệp và thư mục.

15.2.3 Khai báo thư mục

Để web server có thể sử dụng một thư mục trong hệ thống tệp cục bộ, cần khai báo thư mục đó trong các tệp cấu hình. Sau khi khai báo, các thư mục con của thư mục thừa kế các cấu hình của thư mục cha. Một thư mục thường được khai báo như trong Hình 15.2.2 Tùy biến Options cho phép khai báo các thuộc tính của thư mục. Các thuộc tính cơ bản là ExecCGI cho phép

```
<Directory /var/www/>
Options Indexes FollowSymLinks MultiViews
AllowOverride None
Order allow,deny
allow from all
</Directory>
```

Hình 15.2.2: Khai báo thư mục

thực hiện các chương trình CGI, FollowSymLinks cho phép sử dụng các liên kết biểu tượng, Includes cho phép thực hiện liên kết tới các tệp khác trên server, IncludesNOEXEC chỉ cho phép liên kết tới các tệp không thực hiện, Indexes cho phép hiển thị nội dung của thư mục. Nếu có nhiều lệnh Options cho một thư mục, lệnh nào giới hạn nhiều nhấn sẽ được sử dụng. Nếu tất cả các Options đều sử dụng + hoặc - thì các Options sẽ kết hợp với nhau. AllowOverride None khai báo các thuộc tính của thư mục không thừa kế thư mục cha. Để đảm bảo các thư mục con tiếp theo có thể khai báo tường minh các giới hạn về truy cập, tại thư mục gốc khai báo ưu tiên các lệnh hạn chế truy cập, đồng thời mở ra tất cả các truy cập. Như vậy việc giới hạn quyền truy cập về sau trên thư mục gốc không ảnh hưởng đến các thư mục con. Các thư mục con có thể khai báo các giới hạn về truy cập theo nhu cầu sử dụng.

Thư mục gốc của một website được khai báo bằng câu lệnh Server Root. Mặc định địa chỉ IP của máy và tên miền của máy là các URL trỏ về Server-Root.

Các thư mục của thư mục gốc mặc định được hiểu là thư mục trong đường dẫn. Nếu muốn tạo ra các thư mục con khác, không nằm ở vị trí mặc định cần sử dụng câu lệnh Alias.

```
Alias /doc/ "/usr/share/doc/"
<Directory "/usr/share/doc/">
    Options Indexes MultiViews FollowSymLinks
    AllowOverride None
    Order deny,allow
    Deny from all
    Allow from 127.0.0.0/255.0.0.0 ::1/128
</Directory>
```

Khi cần chia sẻ một thư mục để có thể thực hiện các kịch bản, sử dụng câu lệnh ScriptAlias. Câu lệnh này giới hạn việc truy cập vào các kịch bản, không

cho phép đọc các kịch bản mà chỉ cho phép trả về các kết quả khi thực hiện kịch bản.

```
ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
<Directory "/usr/lib/cgi-bin">
AllowOverride None
Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
Order allow,deny
Allow from all
</Directory>
```

15.2.4 Cấu hình web site ảo

[?] Apache2 cho phép có thể cấu hình để với cùng một máy tính có thể lưu trữ nhiều website với tên miền khác nhau. Server sẽ căn cứ vào tên miền có trong yêu cầu (URL) để xác định vị trí vật lý của web site. Các khai báo website ảo có thể dựa trên tên miền hoặc dựa trên IP. Khi cấu hình một web site ảo, quản trị viên cần quan tâm đến các thông tin sau:

- Tên miền hoặc địa chỉ IP của website.
- Quản trị viên của web site.
- Thư mục gốc của web site. Thư mục này cần phải là một thư mục hoặc là thư mục con của thư mục đã được khai báo.

Ví dụ về khai báo web site ảo:

```
<VirtualHost 192.168.40.21:80>
        DocumentRoot /var/www/cfmi
        ServerName cfmi.hut.edu.vn
ServerAlias www.cfmi.hut.edu.vn
</VirtualHost>
```

15.2.5 PHP và MYSQL

PHP và MYSQL là 2 công cụ được sử dụng phổ biến để phát triển các web site động. PHP là ngôn ngữ kịch bản phía server, cho phép NSD có thể xây dựng các ứng dụng web. MySQL là hệ quản trị CSDL mã nguồn mở. Đây là những thành phần không thể thiếu của một ứng dụng web. Tổ hợp LAMP (Linux Apache MySQL PHP) thường được nhắc đến để chỉ các phần mềm phục vụ cho việc cài đặt một website. Để sử dụng các phần mềm này, trước hết cần phải cài đặt chúng

```
#yum install php5
#yum install mysql-server
#yum install  php5-mysql
#echo ‘‘<?php phpinfo(); ?>’’ >/var/www/test.php
#lynx 127.0.0.1/test.php
```

sau đó có thể cài một số phần mềm mã nguồn mở dựa trên PHP như Moodle, Joomla, ...

15.3 Bài tập

Bài tập 15.1 *Cài đặt web server trên máy tính với 3 website ảo khác nhau trên các thư mục khác nhau. Kiểm tra sự khác biệt.*

Bài tập 15.2 *Cài đặt Joomla và Moodle.*

Chương 16

Dịch vụ email

16.1 Giới thiệu hệ thống email

Hệ thống thư điện tử cung cấp một phương thức liên lạc hiệu quả bằng cách cung cấp các dịch vụ như:

- Gửi và nhận các thông báo thông qua webmail hoặc các chương trình email clienr
- Có thể đính kèm các file tài liệu hay các dữ liệu đa phương tiện và thư điện tử
- Có thể tích hợp vào các cơ chế xác thực bằng chữ ký số hoặc chữ ký điện tử để đảm bảo tính xác thực....

16.2 Các thành phần cơ bản của hệ thống thư điện tử

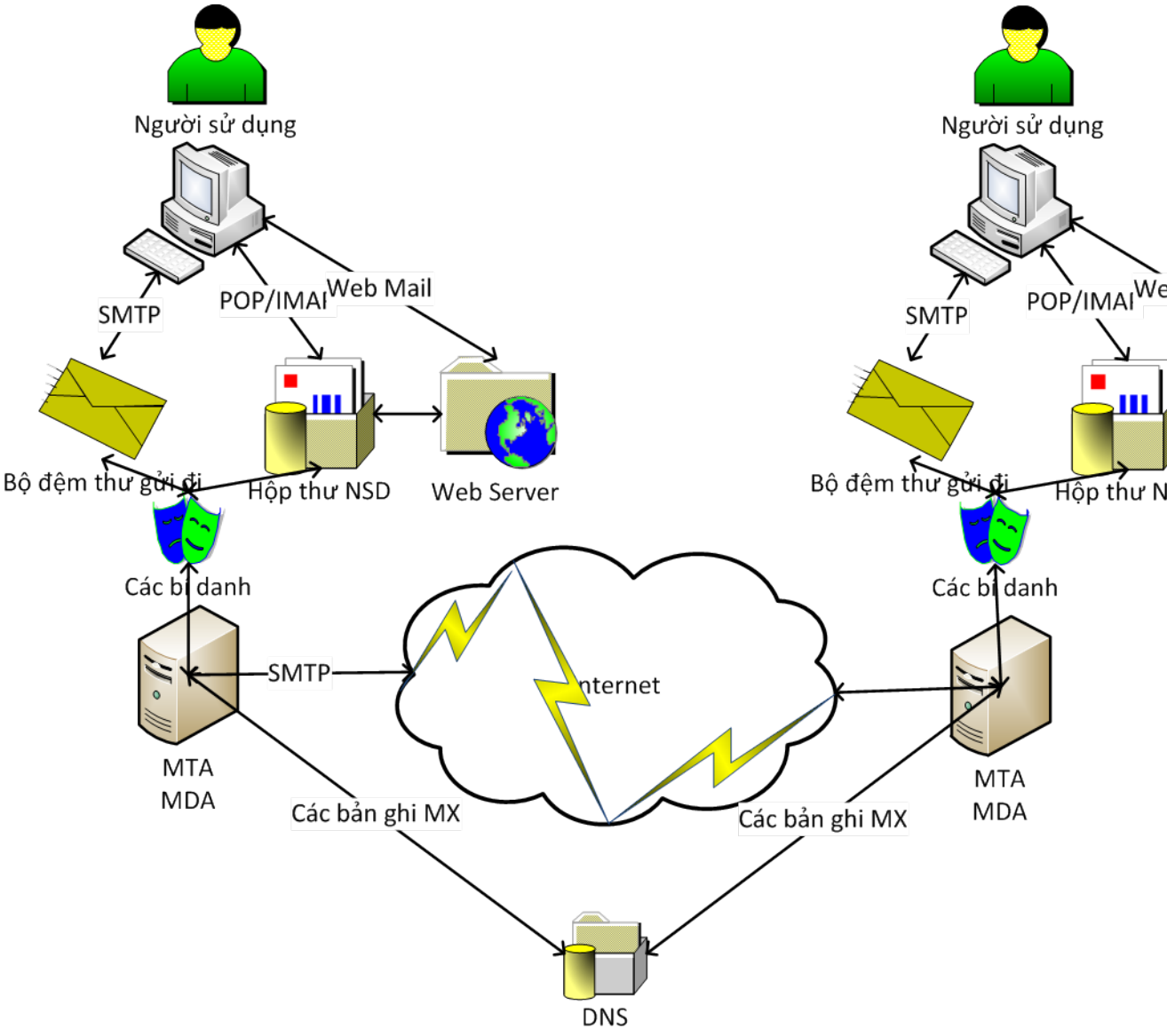
Hình 16.2.1 mô tả các thành phần cơ bản của một hệ thống thư điện tử. Một hệ thống thư điện tử sẽ gồm các thành phần sau:

- Thành phần giao tiếp với các hệ thống thư điện tử khác thường được gọi là MTA (Mail Tranfer Agent), ví dụ: qmail, postfix....
- Thành phần phân phối thư điện tử nhận được đến từng hòm thư của người sử dụng được gọi là MDA (Mail Delivery Agent:).Ví dụ: mail.local, procmail...
- Thành phần giao tiếp với người sử dụng thường được gọi là MUA (Mail User Agent). Ví dụ: Outlook Expresss, Thunderbird...
- Thành phần quản lý truy cập của người sử dụng thông qua các giao thức truy cập từ xa như POP3/IMAP thường được gọi là MAA (Mail Access Agent), ví dụ; dovecot

16.3 Các giao thức của hệ thống email

16.3.1 SMTP - Simple Mail Tranfer Protocol

SMTP (Simple Mail Transfer Protocol) là một trong các giao thức Internet nòng cốt được thiết kế để chuyển thư điện tử một cách tin cậy và hiệu quả. Giao thức SMTP dùng để trao đổi email giữa các thành phần:



Hình 16.2.1: Các thành phần và hoạt động của hệ thống email

- MUA, MTA, MDA: để gửi nhận và phân phối thư
- MX Server - Mail Exchange Server: máy chủ chạy dịch vụ thư điện tử.
- DNS Server - Domain Name Services Server: Máy chủ cung cấp dịch vụ chuyển đổi tên miền thành địa chỉ IP. Giúp cho máy chủ thư điện tử nguồn biết được địa chỉ của máy chủ thư điện tử đích

Các thức gửi nhận thư điện tử của SMTP được miêu tả cụ thể bằng hình ảnh dưới đây:

Đầu tiên người sử dụng dùng MUA để tạo một thư điện tử (ví dụ: 'TenNguoiNhan@TenCongTy.com', cùng với chủ đề (Subject) và nội dung của thông báo). Sau đó MUA sẽ gửi thư điện tử đó đến MTA, MTA phân tích tên miền của địa chỉ e-mail nhận (ví dụ: 'TenCongTy.com') để biết được địa chỉ của người nhận. MTA bắt đầu trao đổi liên lạc với một DNS Server (máy chủ hệ thống tên miền) mà sẽ tìm kiếm và trả về tên (host name) của MTA đích (ví dụ '203.164.1.134') cho tên miền đó. Mỗi một domain nhận thư đều có một MX record trên DNS server để cho biết host nào sẽ nhận thư điện tử cho domain. Khi MTA nhận biết được host nào để gửi thư điện tử đến, nó thiết lập một phiên SMTP để gửi thư điện tử đến (SMTP hand shaking) thông qua cổng 25 của TCP/IP. Nếu tên người dùng của địa chỉ thư điện tử nhận khớp với một trong những tài khoản người dùng được phép trong máy chủ đích, thông báo thư điện tử gốc cuối cùng sẽ được chuyển đến máy chủ này. Sau đó MUA của người nhận sẽ tải thư điện tử về máy theo giao thức POP hoặc IMAP. Một ví dụ về gửi thư bằn SMTP thông qua telnet trực tiếp vào cổng 25 được minh họa trong Hình 16.3.2 Trong trường hợp MTA đầu tiên không thể trao đổi thông tin trực tiếp với máy chủ đích, giao thức SMTP cung cấp các cơ chế để chuyển các thông báo thông qua một hay nhiều MTA chuyển tiếp trung gian. Một máy chủ chuyển tiếp sẽ nhận thông báo gốc và sau đó thử chuyển nó tới máy chủ đích hay gửi nó một lần nữa tới một máy chủ chuyển tiếp khác. Quá trình này sẽ được lặp lại cho đến khi thông báo được chuyển đi hoặc thời gian lưu giữ thông báo hết hạn.

16.3.2 IMAP - Internet Message Access Protocol

Giao thức IMAP (Internet Message Access Protocol) cung cấp lệnh để phần mềm thư điện tử trên máy khách và máy chủ dùng trong trao đổi thông tin. Đó là phương pháp để người dùng cuối truy cập thông điệp thư điện tử hay bảng tin điện tử từ máy chủ về thư trong môi trường cộng tác. Nó cho phép chương trình thư điện tử dùng cho máy khách - như Netscape Mail, Eudora của Qualcomm, Lotus Notes hay Microsoft Outlook - lấy thông điệp từ xa trên máy chủ một cách dễ dàng như trên đĩa cứng cục bộ. IMAP là cơ chế

```
trunghq@ubuntu:~$ telnet 127.0.0.1 25
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
220 ubuntu.localdomain ESMTP Postfix (Ubuntu)
helo trunghq.com
250 ubuntu.localdomain
helo local.domain.name
250 ubuntu.localdomain
MAIL FROM:trunghq@ubuntu.localdomain
250 2.1.0 Ok
RCPT TO:haquoctrung@gmail.com
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Subject:Thu gui thu tu telnet
Day la mot buc thu gui tu telnet
.
250 2.0.0 Ok: queued as 913E010089F
quit
221 2.0.0 Bye
Connection closed by foreign host.
trunghq@ubuntu:~$ tail -3 /var/log/mail.log
May 18 07:34:56 ubuntu postfix/smtpd[3125]: disconnect from localhost[127.0.0.1]
May 18 07:35:02 ubuntu postfix/smtp[3171]: 913E010089F: to=<haquoctrung@gmail.co
dsn=2.0.0, status=sent (250 2.0.0 OK 1274193114 12si8865853pzk.15)
```

Hình 16.3.2: Gửi thư bằng telnet

cho phép lấy thông tin về thư điện tử của bạn, hay chính các thông điệp từ thư điện tử server của môi trường cộng tác. Giao thức thư điện tử này cho phép người dùng kết nối vào máy chủ Internet từ xa, xem xét phần tiêu đề và người gửi của thư điện tử trước khi tải những thư này về máy chủ của mình. Với IMAP người dùng có thể truy cập các thông điệp như chúng được lưu trữ cục bộ trong khi thực tế lại là thao tác trên máy chủ cách xa hàng km. Với khả năng truy cập từ xa này, IMAP dễ được người dùng cộng tác chấp nhận vì họ coi trọng khả năng làm việc lưu động. Người dùng thường xuyên đi lại muốn lưu thông điệp của họ trên máy chủ để đến bất kỳ đầu cuối nào cũng có thể đọc và làm việc được. IMAP cho phép thực hiện điều đó. IMAP khác với giao thức truy cập thư điện tử POP (Post Office Protocol). POP lưu trữ toàn bộ thông điệp trên máy chủ. Người dùng kết nối vào máy chủ và POP sẽ đưa các thông điệp vào Inbox của người dùng, sau đó xóa thư trên máy chủ. Hai giao thức này đã được dùng từ hơn 10 năm nay. Theo một nhà phân tích thì khác biệt chính giữa POP (phiên bản hiện hành 3.0) và IMAP (phiên bản hiện hành 4.0) là POP cho người dùng ít quyền điều khiển hơn trên thông điệp. Hình ảnh IMAP mang lại cho người dùng một phương thức lưu trữ thư điện tử thông minh và nhờ đó có thể xem những thông điệp này trước khi tải chúng xuống, bao gồm cả việc có tải xuống những file đính kèm thư hay không. Người dùng cũng có thể áp dụng các bộ lọc và cơ chế tìm kiếm trên máy chủ và có thể lấy thư từ bất kỳ máy nào, bất cứ ở đâu. Tuy nhiên, các nhà sản xuất đã thông dịch các đặc tả mơ hồ của IMAP 4 theo nhiều cách khác nhau và điều đó dẫn đến sự không nhất quán trong chương trình thư dành cho máy khách và máy chủ, chẳng hạn người dùng có thể sẽ không đọc được file đính kèm trong Netscape thư điện tử bằng chương trình Eudora Pro. Tuy nhiên, theo dự đoán những vấn đề này sẽ nhanh chóng được giải quyết trong thời gian tới.

16.3.3 POP - Post Office Protocol

Giao thức POP3 (Post Office Protocol) là giao thức dùng để một user nhận thư điện tử từ một thư điện tử server. POP lưu trữ toàn bộ thông điệp trên máy chủ. Người dùng kết nối vào máy chủ và POP sẽ đưa các thông điệp vào Inbox của người dùng, sau đó xóa thư trên máy chủ. Quá trình diễn ra như sau:

- Đầu tiên user muốn nhận thư điện tử của người gửi, vì vậy phải thiết lập một phiên kết nối đến máy chủ bằng giao thức POP thông qua cổng 110 (bắt tay ba bước)
- Máy chủ thư điện tử sử dụng giao thức POP3 trả lời tới user rằng đã sẵn sàng truyền thư điện tử tới user bằng một bản tin "OK POP3

TenUser".

- Client gửi lại username là: user1@domain.com
- Máy chủ kiểm tra tên user xem có hợp lệ không, nếu có sẽ gửi tiếp cho client bản tin yêu cầu gửi mật khẩu tương ứng
- Client gửi mật khẩu và nếu đúng máy chủ sẽ cho phản hồi bằng bản tin [OK] cho phép client tải thư về
- Công việc còn lại của bạn là đọc thư điện tử của người gửi. Sau khi quá trình nhận thư điện tử đã hoàn thành thì user sử dụng giao thức TCP gửi bản tin [FIN,ACK] báo với thư điện tử server rằng đã nhận được thư điện tử và xin kết thúc phiên kết nối, các quá trình tiếp theo đó chính là quá trình kết thúc phiên kết nối trong giao thức TCP.

16.4 Hệ thống thư điện tử

Một hệ thống thư điện tử thường có hai thành phần chính là mail server và mail client, hai thành phần này có thể nằm trên hai hệ thống riêng (máy chủ và phần mềm mail client) hoặc trên cùng một hệ thống (máy chủ và webmail). Ngoài ra nó còn có thể có thêm hệ thống mail host hoặc mail gateway.

Mail Server : Máy chủ chứa mailbox của người sử dụng, mail server cũng là máy chủ làm nhiệm vụ giao tiếp để gửi nhận thư với các máy chủ bên ngoài.

Mail Client : Các chương trình hỗ trợ người sử dụng đọc, soạn thảo và quản lý thư. Mail client thường tích hợp các giao thức SMTP, POP, IMAP giúp người sử dụng có thể truy cập vào mail server để đọc, gửi hoặc tải thư về máy

Mail gateway : Máy chủ thư điện tử được sử dụng trong các mô hình của hệ thống thư điện tử hiện đại ngày nay, mail gateway đứng ở bên ngoài vùng local, nhận thư và gửi về các mail server local tương ứng cũng như đóng vai trò định tuyến để các mail server local này gửi thư ra các máy chủ khác bên ngoài.

16.5 Cài đặt mail trên Linux

Trong cuốn sách này tác giả minh họa hệ thống email bằng việc dựng hệ thống postfix trên hệ điều hành Debian.

16.5.1 Cài đặt và cấu hình cơ bản Postfix

Trước khi cài đặt postfix cần cài đặt mysql, apache2. Ở đây tác giải đi thẳng vào phần cài đặt postfix.

```
debian:~# apt-get install postfix
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  openssl openssl-blacklist ssl-cert
Suggested packages:
  ca-certificates postfix-mysql postfix-pgsql postfix-ldap postfix-pcre sasl2-bi
The following packages will be REMOVED:
  exim4 exim4-base exim4-config exim4-daemon-light
The following NEW packages will be installed:
  openssl openssl-blacklist postfix ssl-cert
0 upgraded, 4 newly installed, 4 to remove and 0 not upgraded.
Need to get 8611kB of archives.
After this operation, 14.0MB of additional disk space will be used.
Do you want to continue [Y/n]? y

//Một vài câu hỏi cần trả lời trong quá trình cài đặt
New password for the MySQL "root" user: <-- yourrootsqlpassword
Repeat password for the MySQL "root" user: <-- yourrootsqlpassword
Create directories for web-based administration? <-- No
General type of mail configuration: <-- Internet Site
System mail name: <-- server1.example.com
SSL certificate required <-- Ok
Workgroup/Domain Name: <-- WORKGROUP
Modify smb.conf to use WINS settings from DHCP? <-- No
Web server to reconfigure automatically: <-- apache2
```

Sau khi cài đặt postfix tự động kích hoạt, để test hệ thống có thể thực hiện gửi thư bằng câu lệnh như sau:

```
debian:~# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 server1.example.com ESMTP Postfix (Debian/GNU)
ehlo localhost
250-server1.example.com
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-AUTH LOGIN PLAIN
250-AUTH=LOGIN PLAIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
quit
221 2.0.0 Bye
Connection closed by foreign host.
debian:~#
```

16.5.2 Cài đặt và cấu hình cơ bản Squirrel mail

Tiến hành cài đặt webmail Squirrel

```
debian:~# apt-get install squirrelmail php-pear
```

Cấu hình Squirrel mail

```
debian:~# cp /etc/squirrelmail/apache.conf /etc/apache2/conf.d/squirrelmail.conf
debian:~# /etc/init.d/apache2 restart
debian:~# pear install DB
```

Thử nghiệm giao điện đồ họa bằng cách truy cập vào: http://IPMayChu/squirrelmail

16.5.3 Cài đặt và cấu hình cơ bản Claimav

Cài Amavis để kết nối postfix với ClamAV

```
debian:~# apt-get install amavisd-new clamav clamav-daemon
debian:~# adduser clamav amavis
debian:~# /etc/init.d/amavis restart
```

```
debian:~# /etc/init.d/clamav-daemon restart
debian:~# /etc/init.d/clamav-freshclam restart

//Cấu hình để postfix gọi amavis quét các thư gửi hoặc nhận được
debian:~# postconf -e 'content_filter = amavis:[127.0.0.1]:10024'
debian:~# postconf -e 'receive_override_options = no_address_mappings'

//khởi động lại postfix để thay đổi có hiệu lực
debian:~# /etc/init.d/postfix restart
```

16.5.4 Thử nghiệm với thunderbird

[]*

Tài liệu tham khảo

[1] CREATIVE-COMMONS-CORPORATION. Official ubuntu documentation, October 2009.

[2] The Apache Software Foundation. Apache http server version 2.2 documentation, 2009.

[3] N/A. http://www.wikipedia.org/, 2009.

[4] Inc. Red Hat. Red hat linux reference guide, 2003.

[5] Inc. Red Hat. Red hat linux security guide, 2003.

[6] Inc. Red Hat. Red hat linux system administration primer, 2003.

[7] Inc. Red Hat. Red hat linux x86 installation guide, 2003.