

LỜI GIỚI THIỆU

Vào tháng 3-2004 tại Hà Nội, Bộ Khoa học và Công nghệ cùng với CICC (Trung tâm Hợp tác Quốc tế về Tin học hoá, Nhật Bản) tổ chức Diễn đàn Châu Á lần thứ 3 về phần mềm nguồn mở (OSS). Năm 2000 và 2002 hội nghị toàn quốc về Linux cũng đã họp tại Hà Nội. Năm 2003, một số quốc gia như Trung Quốc, Nhật Bản và Hàn Quốc đã quyết định đưa việc bản địa hoá Linux vào kế hoạch phát triển công nghệ phần mềm của mình. Còn ở Việt Nam cũng đã có khoảng hai dự án với ý định như thế.

Trên đà phát triển của công nghệ thông tin như hiện nay, Linux đã không ngừng mở rộng ra ngoài phạm vi nghiên cứu đại học để phục vụ cho mục đích thương mại và hành chính, hoặc dùng làm hệ điều hành cho các mạng máy tính. Quả thật Linux đã tiến triển và hoàn thiện liên tục với những phiên bản mới, thậm chí năm 2003 các dòng Linux ManDrake và RedHat v.v. đều đã có đến bản 9.0 và hiện nay là 11.0. Mặt khác Linux càng ngày càng có thêm nhiều người sử dụng vì vậy mà nhu cầu cần có một cuốn giáo trình làm tài liệu học tập và tham khảo cho học sinh, sinh viên của các trường Cao đẳng, Đại học là cần thiết.

Các bài trong giáo trình này chủ yếu sẽ hướng dẫn cách cài đặt và sử dụng dòng sản phẩm RedHat vì có lẽ đó là dòng Linux phổ biến nhất và cũng dễ cài đặt nhất từ trước đến nay. Ngoài ra, giáo trình còn cung cấp những hiểu biết khác, thí dụ cập nhật và nâng cấp các phần mềm tương hợp với Linux, hoặc in ấn, hỗ trợ an ninh và quản trị hệ thống một cách thuận tiện.

Cuốn giáo trình này cũng phù hợp cho những người muốn biết thêm về Linux và UNIX mà chưa có dịp sử dụng hai hệ điều hành ấy. Thậm chí, giáo trình sẽ có ích với những người tuy biết cách cài đặt Linux và sử dụng UNIX, nhưng chưa có dịp thực hiện các công việc quản trị hệ thống bao giờ. Cuốn giáo trình sau sẽ giải thích chi tiết về cách quản trị và duy trì hệ thống Linux/UNIX. Một người sử dụng UNIX bình thường khó có quyền làm quản trị hệ thống, song với Linux thì có thể trở thành chủ nhân của toàn bộ hệ thống.

Linux dẫn xuất từ UNIX nên cũng là một hệ điều hành đa người dùng và đa nhiệm (phục vụ nhiều người và thực hiện nhiều việc cùng lúc). Nó có thể chạy trên nhiều bộ vi xử lý (đặc biệt trên họ Intel từ đời 386 trở lại đây) và tương thích với chuẩn mở POSIX. POSIX là một tiêu chuẩn quốc tế cho các hệ điều

hành và phần mềm khả chuyển với những thành phần có thể sử dụng chung, đảm bảo tính mở của chúng.

Trong quá trình biên soạn, dù đã hết sức cố gắng, song không tránh khỏi những sai sót. Rất mong nhận được những ý kiến đóng góp và nhận xét của các thầy cô, các em học sinh, sinh viên và những ai sử dụng cuốn giáo trình này.

Hà Nội, 2013

Tham gia biên soạn

Khoa Công Nghệ Thông Tin

Trường Cao Đẳng Nghề Kỹ Thuật Công Nghệ

Địa Chỉ: Tổ 59 Thị trấn Đông Anh – Hà Nội

Tel: 04. 38821300

Chủ biên: Phùng Quốc Cảnh

*Mọi góp ý liên hệ: Phùng Sỹ Tiến – Trưởng Khoa Công Nghệ Thông Tin
Mobile: 0983393834
Email: tienphungktn@gmail.com – tienphungktn@yahoo.com*

MỤC LỤC

ĐỀ MỤC	TRANG
<u>LỜI GIỚI THIỆU.....</u>	<u>1</u>
<u>MỤC LỤC.....</u>	<u>4</u>
<u>ĐỀ MỤC TRANG.....</u>	<u>4</u>
<u>MÔ ĐUN: HỆ ĐIỀU HÀNH MÃ NGUỒN MỞ (LINUX).....</u>	<u>10</u>
<u>BÀI 1: TỔNG QUAN VỀ LINUX.....</u>	<u>11</u>
<u>A. LÝ THUYẾT.....</u>	<u>12</u>
<u>1. Tìm hiểu chung về Linux</u>	<u>12</u>
<u>Mục tiêu :.....</u>	<u>12</u>
- <u>Nêu được khái niệm Linux</u>	<u>12</u>
- <u>Trình bày được Các bản phát hành Linux</u>	<u>12</u>
<u>1.1. Linux là gì.....</u>	<u>12</u>
<u>1.3. Các bản phát hành Linux.....</u>	<u>14</u>
<u>1.4. Lợi thế của Linux.....</u>	<u>14</u>
<u>1.5. Ai phát triển Linux.....</u>	<u>16</u>
<u>1.6. Linux cộng sinh với Windows.....</u>	<u>16</u>
<u>1.7. Thương mại hoá Linux.....</u>	<u>17</u>
<u>2. Unix và Linux</u>	<u>18</u>
<u>Mục tiêu :.....</u>	<u>18</u>
- <u>Trình bày được lịch sử phát triển của hệ điều hành UNIX và Linux</u>	<u>18</u>
<u>3. Tác quyền và bản quyền Linux</u>	<u>20</u>
<u>BÀI 2: CHUẨN BỊ CÀI ĐẶT LINUX.....</u>	<u>21</u>
<u>A. LÝ THUYẾT.....</u>	<u>21</u>
<u>1. Chọn cấu hình phần cứng</u>	<u>21</u>
<u>1.1. Bộ xử lý.....</u>	<u>22</u>
<u>1.2. Bus hệ thống.....</u>	<u>22</u>
<u>1.3. Bộ nhớ.....</u>	<u>22</u>
<u>1.4. Đĩa cứng.....</u>	<u>23</u>
<u>1.4.1. Dung lượng ổ đĩa cứng.....</u>	<u>23</u>
<u>1.4.2. Phân vùng hoán chuyển.....</u>	<u>23</u>
<u>1.5. Yêu cầu về màn hình.....</u>	<u>24</u>
<u>1.6. Ổ CD.....</u>	<u>24</u>
<u>1.6.1 Các ổ đĩa CD phổ quát.....</u>	<u>25</u>
<u>1.6.2. Các ổ đĩa CD đặc chủng.....</u>	<u>25</u>
<u>1.7. Truy cập mạng.....</u>	<u>25</u>
<u>1.7.1. Truy cập qua Ethernet.....</u>	<u>25</u>
<u>1.7.2. Truy cập qua modem.....</u>	<u>27</u>
<u>1.8. Các thiết bị khác.....</u>	<u>28</u>
<u>1.8.1. Chuột.....</u>	<u>28</u>
<u>1.8.2. Ổ băng từ.....</u>	<u>28</u>

1.8.3. Máy in.....	28
2. Dung lượng đĩa và bộ nhớ	29
3. Phân vùng ổ đĩa cứng	30
Mục tiêu :.....	30
- Biết được cách phân vùng ổ đĩa cứng.....	30
- Sử dụng được các lệnh để phân vùng.....	30
3.1. Tìm hiểu về phân vùng.....	30
3.2. Sử dụng lệnh FDISK.....	31
3.2.1. Các yêu cầu về phân vùng.....	31
3.2.2. Các yêu cầu về DOS.....	31
3.2.3. Các yêu cầu về Linux.....	31
3.2.4. Phân vùng lại ổ DOS.....	32
3.2.5. Cách tránh phân vùng đĩa cứng.....	32
3.2.6. Xoá bỏ phân vùng.....	33
3.2.7. Thêm phân vùng mới.....	33
3.2.8. Định dạng phân vùng.....	33
B. CÂU HỎI VÀ BÀI TẬP.....	34
BÀI 3: CÀI ĐẶT REDHAT LINUX.....	35
A. LÝ THUYẾT.....	35
1. Các cách cài đặt& trình tự cài đặt.....	35
1.1. Các cách cài đặt.....	35
1.2. Trình tự cài đặt	36
1.2.1. Cấu hình hệ thống.....	37
1.2.2. Tùy chọn cài đặt.....	40
1.2.3. Phân vùng đĩa cứng.....	41
1.2.3.1. Sử dụng fdisk của Linux.....	41
1.2.3.2. Sử dụng Disk Druid.....	46
1.2.3.3. Phân vùng tự động.....	49
1.2.4. Cài đặt chương trình khởi động.....	50
1.2.4.1. Các chương trình khởi động khác.....	52
1.2.4.2. Mật khẩu cho GRUB.....	52
2. Thiết lập các cấu hình cho RedHat.....	52
2.1. Thiết lập cấu hình mạng TCP/IP.....	52
2.2. Cấu hình bức tường lửa.....	53
2.3. Các thiết lập khác.....	54
2.3.1. Hỗ trợ ngôn ngữ.....	54
2.3.2. Thiết lập cấu hình thời gian.....	54
2.4. Thiết lập trương khoản người dùng.....	55
2.5. Thiết lập cấu hình xác thực.....	57
3. Chọn các gói phần mềm & cài đặt.....	58
4. Thiết lập cấu hình &Kiểm tra cấu hình X Window	64

4.1. Thiết lập cấu hình X Window.....	64
4.2. Kiểm tra cấu hình X Window.....	65
5. Tạo đĩa mềm khởi động&khởi động lại	65
Mục tiêu :.....	65
- Tạo được đĩa mềm khởi động và khởi động lại.....	65
5.1. Tạo đĩa mềm khởi động.....	65
5.2. Khởi động lại.....	65
6. Cài đặt Linux ở chế độ văn bản	65
6.1. Cấu hình phần cứng cơ bản.....	66
6.2. Các màn hình ở chế độ văn bản.....	66
6.3. Dùng bàn phím để di chuyển.....	66
6.4. Cài đặt ở chế độ văn bản từ đĩa CD.....	66
B. CÂU HỎI VÀ BÀI TẬP.....	67
BÀI 4: BẮT ĐẦU SỬ DỤNG LINUX.....	67
A. LÝ THUYẾT.....	67
1. Thiết lập tài khoản	67
1.1. Giao tiếp qua dòng lệnh.....	68
1.2. Lịch trình nhập lệnh.....	68
1.3. Nhập lệnh bằng sao ghép.....	69
1.4. Tự động điền lệnh.....	69
2. Quản lý người sử dụng.....	69
Mục tiêu :.....	69
- Đăng nhập và đăng xuất được một user.....	69
- Thêm người sử dụng trong Slackware.....	69
2.1. Đăng nhập và đăng xuất.....	69
2.2. Thêm người sử dụng trong Slackware.....	70
2.3. Thêm người sử dụng mới trong RedHat Linux.....	73
2.4. Dùng bảng điều khiển RedHat để quản lý người sử dụng.....	73
2.5. Thay đổi mật khẩu.....	74
3. Sử dụng các lệnh cơ bản	75
Mục tiêu :.....	75
- Biết cách dùng man để tìm trợ giúp cho câu lệnh.....	75
3.1. Dùng man để tìm trợ giúp cho câu lệnh.....	75
3.2. Sử dụng các lệnh can thiệp vào thư mục.....	76
3.2.1. Chuyển đổi thư mục hiện hành bằng lệnh cd.....	76
3.2.2. Liệt kê các tệp và thư mục bằng lệnh ls.....	76
3.2.3. Tạo thư mục mới bằng lệnh mkdir.....	77
3.2.4. Xoá bỏ thư mục bằng lệnh rmdir.....	77
3.3. Sử dụng các lệnh thao tác tệp.....	77
3.3.1. Chép các tệp bằng lệnh cp.....	77
3.3.2. Chuyển tệp bằng lệnh mv.....	78

3.3.3. Xoá tệp bằng lệnh rm.....	78
3.3.4. Hiển thị nội dung tệp bằng lệnh more.....	78
3.3.5. Sử dụng lệnh less.....	79
4. Xử lý các tệp DOS trong Linux.....	79
Mục tiêu :.....	79
5. Đóng tắt Linux và chạy các chương trình Linux	80
5.1. Đóng tắt Linux.....	80
5.2. Chạy các chương trình Linux.....	81
5.2.1. Sử dụng chương trình CD Player.....	81
5.2.2. Sử dụng Gnumeric và KSpread.....	81
5.2.3. Sử dụng bc Calculator.....	82
5.2.4. Sử dụng chương trình minicom.....	82
6. Chạy các chương trình DOS trong Linux	83
6.1. Cài đặt DOSEMU.....	84
6.2. Lập cấu hình DOSEMU.....	84
6.3. Chạy DOSEMU.....	85
7. Chạy các chương trình Windows với Linux	86
B. CÂU HỎI VÀ BÀI TẬP.....	87
Last login: Wed Apr 7 08:35:50 from 131.16.16.21 [user12@linux user12]\$......	88
3. Thoát khỏi hệ thống.....	88
Thoát khỏi phiên làm việc : #exit hoặc #logout.....	88
Chấm dứt hoạt động của hệ thống : #shutdown -h now.....	88
BÀI 5: NÂNG CẤP VÀ CÀI ĐẶT PHẦN MỀM VỚI RPM.....	88
A. LÝ THUYẾT.....	89
1. Chính sách nâng cấp phần mềm	89
2. Cài đặt phần mềm	89
Mục tiêu :.....	89
- Nêu được các công việc và trách nhiệm của quản trị viên hệ thống	89
2.1. Giới thiệu.....	89
2.2. Công việc của quản trị viên hệ thống.....	90
3. Sử dụng RPM	91
3.1. Vị trí của các gói phần mềm.....	92
3.2. Cài đặt gói phần mềm bằng RPM.....	92
3.3. Gỡ bỏ cài đặt gói phần mềm bằng RPM.....	93
3.4. Cập nhật gói phần mềm bằng RPM.....	94
3.5. Tìm các gói phần mềm.....	94
3.6. Kiểm tra gói phần mềm.....	95
3.7. Cài đặt phần mềm không của Linux.....	96
3.7.1. Các định dạng của gói phần mềm.....	96
3.7.2. Cài đặt phần mềm.....	96
3.7.3. Sử dụng lệnh tar.....	98

3.8. Xem lại các quyền truy cập.....	99
3.9. Giải quyết vấn đề.....	99
3.10. Gỡ bỏ các ứng dụng.....	100
4. Nâng cấp Kernel	100
5. Cài đặt trong môi trường X bằng RPM	101
5.1. Khởi động GNOME-RPM.....	101
5.2. Chọn gói phần mềm.....	101
5.3. Cài đặt phần mềm mới.....	102
5.4. Lập cấu hình mặc định cho trình cài đặt.....	102
5.5. Gỡ bỏ phần mềm	102
B. CÂU HỎI VÀ BÀI TẬP.....	102
BÀI 6: QUẢN TRỊ HỆ THỐNG LINUX.....	102
A. LÝ THUYẾT.....	103
1. Các hệ thống và các thành phần xử lý	103
Mục tiêu :.....	103
- Trình bày được các thành phần của hệ thống xử lý tập trung.....	103
- Trình bày được các thành phần của hệ thống xử lý phân tán.....	103
1.1. Các hệ thống xử lý tập trung.....	103
1.2. Các thành phần của mô hình xử lý tập trung.....	104
1.3. Các hệ thống xử lý phân tán.....	105
1.4. Các thành phần của mô hình xử lý phân tán.....	105
2. Các mô hình và quản trị trong môi trường mạng.....	106
2.1. Mô hình client/server.....	106
2.2. Quản trị trong môi trường mạng.....	107
2.3. Xác định vai trò quản trị viên mạng.....	107
2.4. Lựa chọn phần cứng và phần mềm.....	107
2.5. Những công việc chung trong quản trị mạng.....	108
2.5.1. Thiết lập hệ thống.....	108
2.5.2. Thao tác các thiết bị ngoại vi	109
2.5.3. Giám sát hệ thống.....	109
2.5.4. Nâng cấp phần mềm.....	109
2.6. Huấn luyện quản trị viên.....	110
B. CÂU HỎI VÀ BÀI TẬP.....	111
.....	112
Cấu hình dịch vụ mạng bằng tiện ích redhat-config-network.....	112
Câu 6: Thực hành cấu hình dịch vụ DNS?.....	112
e. Tập tin /var/named/edu.vn.zone.....	113
h. Tập tin /var/named/localhost.zone.....	114
BÀI 7: KHỞI ĐỘNG VÀ ĐÓNG TẮT.....	116
A. LÝ THUYẾT.....	117
1. Trình quản lý môi LILO	117

1.1. Thiết lập cấu hình LILO.....	117
1.2. Sử dụng LILO.....	118
2. Tiến trình khởi động	118
3. Đóng tắt Linux	126
Mục tiêu :.....	126
- Nêu được các lưu ý khi đóng tắt linux.....	126
- Đóng tắt được hệ thống Linux bằng các lệnh cơ bản	126
B. CÂU HỎI VÀ BÀI TẬP.....	127
BÀI 8: QUẢN LÝ TÀI KHOẢN.....	127
A. LÝ THUYẾT.....	127
1. Làm việc với các user và nhóm user	127
1.1. Làm việc với các user.....	128
1.1.1. Thêm vào một user.....	128
1.1.2. Sử dụng lệnh adduser.....	129
1.1.3. Thiết lập mật khẩu cho user.....	130
1.1.4. Gỡ bỏ một user.....	131
1.2. Làm việc với nhóm.....	132
1.2.1. Thêm vào một nhóm.....	132
1.2.2. Xoá bỏ một nhóm.....	132
2. Quản lý home directory.....	132
Mục tiêu :.....	132
- Trình bày được cách quản lý home directory	132
3. Quản trị qua giao diện web	133
Mục tiêu :.....	133
- Quản trị được các user và nhóm qua giao diện web	133
B. CÂU HỎI VÀ BÀI TẬP.....	133
BÀI 9: SAO LƯU DỮ LIỆU.....	137
A. LÝ THUYẾT.....	137
1. Vấn đề về sao lưu.....	137
2. Các thủ thuật sao lưu	138
Mục tiêu :.....	138
- Trình bày được các thủ thuật sao lưu hệ thống	138
3. Hoạch định thời biểu sao lưu.....	139
Mục tiêu :.....	139
- Nêu được ý nghĩa của việc hoạch định thời biểu sao lưu.....	139
- Hoạch định được thời biểu của việc sao lưu dữ liệu	139
4. Thực hiện sao lưu và phục hồi tệp	140
4.1. Tiên ích tar.....	140
4.2. Sử dụng cpio.....	143
B. CÂU HỎI VÀ BÀI TẬP.....	144
BÀI 10: QUẢN LÝ TỆP VÀ THƯ MỤC.....	144

<u>A. LÝ THUYẾT.....</u>	<u>145</u>
<u>1. Các thao tác cơ bản với tệp</u>	<u>145</u>
<u>Mục tiêu :.....</u>	<u>145</u>
<u>- Trình bày được các thao tác cơ bản với tệp.....</u>	<u>145</u>
<u>- Thực hiện được thành thạo các thao tác cơ bản với tệp</u>	<u>145</u>
<u>1.1. Liệt kê tệp.....</u>	<u>145</u>
<u>1.2. Tổ chức tệp.....</u>	<u>148</u>
<u>1.3. Sao chép tệp.....</u>	<u>149</u>
<u>1.4. Di dời và đặt tên lại tệp.....</u>	<u>149</u>
<u>1.5. Xoá tệp hoặc thư mục.....</u>	<u>150</u>
<u>1.6. Xem nội dung của tệp</u>	<u>151</u>
<u>1.6.1. Các thiết bị xuất nhập chuẩn.....</u>	<u>152</u>
<u>1.6.2. Xem tệp bằng lệnh cat.....</u>	<u>152</u>
<u>1.6.3. Xem tệp bằng lệnh more.....</u>	<u>152</u>
<u>1.6.4. Xem tệp bằng lệnh less.....</u>	<u>153</u>
<u>1.6.5. Duyệt tìm xuyên tệp và thoát khỏi shell.....</u>	<u>153</u>
<u>1.6.6. Xem tệp bằng những cách khác.....</u>	<u>153</u>
<u>1.7. Duyệt tìm tệp.....</u>	<u>155</u>
<u>1.8. Thay đổi nhãn ngày giờ.....</u>	<u>157</u>
<u>2. Nén và nối tệp</u>	<u>157</u>
<u>3. Hệ thống thư mục trong Linux</u>	<u>158</u>
<u>Mục tiêu :.....</u>	<u>158</u>
<u>- Trình bày được thư mục UNIX cổ điển</u>	<u>158</u>
<u>- Nêu được Các thư mục trong Linux</u>	<u>158</u>
<u>3.1. Thư mục UNIX cổ điển.....</u>	<u>158</u>
<u>3.2. Các thư mục trong Linux.....</u>	<u>160</u>
<u>B. CÂU HỎI VÀ BÀI TẬP.....</u>	<u>161</u>
<u>CÂU HỎI ÔN TẬP TRẮC NGHIỆM.....</u>	<u>164</u>
<u>TÀI LIỆU THAM KHẢO.....</u>	<u>192</u>

MÔ ĐUN: HỆ ĐIỀU HÀNH MÃ NGUỒN MỞ (LINUX)

❖ **Mã mô đun:** MĐ37

❖ **Vị trí, ý nghĩa, vai trò của mô đun:**

- Vị trí:

+ Mô đun được bố trí sau khi học xong các môn chung và bố trí song song với môn học/mô đun đào tạo chuyên ngành.

- Tính chất:

- + Là mô đun chuyên ngành tự chọn.
- Ý nghĩa, vai trò của mô đun:
 - + Mô đun này cung cấp các thông tin về bản quyền hệ điều hành và phần mềm mã nguồn mở có chi phí rẻ hơn so với các phần mềm truyền thống, mặt khác rất dễ nâng cấp, cải tiến.
 - + Ngoài ra còn giúp cho chúng ta biết được nhiều ưu điểm nổi bật của hệ điều hành Linux như: Độ an toàn cao, tích hợp cho quản trị mạng,...
 - + Đối với những người làm tin học, đặc biệt là đối với sinh viên, thì việc tìm hiểu và nghiên cứu hệ điều hành Linux và phần mềm mã nguồn mở là một điều kiện rất tốt để nâng cao hiểu biết của mình.

❖ **Mục tiêu của mô đun:**

- Hiểu được nguyên lý hệ điều hành Linux, các yếu tố hợp thành hệ điều hành Linux. Biết chọn được phần cứng thích hợp để cài được hệ điều hành Linux, cài được HĐH Linux.
- Cài đặt các phần mềm và các ứng dụng trên hệ Linux, sử dụng được một số ứng dụng cơ bản trên hệ Linux, biết cơ bản về quản lý hệ thống Linux, các tập tin, thư mục, tài khoản, phân chia quyền hạn...
- Có ý thức về vấn đề bản quyền phần mềm.
- Rèn luyện tinh thần chia sẻ, giúp đỡ nhau cùng phát triển.

❖ **Nội dung mô đun:**

Mã bài	Tên bài	Thời lượng			
		Tổng số	Lý thuyết	Thực hành	Kiểm tra
MĐ37-01	Bài 1: Tổng quan về Linux.	04	03	01	
MĐ37-02	Bài 2: Chuẩn bị cài đặt Linux	04	01	02	01
MĐ37-03	Bài 3: Cài đặt RedHat Linux	12	04	06	02
MĐ37-04	Bài 4: Bắt đầu sử dụng Linux	8	04	04	
MĐ37-05	Bài 5: Nâng cấp và cài đặt phần mềm với RPM	16	04	10	02
MĐ37-06	Bài 6: Quản trị hệ thống Linux	28	10	16	02
MĐ37-07	Bài 7: Khởi động và đóng tắt	04	01	03	
MĐ37-08	Bài 8: Quản lý tài khoản	20	06	12	02
MĐ37-09	Bài 9: Sao lưu dữ liệu	08	01	07	
MĐ37-10	Bài 10: Quản lý tệp và thư mục	16	02	13	01

BÀI 1: TỔNG QUAN VỀ LINUX

Mã bài: **MĐ37-01**

❖ **Giới thiệu**

Bài này chủ yếu dành cho các nhà quản lý dự án công nghệ thông tin. Tuy không thật cần thiết cho việc cài đặt và sử dụng Linux, nhưng nội dung của nó cũng có thể bổ ích cho bất kỳ ai muốn tìm hiểu về những chủ đề sau đây:

- *Linux là gì?*
- *Tại sao Linux phát triển?*
- *Các bản phát hành Linux*
- *Lợi ích của Linux*
- *Ai phát triển Linux?*
- *Linux cộng sinh với Windows*
- *Thương mại hóa Linux*
- *UNIX và Linux*
- *Tác quyền và bản quyền Linux*

❖ Mục tiêu

- **Nắm được kiến thức chung về sự ra đời, mục đích ra đời của hệ điều hành Linux, các giai đoạn phát triển cũng như các phiên bản của hệ điều hành Linux**

- **Nắm được sự khác biệt giữa Linux và Unix, và sinh viên sẽ hiểu rõ hơn về tác giả và bản quyền của hệ điều hành Linux trước khi sử dụng chúng.**

- Nhận thức về bản quyền phần mềm.
- Nâng cao tính chia sẻ cộng đồng.

❖ Nội dung chính

A. LÝ THUYẾT

1. Tìm hiểu chung về Linux

Mục tiêu :

- *Nêu được khái niệm Linux*
- *Trình bày được Các bản phát hành Linux*
- *Biết được lợi ích của Linux*

1.1. Linux là gì

Linux xuất hiện như một sản phẩm nguồn mở miễn phí và đến nay đã có thể sánh vai với các hệ điều hành thương phẩm như MS Windows, Sun Solaris v.v. Linux ra đời từ một dự án hồi đầu những năm 1990 có mục đích tạo ra một hệ điều hành kiểu UNIX cài đặt trên máy tính cá nhân chạy với bộ vi xử lý Intel, tương hợp họ máy tính IBM-PC (còn gọi tắt là PC). Từ lâu, UNIX đã nổi tiếng là một hệ điều hành mạnh, tin cậy và linh hoạt, nhưng vì khá đắt nên chủ yếu chỉ dùng cho các trạm tính toán hoặc máy chủ cao cấp.

Ngày nay Linux có thể cài đặt trên nhiều họ máy tính khác nhau, không chỉ riêng cho họ PC. Qua Internet, Linux được hàng nghìn nhà lập trình khắp trên thế giới tham gia thiết kế, xây dựng và phát triển, với mục tiêu không lệ thuộc vào bất kỳ thương phẩm nào và để cho mọi người đều có thể sử dụng thoải mái. Khởi thủy, Linux xuất phát từ ý tưởng của Linus Torvalds, khi đó chàng sinh viên

Đại học Helsinki ở Phần Lan đã muốn thay thế Minix, một hệ điều hành nhỏ kiểu UNIX.

Về cơ bản, Linux bắt chước UNIX cho nên cũng có nhiều ưu điểm của UNIX. Tính đa nhiệm thực sự của Linux cho phép chạy nhiều chương trình cùng lúc. Với Linux, chúng ta có thể đồng thời thực hiện một số thao tác, thí dụ chuyển tệp, in ấn, sao tệp, nghe nhạc, chơi game v.v.

Linux là hệ điều hành đa người dùng, nghĩa là nhiều người có thể đăng nhập và cùng lúc sử dụng một hệ thống. Ưu điểm này có vẻ không phát huy mấy trên máy PC ở nhà, song ở trong công ty hoặc trường học thì nó giúp cho việc dùng chung tài nguyên, từ đó giảm thiểu chi phí đầu tư vào máy móc.

Ngay cả khi ở nhà, chúng ta cũng có thể đăng nhập vào Linux với nhiều trương khoản (account) khác nhau qua các terminal ảo và tổ chức dịch vụ trên mạng riêng cho mình bằng cách sử dụng Linux với nhiều modem.

Có thể kể tên các hệ điều hành miễn phí khác như FreeBSD, OpenBSD, NetBSD v.v. Cũng phải kể đến ảnh hưởng lớn công ty Sun (chủ nhân của ngôn ngữ Java) vì Sun muốn cung cấp hệ điều hành Solaris dùng miễn phí trên máy PC. Phiên bản Solaris chạy trên chip Intel sẽ trở nên một đối thủ đáng gờm của Linux với mã nguồn mở và nhờ danh tiếng là hệ điều hành rất ổn định và tương thích với hệ Solaris chạy trên chip Sun SPARC.

Bản thân việc độc lập với những công ty lớn cũng tiềm tàng một điểm yếu của Linux. Khi chưa có một mạng lưới riêng cung cấp dịch vụ bảo trì thì tất nhiên người ta sẽ ngại sử dụng Linux. Tuy thế, với sự phát triển của Internet, các tổ chức hỗ trợ người dùng Linux đã tạo nên các Website và forum để tháo gỡ cho chúng ta nhiều vấn đề khó khăn.

Hơn nữa Linux có thể không chạy tốt với một số phần cứng ít phổ biến, thậm chí việc hỏng hóc hoặc xóa mất dữ liệu đôi khi cũng xảy ra, bởi vì Linux luôn thay đổi và khó được thử nghiệm đầy đủ trước khi đưa lên Internet.

Linux không phải là đồ chơi sẵn có, nó được thiết kế nhằm mang đến cho người sử dụng cảm giác cùng tham gia vào một dự án mới. Tuy nhiên thực tế cho thấy Linux chạy tương đối ổn định và cho chúng ta một cơ may không tốn kém để học và sử dụng UNIX, một họ hệ điều hành chuyên nghiệp hiện nay đang được rất nhiều người dùng trên các máy chủ và trạm tính toán cao cấp.

1.2. Tại sao Linux phát triển

Trước hết, Linux phát triển vì là một trong những hệ điều hành miễn phí và có khả năng đa nhiệm cho nhiều người sử dụng cùng lúc trên các máy tính tương thích với PC. So với những hệ điều hành thương phẩm, Linux giúp chúng ta ít phải nâng cấp và lại không cần trả tiền, cũng như phần lớn các phần mềm ứng dụng cho nó. Hơn nữa, Linux và những ứng dụng đó được cung cấp với cả mã nguồn miễn phí mà chúng ta có thể lấy về từ Internet, sau đó chỉnh sửa và mở rộng chức năng của chúng theo nhu cầu riêng.

Linux có khả năng thay thế một số hệ điều hành thuộc họ UNIX đắt tiền. Nếu tại nơi làm việc mà chúng ta sử dụng UNIX thì ở nhà chúng ta cũng thích sử dụng một hệ nào đó giống như thế nhưng rẻ tiền. Linux giúp chúng ta dễ dàng truy cập, lướt qua các Website và gửi nhận thông tin trên mạng Internet. Nếu chúng ta là một quản trị viên UNIX thì về nhà chúng ta cũng có thể sử dụng Linux để thực hiện mọi công việc quản trị hệ thống.

Một nguyên nhân khác làm cho Linux dễ đến với người dùng là nó cung cấp mã nguồn mở cho mọi người.

Chính điều này đã khiến một số tổ chức, cá nhân hay quốc gia đầu tư vào Linux nhằm mở rộng sự lựa chọn ra ngoài các phần mềm đóng kín mã nguồn. Họ cho rằng, mặc dù có dịch vụ hậu mãi nhưng không gì đảm bảo được rằng khi dùng các sản phẩm đóng kín này trên Internet, các thông tin cá nhân hay quốc gia của họ có bị gửi về một tổ chức hay một quốc gia nào khác hay không. Thí dụ Trung Quốc đã phát triển hệ điều hành Hồng Kỳ từ kernel của Linux để không bị lệ thuộc Microsoft Windows, cũng như họ đang tự nghiên cứu bộ vi xử lý Hồng Tâm để thay thế cho họ chip Intel.

Tại Việt Nam, việc nghiên cứu xây dựng một hệ điều hành từ kernel Linux đã thu được một số thành công nhất định. Chắc chúng ta cũng đã biết đến Vietkey Linux và CMC RedHat Linux (phiên bản tiếng Việt của RedHat Linux 6.2).

Gần đây, các công ty nổi tiếng như IBM, Sun, Intel, Oracle cũng bắt đầu nghiên cứu Linux và xây dựng các phần mềm ứng dụng cho nó.

1.3. Các bản phát hành Linux

Nhiều người đã biết đến các nhà sản xuất phần mềm RedHat, ManDrake, SuSE, Corel và Caldera. Có thể chính chúng ta cũng đã từng nghe đến tên các phiên bản Linux như Slackware, Debian, TurboLinux và VA Linux, v.v. Quả thật, Linux được phát hành bởi nhiều nhà sản xuất khác nhau, mỗi bản phát hành là một bộ chương trình chạy trên nhóm tệp lõi (kernel) của Linus Torvalds. Mỗi bản như vậy đều dựa trên một kernel nào đó, thí dụ bản RedHat Linux 6.2 sử dụng phiên bản kernel 2.2.4.

Hãng RedHat đã làm ra chương trình quản lý đóng gói RPM (RedHat Package Manager), một công cụ miễn phí giúp cho bất cứ ai cũng có thể tự đóng gói và phát hành một phiên bản Linux của chính mình. Thí dụ bản OpenLinux của Caldera cũng đã được tạo ra như thế.

1.4. Lợi thế của Linux

Tại sao có thể chọn Linux thay vì chọn một trong những hệ điều hành khác chạy trên PC như DOS, Windows 95/98, Windows NT, hoặc Windows 2000 ?

Linux cung cấp cho chúng ta một môi trường học lập trình mà hiện nay chưa có hệ nào sánh được. Với Linux, chúng ta có đầy đủ cả mã nguồn, trong khi đó các sản phẩm mang tính thương mại thường không bao giờ tiết lộ mã nguồn.

Cuối cùng, Linux mang đến cho chúng ta cơ hội sống lại bầu không khí của cuộc cách mạng vi tính trước kia. Cho đến giữa thập niên 1970, máy tính điện tử

còn là sân chơi riêng của các tổ chức lớn, chẳng hạn như chính quyền, tập đoàn doanh nghiệp và trường đại học. Người dân thường đã không thể sử dụng những thành tựu kỳ diệu của công nghệ thông tin.

Song với sự xuất hiện của bộ vi xử lý đầu tiên (1971) rồi máy tính cá nhân (1975), mọi việc đã thay đổi. Thoạt tiên, đó là đất dụng võ của các tay hacker say mê vi tính. Họ thậm chí có thể tự làm ra những máy tính cá nhân và hệ điều hành đơn giản, nhưng các hệ này chưa làm gì được nhiều ở góc độ hiệu năng. Với kinh nghiệm tích lũy dần theo năm tháng, một số hacker đã trở thành nhà doanh nghiệp, rồi cùng với khả năng tích hợp ngày càng cao của các vi mạch, PC đã trở thành phổ biến (rất tiếc hiện nay xã hội thường nghĩ xấu về chữ "hacker", xin mời chúng ta xem thêm mục "Ai phát triển Linux?" ở cuối chương này để phân biệt rõ hơn hacker và cracker là những ai).

Ngày nay Linux đang làm một cuộc cách mạng ở lĩnh vực phần mềm hệ thống. Linux là lá cờ tập hợp những con người không muốn bị kiểm sát bởi các hãng khổng lồ nhân danh kinh tế thị trường để làm xơ cứng óc sáng tạo và cải tiến.

Với Linux chúng ta sẽ khai thác được nhiều thế mạnh của UNIX. Trong số những hệ điều hành thông dụng hiện nay, Linux là hệ điều hành miễn phí được nhiều người sử dụng rộng rãi nhất. Bản thân Linux đã hỗ trợ sẵn sàng bộ giao thức mạng TCP/IP, giúp chúng ta dễ dàng kết nối Internet và gửi thư điện tử. Linux thường đi kèm XFree86 là một giao diện đồ họa cho người sử dụng (GUI) và cũng được phát hành miễn phí. XFree86 cung cấp cho chúng ta các chức năng phổ biến ở một số thương phẩm khác, chẳng hạn như Windows.

Tính khả chuyển của một hệ điều hành giúp chúng ta chuyển nó từ một nền này sang nền khác mà vẫn hoạt động tốt. Thí dụ UNIX là một hệ có tính khả chuyển cao. Ban đầu UNIX chỉ hoạt động trên một nền duy nhất, đó là máy tính mini DEC PDP-7.

Hiện nay UNIX và Linux có khả năng chạy trên bất kỳ nền nào, từ máy xách tay cho đến máy tính lớn. Nhờ tính khả chuyển, các máy tính chạy UNIX và Linux trên nhiều nền khác nhau có thể liên lạc với nhau một cách chính xác và hữu hiệu. Những hệ này có thể hoạt động mà không cần phải bổ sung thêm bất kỳ giao diện liên lạc đắt tiền nào, mà thông thường chúng ta phải mua thêm sau khi mua những hệ điều hành khác.

Linux đã có hàng ngàn ứng dụng, từ các chương trình bảng tính điện tử, quản trị cơ sở dữ liệu, xử lý văn bản đến các chương trình phát triển phần mềm cho nhiều ngôn ngữ, chưa kể nhiều phần mềm viễn thông trọn gói. Ngoài ra Linux cũng có hàng loạt trò chơi giải trí trên nền ký tự hoặc đồ họa. Phần lớn những chương trình tiện ích và ứng dụng có sẵn cho Linux lại không mất tiền mua. Các chúng ta chỉ phải trả chi phí cho việc tải chúng từ Internet xuống hoặc trả cước phí bưu điện.

Đến với Linux, giới lập trình sẽ có một loạt các công cụ phát triển chương trình, bao gồm các bộ biên dịch cho nhiều ngôn ngữ lập trình hàng đầu hiện nay, chẳng hạn như C, C++. Chúng ta cũng có thể dùng ngôn ngữ Pascal thông qua trình biên dịch FreePascal. Nếu chúng ta không thích sử dụng những ngôn ngữ vừa kể, Linux có sẵn các công cụ như Flex và Bison để chúng ta xây dựng ngôn ngữ riêng cho mình.

Hai khái niệm hiện nay được đề cập rất nhiều là hệ thống mở (open system) và tính liên tác (interoperability) đều gắn với khả năng của những hệ điều hành có thể liên lạc với nhau. Phần lớn các hệ mở đòi hỏi phải thỏa mãn tương thích tiêu chuẩn IEEE POSIX (giao diện hệ điều hành khả chuyển). Linux đáp ứng những tiêu chuẩn ấy và được lưu hành với mã nguồn mở.

1.5. Ai phát triển Linux

Nói chung, Linux là một hệ thống được xây dựng bởi các hacker và cho các hacker. Mặc dù hiện nay trong xã hội từ hacker thường có hàm ý tiêu cực, song nếu theo nghĩa ban đầu thì hacker không phải là tội phạm. Hacker tìm hiểu những gì có bên trong một hệ thống cho đến từng chi tiết và có khả năng sửa chữa nếu hệ thống ấy bị hỏng hóc. Đa số các hacker không xâm nhập hệ thống vì tiền bạc hoặc ác ý, mặc dù sau này đã có những người vượt qua giới hạn ấy và bị tập thể các hacker gọi là cracker (tin tặc) hay hacker mũ đen. Giới hacker cảm thấy bị xúc phạm khi mọi người xem họ như lũ phá hoại và gọi chung là tin tặc.

Thực ra, những hacker chân chính, còn gọi là hacker mũ trắng, rất có công trong việc phát hiện kẽ hở của các phần mềm, giúp mọi người và chủ nhân của những phần mềm ấy cảnh giác trước sự tấn công của giới tin tặc. Cũng nhờ công cuộc bảo vệ này mà Linux và các ứng dụng Linux (nói rộng hơn là các phần mềm nguồn mở) càng ngày càng an toàn hơn.

Ngoài đời, phần lớn những người sử dụng UNIX chỉ được cấp cho một số trương khoản với quyền hạn thu hẹp, do đó một người bình thường không thể thử nghiệm đầy đủ các câu lệnh UNIX. Với Linux chúng ta có một phiên bản hoạt động tương tự UNIX nhưng cho phép quản trị, sử dụng, vào ra thoải mái không giới hạn, một điều hiếm gặp trong cuộc sống. Linux cho chúng ta biết thế nào là làm hacker, song chúng tôi hy vọng từ đó chúng ta sẽ không trở thành cracker.

1.6. Linux cộng sinh với Windows

Về nguyên tắc, tất cả các phần mềm đang chạy trên DOS hoặc Windows sẽ không chạy trực tiếp với Linux, nhưng 3 hệ điều hành này có thể cộng sinh trên cùng một máy PC, dĩ nhiên mỗi lúc chỉ chạy được một hệ điều hành thôi. Chúng ta cũng có thể cài thêm một chương trình đặc biệt tên là "VMWARE" để phỏng tạo một hay nhiều hệ điều hành khác nhau chạy đồng thời trên cùng một máy với điều kiện máy của chúng ta phải có một cấu hình thích hợp và đủ mạnh.

Người ta còn xây dựng những chương trình phỏng tạo môi trường Windows và DOS trên nền Linux. Công ty Caldera đã chuyển WABI (Windows Applications

Binary Interface) của Sun sang Linux., cho phép các ứng dụng Windows 3.1 chạy với Linux. Caldera bán sản phẩm vừa kể trên và nhiều ứng dụng Linux song vẫn biểu không phiên bản RedHat để chạy các ứng dụng do hãng bán ra. Caldera còn thử chuyển một phiên bản DR DOS sang Linux.

Chương trình WINE cũng được sử dụng như một môi trường phỏng tạo Windows để có thể chạy các ứng dụng Windows trong Linux. Nói chung, Linux có khả năng chạy các ứng dụng Macintosh, DOS và Windows.

Ngược lại, cũng có nhiều người đang soạn thảo những chương trình phỏng tạo Linux trên nền Windows như đã từng có chương trình cho phép chạy các phần mềm Macintosh trên nền Sun và Windows. Chúng ta có thể xem các thông tin liên quan mới nhất trên các Web site về Linux.

Muốn cài đặt Linux chúng ta phải phân vùng lại ổ cứng máy mình, mặc dù không phải lúc nào cũng nhất thiết làm như thế. Chúng ta phải xoá một phần ổ cứng chứa chương trình và dữ liệu có sẵn trong đó. Hiện nay, việc cài đặt Linux mà không phân vùng lại ổ cứng đã được giải quyết nhưng khi chạy vẫn còn chậm. Do đó khi dự định cài đặt Linux chúng ta nên sao lưu ổ cứng ra vài ba bản.

Ổ cứng cũng cần phải còn đủ chỗ cho cả Linux và những hệ điều hành khác, chúng ta phải quyết định cái nào giữ lại và cái nào bỏ đi. Chúng ta có nhiều lựa chọn để phân vùng lại ổ cứng. Chẳng hạn chúng ta có thể dành chỗ riêng cho DOS và Linux, hoặc chúng ta chạy một chương trình phân vùng ổ cứng mà không phải xoá các tệp có sẵn. Tuy nhiên rủi ro mất dữ kiện khi cài đặt vẫn còn đó.

Khi phân vùng lại ổ cứng, chúng ta sẽ kiểm soát vùng đĩa dành riêng cho Linux hữu hiệu hơn, và Linux cũng chạy tốt hơn. Dung lượng đĩa cứng dành cho Linux sẽ tùy vào việc chúng ta muốn cài bao nhiêu ứng dụng và đó là phiên bản Linux nào. Chúng ta cần có ít nhất 300 MB trống trên ổ đĩa cứng nếu muốn cài RedHat 7.2, chưa kể đến tất cả các chương trình và dữ liệu mà chúng ta muốn giữ lại từ hệ điều hành trước đó. Nếu ổ cứng của chúng ta còn nhiều hơn thì càng tốt.

Chúng ta cần phải học cách quản lý hệ thống Linux để trở thành quản trị viên hệ thống (system administrator hoặc sys admin). Công việc của quản trị viên hệ thống bao gồm: thêm bớt trương khoản cho những người sử dụng, điều đặn sao lưu dữ liệu, cài đặt thêm phần mềm mới, thiết lập cấu hình hệ thống, và giải quyết các hỏng hóc. Linux càng ngày càng phổ biến vì thế nguồn tài liệu hiện nay rất phong phú. Phần lớn các bản phát hành Linux đều kèm theo hàng ngàn trang tài liệu. Có thể dễ dàng tìm thấy những thông tin tương tự tại thư mục /DOCS trên các CD chứa Linux.

1.7. Thương mại hoá Linux

Cũng như mọi phần mềm, Linux chưa thể khắc phục hết ngay những bất tiện và sai sót. Nhưng rõ ràng càng ngày càng có thêm công ty mới đầu tư cho Linux và đưa ra các giải pháp ít nhiều có tính thương mại với giá rất rẻ. Xin nêu tên hai trong số các công ty đó là RedHat và Caldera.

Cả hai công ty này đều trợ giúp kỹ thuật qua e-mail, fax và qua mạng cho những người đã mua các phiên bản Linux và sản phẩm của họ mà không dành cho những người sao chép các bản miễn phí.

Vì tính kinh tế, Linux và các chương trình kèm theo thường được chạy trên mạng nội bộ của nhiều doanh nghiệp, chẳng hạn làm các dịch vụ Web, tên miền (DNS), định tuyến (routing) và bức tường lửa. Nhiều nhà cung cấp dịch vụ Internet (ISP) cũng dùng Linux làm hệ điều hành chính.

Ngoài việc phân phối RedHat Linux với RPM, doanh nghiệp RedHat còn có những sản phẩm khác, thí dụ bộ ứng dụng văn phòng Applixware, bao gồm một phần mềm xử lý văn bản, một phần mềm bảng biểu, một phần mềm trình diễn, một công cụ thư điện tử cùng với nhiều công cụ triển khai lập trình và giao diện đồ họa XFree86... Nhưng chỉ cần trả khoảng một nửa giá bán của riêng Windows XP thôi, chúng ta sẽ nhận được một bản RedHat kèm các phần mềm nói trên mà không cần phải mua thêm MS Office, v.v.

Caldera lúc đầu chỉ phát hành từ mạng Internet các sản phẩm dựa trên RedHat và Novell, trước khi có OpenLinux, một hệ điều hành giá rẻ với kernel 2.x. Sản phẩm này bao gồm một giao diện đồ họa có khả năng quản lý hệ thống và tài nguyên mạng, cùng với các ứng dụng mạng chủ yếu. OpenLinux tích hợp một X server thương mại của MetroLink và một phiên bản trình duyệt đã đăng ký đầy đủ của Netscape Navigator. Hiện nay, Caldera tách riêng OpenLinux thành 2 sản phẩm khác nhau: một để dùng cho máy tính cá nhân và một để dùng cho máy chủ. Caldera cũng phát hành bản Corel WordPerfect cho Linux, cùng với một bộ ứng dụng văn phòng hướng Internet. Ngoài ra Caldera còn phát triển phần mềm tương thích công nghệ WABI của SunSoft, cho phép người dùng cuối chạy các ứng dụng Windows trên nền Linux.

2. Unix và Linux

Mục tiêu :

- Trình bày được lịch sử phát triển của hệ điều hành UNIX và Linux

Lịch sử Linux phát xuất từ UNIX và cụ thể liên quan đến Minix. Minix là một hệ điều hành nhỏ kiểu UNIX, minh họa bộ sách giáo khoa rất nổi tiếng do Tannebaum viết từ giữa những năm 1980. Minix đã từng phổ biến trên nhiều máy tính mini và PC. Còn Bell Laboratories thuộc công ty AT&T là nơi hệ điều hành UNIX sinh ra, song chính các tập thể và cá nhân khác đã cải thiện UNIX qua nhiều năm. Từ năm 1969, Thompson và các cộng sự ở Bell Laboratories đã phát triển UNIX, một hệ điều hành rất linh động và phù hợp với nhiều yêu cầu khác nhau của giới lập trình. Khởi thủy, hệ điều hành MULTICS của Viện MIT đã gợi ý cho Thompson viết được sản phẩm của mình, nhưng sau này chỉ có UNIX trở thành một tiêu chuẩn công nghiệp cho các hệ điều hành đa nhiệm và đa người dùng.

Năm 1978, Berkeley Software Distribution (BSD) thuộc Đại học Berkeley tại California đã phát triển phiên bản UNIX đầu tiên của mình từ nền phiên bản

UNIX v.7 của AT&T, với ý đồ sao cho UNIX trở nên thân thiện hơn với người sử dụng. Mặc dù không hoàn toàn tương thích với UNIX nguyên thủy của AT&T, phiên bản BSD UNIX vẫn đạt được mục tiêu đề ra nhờ những tiện ích mới đã làm nhiều người hài lòng.

Sau đó BSD đã phát hành FreeBSD, một phiên bản dành cho họ vi xử lý Intel 386 và phân phối khá hạn chế qua Internet hoặc CD-ROM, rồi các tác giả đã công bố bản này trên tạp chí Dr. Dobbs's. Hiện nay bản thương phẩm của FreeBSD đã trở thành một hệ điều hành thông dụng tương tự như Linux.

UNIX System Laboratories (USL) là một công ty ra đời từ AT&T và đã từng triển khai UNIX System V từ đầu thập niên 1980. Trước khi được Novell mua lại hồi năm 1993, USL sở hữu mã nguồn của tất cả các phiên bản xuất xứ từ UNIX System V. Tuy nhiên hồi ấy USL chưa bán ra được những bản sẵn sàng cho người tiêu dùng. Bản phát hành đáng nhớ nhất của USL là UNIX System V Release 4.2 (SVR4.2). Đây là lần đầu tiên mà USL tham gia vào thị trường với qui mô lớn. Lúc ấy Novell và USL khai trương một công ty liên doanh mang tên Univel để sản xuất hàng loạt phiên bản SVR4.2 gọi là UnixWare. Khi mua lại USL, Novell đã chuyển vai trò trọng tâm của USL từ nhà sản xuất mã nguồn thành nhà phát hành UnixWare. Cuối cùng Novell lại bán UNIX của mình cho công ty Santa Cruz Operation (SCO). Gần đây SCO phát hành bản SCO UNIX một người dùng (single-user), tuy nhiên chi phí lên đến 19 USD, khó cạnh tranh được với Linux đa người dùng. Hơn nữa SCO không công bố mã nguồn hệ điều hành của mình.

Từ cuối thập niên 1970, Microsoft cũng đã từng phát triển phiên bản UNIX của mình, gọi là XENIX. Đến năm 1981, trong thời kỳ cao điểm của cuộc cách mạng vi tính, máy tính cá nhân IBM-PC ra đời với hệ điều hành đơn nhiệm một người dùng DOS. Khả năng xử lý của PC tăng dần và bắt đầu sánh ngang các máy tính mini vào cuối thập niên 1980, khi sự ra đời của bộ vi xử lý Intel 386 cho phép XENIX có thể chạy trên PC. Microsoft và AT&T đã đồng ý nhập XENIX và UNIX vào thành một phiên bản duy nhất gọi là System V/386 Release 3.2, có khả năng hoạt động hầu như trên mọi cấu hình phần cứng của PC 386.

Sun Microsystems có đóng góp lớn lao vào việc mở rộng thị trường UNIX khi sản xuất ra các máy chủ và máy trạm chạy với hệ điều hành SunOS trên nền UNIX BSD. Cuối cùng BSD và SVR4 cũng đã hội tụ và tương thích với nhau.

IBM bước vào thế giới của UNIX bằng sản phẩm mang tên hệ điều hành AIX (Advanced Interactive Executive). Các công ty HP và Apple cũng phát triển phiên bản UNIX của mình, gọi là HP-UX và A/UX. Mặc dù AIX, HP-UX và A/UX không nổi tiếng bằng vài phiên bản UNIX khác, song chúng chạy rất tốt và có một thị phần đáng kể.

Các công ty nói trên đều giữ bản quyền phiên bản UNIX của mình, trong khi DOS và MS Windows thuộc về Microsoft. Vậy ai là chủ sở hữu của Linux?

3. Tác quyền và bản quyền Linux

Mục tiêu :

- Trình bày được sự tác quyền và bản quyền Linux

Nói chung, Linux không phải là phần mềm công cộng, bởi vì các thành tố của nó đã được nhiều người khác đăng ký tác quyền. Linus Torvalds giữ tác quyền về kernel Linux. Công ty RedHat là chủ của phiên bản RedHat Linux, và Patrick Voldkerding giữ tác quyền bản Slackware Linux v.v.

Nhưng nhiều tiện ích Linux lại có giấy phép công cộng GPL (GNU General Public License). Quả thực, Torvalds cùng nhiều người đóng góp cho Linux đã đặt công trình của mình dưới sự bảo vệ của GPL. Chúng ta có thể xem toàn văn GPL trên Internet hoặc trong tệp mang tên "copying" của mọi bản phát hành Linux. Bản quyền ấy đôi khi được gọi dí dỏm là Copyleft để đối lập chữ Copyright. GPL áp dụng cho phần mềm thuộc phong trào GNU (cũng chơi chữ: GNU's Not UNIX) và FSF (Free Software Foundation), cho phép tạo ra phần mềm tự do cho tất cả mọi người. Tự do hiểu là mỗi người đều có quyền sử dụng phần mềm GPL và tùy thích chỉnh sửa nó theo nhu cầu riêng của mình nhưng phải nhớ rằng không được giữ riêng bản chỉnh sửa ấy mà phải phổ biến rộng rãi để cho người khác cùng sử dụng và tiếp tục thay đổi theo ý họ.

GPL cho phép tác giả chương trình được giữ tác quyền pháp lý; song tác giả phải để cho người khác thao tác, thay đổi, và thậm chí bán chương trình mới được viết lại. Tuy nhiên một khi đã bán đi rồi thì người bán không được cấm người mua thay đổi chương trình đó và phải cung cấp mã nguồn. Đó là lý do tại sao Linux đến với chúng ta cùng toàn bộ mã nguồn đầy đủ và mở.

B. CÂU HỎI VÀ BÀI TẬP

Câu 1: Trình bày các giai đoạn phát triển và các phiên bản của hệ điều hành Linux?

Câu 2: Nêu sự khác biệt giữa Linux và Unix?

Câu 3: Trình bày sự tác quyền và bản quyền Linux?

BÀI 2: CHUẨN BỊ CÀI ĐẶT LINUX

Mã bài: MĐ37-02

❖ Giới thiệu

Chúng ta sẽ tìm thấy sau đây các thông tin cần thiết trước khi cài đặt bất cứ bản phát hành Linux nào trên PC. Xin nhớ rằng Linux không phải là một thương phẩm, do đó cần phải chuẩn bị đối phó các trục trặc nếu có. Chúng ta có thể đọc thêm các HOW-TO, ngoài những mục hướng dẫn khá đầy đủ trong chương này như:

- Chọn cấu hình phần cứng
- Dung lượng đĩa và bộ nhớ
- Những cách cài đặt Linux
- Phân vùng ổ đĩa cứng

❖ Mục tiêu

- Xác định cấu hình phần cứng phù hợp, sinh viên có khả năng từ nhu cầu thực tế để tính toán đưa ra cấu hình phần cứng thích hợp nhất từ bộ nhớ, dung lượng bộ nhớ, chuẩn thiết bị đầu cuối... để chuẩn bị cho bước cài đặt Linux tốt nhất.

- Nâng cao nhận thức về tính tương thích.

❖ Nội dung chính

A. LÝ THUYẾT

1. Chọn cấu hình phần cứng

Mục tiêu :

- Trình bày được cách chọn cấu hình phần cứng để cài đặt hệ điều hành Linux

Điều kiện cài đặt Linux thành công là có các phần cứng phù hợp. Muốn chọn cấu hình cho tương xứng, chúng ta phải biết trước bao nhiêu người sẽ sử dụng hệ thống và sẽ chạy những ứng dụng nào. Từ đó chúng ta tính ra các yêu cầu về bộ nhớ, dung lượng ổ đĩa cứng, chủng loại thiết bị đầu cuối, v.v.

Ngày nay, đa số các máy tính có cài đặt Linux đều là PC và thường cũng chỉ cài đặt phiên bản cho một người sử dụng, mặc dù các máy ấy có thể liên kết với nhiều hệ thống Linux và UNIX lớn hơn.

Nếu chúng ta cài đặt phiên bản Linux cho một người dùng (trường hợp hay gặp nhất) thì chúng ta cũng là quản trị viên của hệ thống. Chúng ta có trách nhiệm hiểu rõ hệ thống để thực hiện chức năng quản trị, sao cho hệ thống chạy tối ưu. Chúng ta phải bảo đảm dung lượng tối thiểu trên ổ đĩa cứng, sao lưu đều đặn, các thiết bị kết nối với hệ đều có trình điều khiển (driver) và các phần mềm cài đặt thích hợp, v.v..

Chúng ta nên chọn lựa các loại phần cứng mà chính đa số những người tạo ra Linux đã sử dụng. Các công ty phát triển phần mềm thương phẩm thường chạy thử sản phẩm của họ trên nhiều phần cứng khác nhau, còn cộng đồng tình nguyện triển khai Linux chỉ có máy tính của chính mình.

Cũng may là cộng đồng Linux khá đông đảo cho nên hầu hết những phần cứng tiêu chuẩn của PC đều được Linux chấp nhận.

Lưu ý: Linux là một hệ thống tiến hoá và thỉnh thoảng lại có thông tin cập nhật. Bản phát hành RedHat sử dụng trong tài liệu này chạy khá ổn định, tuy nhiên thực tế có những phần cứng thay đổi mà chưa được Linux biết đến. Mặc dù nhiều phần cứng có thể đã thay đổi bằng các linh kiện “nhái” hoặc tương thích Intel, song không phải tất cả những phần cứng ấy đều chạy được với Linux.

1.1. Bộ xử lý

Hệ thống phần cứng phù hợp Linux thường là một PC có bộ xử lý Intel 386 hoặc hiện đại hơn, chẳng hạn như 486, 586 hoặc Pentium. Những bộ xử lý nhái Intel như của Cyrix hoặc AMD cũng đều chấp nhận Linux.

Một số PC không có bộ đồng xử lý toán học, nhưng Linux không nhất thiết cần đến bộ phận này vì có thể phỏng tạo nó bằng cách sử dụng các chương trình con, dù rằng như thế sẽ giảm tốc độ thi hành.

Kernel Linux cũng được phát triển cho một số bộ xử lý khác, chẳng hạn như DEC Alpha, IBM PowerPC và Sun Sparc, thậm chí cho cả các bộ xử lý dùng trong hệ thống nhúng (embedded) như Network PC của Caldera.

1.2. Bus hệ thống

Linux thường chạy với các loại bus như ISA, EISA và PCI. Các kernel mới của Linux (từ 2.2 trở đi) có thể chạy với bus AGP. Với bus MCA trên máy tính PS/2 của IBM, chỉ các bản kernel từ 2.0.7 là chạy được. Một số hệ thống sử dụng loại bus cục bộ, gọi là VLB, để truy cập đĩa cứng và hiển thị màn hình nhanh hơn cũng được Linux chấp nhận.

1.3. Bộ nhớ

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

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

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

1.4. Đĩa cứng

Chúng ta có thể khởi động Linux từ một đĩa mềm. Mặc dù trên nguyên tắc vẫn có thể chạy Linux từ đĩa mềm, song thực tế không ai làm như thế.

Nếu sử dụng PC ở nhà, chúng ta cần có một ổ đĩa mềm loại 3.5" (1.44 MB), cho dù có thể chạy thẳng Linux từ CD với bản demo của SUSE Linux 7.3 hoặc KNOPIX

Để hệ thống chạy hiệu quả hơn chúng ta nên cài đặt Linux vào ổ đĩa cứng có giao diện IDE. Linux chấp nhận cả giao diện ESDI, nhưng đối với loại ổ đĩa IDE cải tiến (tức EIDE) thì chỉ có các bản kernel Linux từ 2.2.x trở lên mới tương thích hoàn toàn.

Linux chấp nhận giao diện SCSI với bìa điều khiển của các hãng Adaptec, Future Domain, Seagate, UltraStore, cũng như với các bộ thích nghi (adapter) trên bìa ProAudio Spectrum 16 và Western Digital.

1.4.1. Dung lượng ổ đĩa cứng

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

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

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

Thông thường lệnh DOS **fdisk** hoặc một vài thương phẩm khác cho phép chúng ta phân chia lại ổ đĩa cứng và Linux cũng có tiện ích tương tự gọi là FIPS.

Chú ý: Nếu chúng ta cài đặt Linux vào một ổ đĩa cứng mới nguyên thì không sao, còn đối với ổ đang dùng thì phải phân vùng và định dạng lại. Việc này sẽ xóa sạch toàn bộ thông tin trên ổ đĩa cứng, do đó chúng ta phải sao lưu cẩn thận trước khi cài đặt Linux. Nếu ổ đĩa cứng có dung lượng lớn, chúng ta có thể phân thành nhiều vùng và sao chép thông tin trở lại vào các vùng đã khai báo.

1.4.2. Phân vùng hoán chuyển

Như đã nói ở trên, nếu chúng ta có ít RAM thì phải cần đến phân vùng hoán chuyển (swap partition).

Một số hệ điều hành như Microsoft Windows lưu trữ tệp hoán chuyển trên ổ đĩa cứng như bất kỳ tệp nào khác, trong khi đó Linux cho phép tệp hoán chuyển cư trú trên một phân vùng dành riêng cho nó. Khi cài đặt, nhiều người sử dụng phân vùng hoán chuyển thay vì tệp hoán chuyển. Bởi vì có thể tạo ra nhiều phân vùng trên cùng một ổ đĩa cứng vật lý nên chúng ta có thể đặt phân vùng hoán chuyển trên cùng ổ đĩa với Linux. Tuy nhiên nếu đặt phân vùng hoán chuyển trên ổ đĩa khác, Linux sẽ chạy tốt hơn. Linux cho phép chúng ta tạo ra đến 8 phân vùng hoán chuyển. Nên đặt kích cỡ phân vùng hoán chuyển to gấp đôi số lượng RAM vật lý của máy chúng ta. Ví dụ máy chúng ta có 8 MB RAM thì phân vùng hoán chuyển nên là 16 MB.

1.5. Yêu cầu về màn hình

Đối với các thiết bị cuối làm việc ở chế độ văn bản (ASCII terminal), Linux chấp nhận tất cả mọi loại màn hình (video monitor) và bìa điều khiển màn hình (video adapter) hợp các chuẩn Hercules, CGA, EGA, VGA và SuperVGA.

Khi làm việc ở chế độ đồ họa, chúng ta cũng có thể chạy được bất kỳ tổ hợp màn hình và bìa điều khiển nào. Để tận dụng việc Linux có khả năng hiển thị đầy đủ các màu, chúng ta nên sử dụng màn hình màu.

Nhưng phiên toái có thể sinh ra khi chúng ta chạy XFree86 (phiên bản giao diện đồ họa X Window phát hành kèm theo Linux). Muốn chạy XFree86 chúng ta cần có một trong các bìa điều khiển màn hình liệt kê tại bảng sau.

Ghi chú: Bộ xử lý đồ họa là một nhóm các mạch tích hợp (chip, intergrated circuit) có chức năng lấy thông tin đầu ra từ máy vi tính và chuyển chúng thành một hình gồm những điểm sáng hiển thị trên màn hình. Muốn biết chính xác bìa điều khiển dùng bộ xử lý đồ họa nào, cần xem kỹ hồ sơ đi kèm với bìa đó.

Một số nhà sản xuất bìa điều khiển màn hình không cung cấp đủ thông tin cần thiết để lập trình trong XFree86, do đó khi chạy phần mềm này có thể các thông tin không được hiển thị trọn tru. Một vài hãng bằng lòng cung cấp thông tin nhưng đòi hỏi phải trả phần trăm quyền sở hữu, hoặc yêu cầu bảo mật.

Ghi chú: Trước đây các bìa điều khiển màn hình của hãng Diamond không chạy được với Linux vì những lý do liên quan đến quyền sở hữu. Hiện Diamond đã bắt đầu làm việc với nhóm XFree86 để tìm giải pháp tương thích với Linux.

1.6. Ổ CD

Muốn cài đặt Linux từ đĩa CD, máy chúng ta phải có ổ CD tương thích với Linux. Đa số các ổ CD trước kia sử dụng giao diện SCSI, do đó bất kỳ bộ điều khiển SCSI nào được liệt kê ở mục "Ổ đĩa cứng" nói trên đều được Linux chấp nhận. Hiện nay Linux tương thích với nhiều ổ CD loại mới, sử dụng giao diện EIDE và ATAPI đang có trên thị trường.

Nhiều ổ CD được bán theo dạng trọn gói multimedia có thể tương thích hay không tương thích với Linux, tùy vào việc bộ điều khiển có giao diện chuẩn SCSI thật hay chỉ là bộ thích nghi theo chuẩn riêng. Hầu hết các bộ thích nghi theo chuẩn riêng không hoạt động với Linux. Tuy nhiên Linux lại tương thích với

các ổ CD kiểu Creative Labs Soundblaster và cung cấp một cấu hình cài đặt riêng biệt cho các CD này. Sau đây là một số ổ CD tương thích với Linux:

1.6.1 Các ổ đĩa CD phổ quát

Các ổ đĩa có giao diện SCSI (xem tài liệu CD HOW-TO): Bất kỳ ổ SCSI CD với khối (block) 512 hay 2048 bytes đều có thể làm việc được trong Linux.

Các ổ đĩa có giao diện EIDE (ATAPI) CD và IDE CD: Hầu như tất cả các ổ đĩa CD tốc độ 2X, 4X, 6X đều được Linux hỗ trợ

Thí dụ các ổ đĩa CD phổ quát : Mitsumi FX400, Nec-260, Sony 55E v.v.

1.6.2. Các ổ đĩa CD đặc chủng

Aztech CDA268-01A, Orchid CDS-3110, Okano/Wearnes CDD-110

Conrad TXC, CyCDROM CR520ie/CR540ie/CR940ie (AZTCD)

Creative Labs CD-200(F) (SBPCD)

Funai E2550UA/MK4015 (SBPCD)

GoldStar R420 (GSCD)

IBM External ISA (SBPCD)

Kotobuki (SBPCD)

Lasermate CR328A (OPTCD)

LMS Philips C MB 206 (CM206)

Longshine LCS-7260 (SBPCD)

Matsushita/Panasonic CR-521/522/523/562/563/(SBPCD)

MicroSolutions Backpack parallel portdrive (BPCD)

Mitsumi CR DC LU05S (MCD/MCDX)

Mitsumi FX001D/F (MCD/MCDX)

Optics Storage Dolphin 8000AT (OPTCD)

Sanyo CDR-H94A (SJCD)

Sony CDU31A/CDU33A (CDU31A)

Sony CDU-510/CDU-515 (SOMYCD535)

Sony CDU-535/CDU-531 (SONYCD535)

Teac CD-55A SuperQuad (SBPCD)

1.7. Truy cập mạng

Hiện nay có thể kết nối các hệ thống Linux bằng nhiều cách, song hai cách phổ biến nhất và có sẵn nhiều thiết bị nhất là sử dụng giao diện mạng hoặc modem.

Các giao diện mạng bao gồm Token Ring, FDDI, ATM và Ethernet. Hầu hết các mạng thông thường đều sử dụng giao diện Ethernet.

1.7.1. Truy cập qua Ethernet

Khởi thủy do Xerox, DEC và Intel đề ra, Ethernet từ lâu đã trở thành giao diện mạng phổ biến nhất. Mặc dù ở nhà ít ai dùng máy Linux để kết nối vào mạng Ethernet, song ở các cơ quan, doanh nghiệp và trường học thì đó là điều thường thấy.

Bảng sau đây liệt kê những bìa giao diện mạng Ethernet được Linux chấp nhận:

3Com 3c501, 3c503, 3c505, 3c507
 3Com 3c509/3c509B (ISA)
 3Com 3c579 (EISA)
 3Com Etherlink III Vortex Ethercards 3c590, 3c592, 3c595, 3c597 (PCI)
 3Com Etherlink XL Boomerang 3c900, 3c905 (PCI)
 3Com Cyclone 3c905B, 3c980
 3Com Fast EtherLink Ethercard 3c515 (ISA)
 3Com 3ccfe575 Cyclone Cardbus (3c59x driver, PCMCIA)
 3Com 3c575 series Cardbus (3c59x driver, PCMCIA)
 AMD LANCE 79C960/PCnet (ISA/PCI)
 AT&T GIS WaveLAN
 Allied Telesis AT1700
 Allied Telesis LA100PCI-T
 Allied Telesyn AT2400T/BT ("ne" module)
 Ansel Communications AC3200 (EISA)
 Apricot Xen-II/82596
 Cabletron E21xx
 Cogent EM110
 Crystal Lan CS8920, Cs8900
 Danpex EN-9400
 DEC DE425 (EISA) / DE434/DE435 (PCI) / DE450/DE500 (DE4x5 driver)
 DEC DE450/DE500-XA (dc21x4x) (Tulip driver)
 DEC DEPCA và EtherWORKS
 DEC EtherWORKS 3 (DE203, DE204, DE205)
 DEC QSilver's (Tulip driver)
 Digi International RightSwitch
 DLink DE-220P, DE-528CT, DE-530+, DFE-500TX, DFE-530TX
 Fujitsu FMV-181/182/183/184
 HP PCLAN (27245 và 27xxx series)
 HP PCLAN PLUS (27247B và 27252A)
 HP 10/100VG PCLAN (J2577, J2573, 27248B, J2585) (ISA/EISA/PCI)
 ICL EtherTeam 16i/32 (EISA)
 Intel EtherExpress
 Intel EtherExpress Pro
 KTI ET16/P-D2, ET16/P-DC ISA
 Macromate MN-220P (PnP hoặc NE2000 mode)
 NCR WaveLAN
 Novell NE2000/NE1000
 Netgear FA-310TX (Tulip)

New Media Ethernet
 PureData PDUC8028, PCI8023
 SEEQ 8005
 SMC Ultra/EtherEZ (ISA)
 SMC 9000 series
 SMC PCI EtherPower 10/100 (Tulip driver)
 SMC EtherPower II (epic100.c driver)
 Sun LANCE adapters (kernel 2.2 và mới hơn)
 Sun Intel adapters (kernel 2.2 và mới hơn)
 Schneider & Koch G16
 Western Digital WD80x3
 Zenith Z-Note / IBM ThinkPad 300 built-in adapter
 Znyx 312 EtherArray (Tulip driver)

1.7.2. Truy cập qua modem

Khi làm việc ở nhà, chúng ta thường kết nối với bên ngoài qua modem và mạng điện thoại bằng các giao thức liên lạc nối tiếp như SLIP hoặc PPP.

Nói chung, Linux tương thích với mọi loại modem sử dụng cổng Serial RS-232 đang bán trên thị trường. Đa số các loại modem nối qua cổng USB và modem gắn trong (internal modem) các PC cũng chạy được dưới Linux.

Trong một số trường hợp, chúng ta không có trình điều khiển riêng cho modem chạy với Linux và sẽ phải sử dụng trình điều khiển phổ quát (generic driver).

Nếu sử dụng được một modem dưới DOS thì chúng ta sẽ không có khó khăn gì cho nó chạy dưới Linux.

Dưới đây là danh sách các modem tương thích Linux hiện nay:

3Com 3CXM256/3CCM256 và 3CXM656/3CCM656 PCMCIA
 AOpen FM56-P và FM56-H
 AT&T/Lucent winmodem
 Boca Research 28.8 internal modem (model MV34AI)
 Boca Research 33.6 internal modem (model MV34)
 MC2920A-3.3, E6030D 4035-01 và 1721 8011 A
 Cirrus Logic CL-MD3450D-SC-B
 Cirrus Logic MD1724-11VC-D
 Datatronic VLM301-1
 Omron G5V-1
 AST M628032-20E1
 Cirrus Logic CD-MD4450C-SC-A
 Abracon 23-040-20
 Compaq 192PCMCIA modem/serial card
 HP Fastmodem D4810B
 IBM Mwave ("Dolphin" card

Multiwave Innovation CommWare V.34 modem
 Megahertz XJ/CC2560 PCMCIA
 New Media Winsurfer PCMCIA modem/serial card
 Rockwell SoftK56
 US Robotics WinModem Series
 Zoltrix 33.6 Win HSP Voice/Speaker Phone modem
 Zoltrix Phantom 56K, model FM-HSP56PCI, bộ xử lý PCTel (PCI)

1.8. Các thiết bị khác

Những mục sau đây sẽ nói về tính tương thích với Linux của các thiết bị ngoại vi khác như chuột, ổ băng từ, máy in. Các thiết bị này giúp chúng ta sử dụng Linux tiện lợi hơn, song không nhất thiết phải có.

1.8.1. Chuột

Chuột máy tính là thiết bị dùng để điều khiển con chạy (cursor) trên màn hình.

Nếu chỉ chạy ở chế độ văn bản thì chẳng cần đến chuột, mặc dù Linux cho phép chúng ta dùng chuột cắt những đoạn chữ từ màn hình rồi dán sang dòng lệnh, trong khi UNIX thường không làm được như vậy.

Muốn chạy ở chế độ đồ họa dưới giao diện X Window thì chúng ta phải sử dụng chuột. Linux tương thích với hầu hết các loại chuột nối tiếp của những hãng như Logitech, Kensington, Mouseman, Microsoft, v.v. Linux cũng chấp nhận các loại chuột bus của Microsoft, Logitech, ATIXL và IBM, v.v.

Những thiết bị khác dùng để điều khiển con chạy như quả cầu (trackball) và màn hình xúc giác (touch screen), mô phỏng các loại chuột vừa liệt kê, cũng đều chạy được với Linux.

1.8.2. Ổ băng từ

Ổ băng từ có dung lượng lớn nên cho phép thoải mái sao lưu dữ liệu hệ thống. Linux tương thích với nhiều ổ băng từ có giao diện SCSI được liệt kê ở bảng dưới đây. Linux cũng chấp nhận các ổ băng từ rẻ tiền như Colorado Memory Systems loại 120 MB và 250 MB, được cắm thẳng vào giao diện điều khiển ổ đĩa. Các ổ băng từ cắm thẳng vào cổng song song (cổng máy in) hiện chưa được Linux chấp nhận. Hầu hết các ổ băng từ tương thích chuẩn QIC-02 chạy được với Linux.

Sau đây là một số ổ băng từ tương thích với Linux:

Hãng sản xuất	Kiểu
Exabyte	Tất cả các kiểu có giao diện SCSI
Sanko	CP150SE
Tandberg	3600
Wangtek	552ES, 515ES, 5099EN

1.8.3. Máy in

Linux tương thích với toàn bộ các máy in nối qua cổng song song. Với máy in song song, thì khó khăn lớn nhất có thể gặp là hiệu ứng nấc thang. Hiệu ứng

nấc thang xảy ra do cách UNIX và Linux xử lý khi xuống dòng và điều khiển đầu in quay về đầu dòng. Với UNIX, lệnh đưa giấy lên thêm một dòng (LF: line feed) sau đó đặt đầu in tại vị trí đầu dòng mới (CR: carriage return) do một ký tự duy nhất là LF điều khiển, trong khi các hệ như DOS hoặc Windows lại sử dụng cặp ký tự CR-LF cho hai lệnh trên. Khi chúng ta in một tệp UNIX bằng máy in được cấu hình cho DOS, chúng ta sẽ bị hiệu ứng nấc thang bởi vì tệp chỉ chứa ký tự LF chứ không chứa ký tự CR.

Ở chiều ngược lại, các tệp văn bản soạn trong môi trường DOS/Windows cũng cần được chỉnh lý (cặp ký tự CL-LF đổi thành LF) khi chuyển sang môi trường Linux hoặc UNIX.

Việc đặt cấu hình Linux để chạy với máy in nối tiếp thường khó hơn. Các chương trình cài đặt Linux về cơ bản không có sẵn công cụ hỗ trợ máy in nối tiếp.

2. Dung lượng đĩa và bộ nhớ

Mục tiêu :

- Trình bày được cách chọn đĩa cứng và bộ nhớ để cài đặt

Tùy theo cách cài đặt, mở đầu chúng ta có thể cần ổ đĩa cứng khoảng 80G đã được định dạng để tạo ra đĩa môi cho Linux. Sau đó chúng ta phải dành đủ dung lượng đĩa cứng; nếu cài hết mọi thứ có trên CD thì phải cần 3.5 GB, tuy nhiên vẫn có thể sử dụng ít hơn. Cần tính xem có bao nhiêu trường khoản người dùng. Nếu hệ thống chỉ có một trường khoản thì 80 MB là đủ cho RedHat 6.1, nếu không cài X Window. Tuy nhiên, càng ngày yêu cầu về dung lượng đĩa càng tăng. Bản RedHat 7.2 cài tối thiểu cũng cần tới 350 MB đĩa cứng.

Tiếp theo chúng ta tính xem cần bao nhiêu cho vùng hoán chuyển (swap space) mà thường là khoảng 500 MB. Nhưng nếu chúng ta phải tạo một máy chủ cơ sở dữ liệu (database server) thì dung lượng này là không đủ. Đặc biệt với ORACLE 9i, chúng ta cần vùng hoán chuyển lớn gấp 2 hay 3 lần bộ nhớ vật lý (mà bộ nhớ vật lý tối thiểu cho ORACLE 9i đã là 256 MB).

Cuối cùng chúng ta nên chừa ít nhất 1 GB cho thư mục gốc (root). Đây là thư mục chính mà từ đó chúng ta truy cập đến tất cả các thư mục thứ cấp của Linux.

Cần nhắc lại là cách cài đặt Linux tối thiểu sẽ chỉ cần 350 MB (bản RedHat 7.2), trong khi cài đặt đầy đủ và dành chỗ cho nhiều user sẽ cần khoảng 5 GB.

Ghi chú: Chúng ta có thể chạy một phần hệ thống Linux từ CD mà không nhất thiết phải cài đặt toàn bộ lên đĩa cứng, song như thế không đọc được các CD khác.

Nếu chúng ta quyết định dùng giao diện đồ họa X Window, trước tiên nên ghi ra giấy xem bìa đồ họa của chúng ta được bộ xử lý nào điều khiển. Nếu chúng ta có chuột nối tiếp và modem, cũng nên ghi rõ tên cổng nối tiếp của chúng. Trong quá trình cài đặt chúng ta sẽ cần những thông số ấy.

3. Phân vùng ổ đĩa cứng

Mục tiêu :

- *Biết được cách phân vùng ổ đĩa cứng*
- *Sử dụng được các lệnh để phân vùng*
- *Thực hiện thành thạo phân vùng ổ đĩa cứng*

Sau khi đã sao lưu dữ liệu và tạo ra đầy đủ các đĩa môi và đĩa phụ trợ, hãy chuẩn bị ổ đĩa cứng cho Linux.

Lưu ý: Đây là một thao tác nguy hiểm nhất bởi vì sẽ xóa mất dữ liệu cũ. Nếu chưa sao lưu hệ thống, chúng ta nên làm công việc này trước khi cài đặt hệ điều hành mới vào máy. Mặc dù có thể sử dụng chương trình thử nghiệm FIPS và các chương trình thương mại như Partition Magic để phân chia lại ổ đĩa cứng mà không phá hủy dữ liệu, chúng tôi vẫn khuyên chúng ta sao lưu toàn bộ và đầy đủ.

3.1. Tìm hiểu về phân vùng

Khi PC xuất hiện, phần lớn các hệ điều hành, chương trình và dữ liệu đều để trên đĩa mềm. Khi máy PC XT ra đời, hãng IBM mới có thêm ổ đĩa cứng 10 MB. Các hệ điều hành sơ khai như DOS chỉ truy cập được một dung lượng rất hạn chế trên ổ đĩa cứng. Sau đó các nhà sản xuất hàng năm đều tăng dung lượng của ổ đĩa cứng khiến cho DOS khó theo kịp để quản lý những dung lượng mới lớn hơn. DOS tránh né vấn đề bằng cách cho phép người sử dụng chia ổ đĩa cứng thành nhiều vùng logic, gọi là phân vùng. Các phân vùng này lưu giữ được những tệp chương trình, các hệ điều hành khác, hoặc dữ liệu. DOS thường được khởi động tại một ổ đĩa cứng gọi là ổ C. Nếu chia ổ này làm ba thì các phân vùng ổ đĩa logic sẽ gọi là C, D, E. DOS cho phép lắp nhiều ổ đĩa (ổ cứng hoặc CD), do đó nếu lắp thêm một ổ đĩa nữa, nó sẽ được gọi là F, v.v.

UNIX và Linux không dùng các chữ cái ấy để gọi phân vùng, mà dùng cách khác. Người sử dụng Linux có thể đặt nhiều thư mục khác nhau trên những phân vùng khác nhau (của cùng một ổ đĩa cứng) và ngay cả trên các ổ đĩa cứng khác. Chúng ta cũng có thể đặt các hệ điều hành khác nhau trên các phân vùng khác nhau.

Thông tin phân vùng được ghi rõ trên cung đầu tiên của ổ đĩa cứng gọi là Master boot record (MBR) và mang tên là bảng phân vùng. Bảng này được sử dụng để biết xem phải khởi động hệ điều hành ở phân vùng nào. Chức năng của MBR là môi (boot), nghĩa là để móc nối với cơ chế khởi động hệ điều hành. Chương trình môi LILO của Linux và các phần mềm quản lý môi khác đều sử dụng MBR để biết xem nên khởi động hệ điều hành nào.

Bảng phân vùng ghi rõ vị trí và kích thước của nhiều phân vùng trên ổ đĩa cứng. Có ba loại phân vùng: sơ cấp (Primary), mở rộng (Extended) và logic (Logical). DOS và vài hệ điều hành khác phải khởi động từ phân vùng sơ cấp. Chỉ có thể tạo tối đa 4 phân vùng sơ cấp trên một ổ đĩa cứng. Bản thân một phân vùng mở rộng không chứa dữ liệu mà chỉ ghi lại cách phân vùng cho các phân

vùng khác trên ổ đĩa cứng. Số phân vùng logic trên một ổ đĩa cứng là không giới hạn. Do đó để giải quyết giới hạn của bốn phân vùng sơ cấp, chúng ta có thể chỉ định một phân vùng mở rộng và khai báo một số phân vùng logic khác ở bên trong phân vùng mở rộng.

DOS và các phiên bản của OS/2 trước phiên bản 2.0 đòi hỏi phải được cài đặt trên một phân vùng sơ cấp, tuy nhiên chúng có thể truy cập các ổ logic bên trong các phân vùng mở rộng. Việc này rất quan trọng nếu chúng ta muốn cài DOS và Linux trên cùng ổ đĩa cứng. DOS phải nằm trên phân vùng sơ cấp.

3.2. Sử dụng lệnh FDISK

Trên PC các phân vùng được tạo ra, xoá đi và quản lý bởi một chương trình gọi là **fdisk**. Mỗi hệ điều hành có **fdisk** riêng của mình, do đó trước khi sử dụng chúng ta phải dò lại xem có đúng phiên bản hay chưa. Nếu đang sử dụng DOS hoặc dự định sử dụng nó, trước tiên chúng ta phải phân chia lại ổ đĩa cứng bằng **fdisk** của DOS. Sau này chúng ta sẽ dùng **fdisk** của Linux để tạo phân vùng của Linux.

3.2.1. Các yêu cầu về phân vùng

Trước tiên chúng ta phải xác định mình cần bao nhiêu phân vùng. Trong khi DOS cần phân vùng sơ cấp thì Linux có thể cư trú trên các phân vùng khác. Nên nhớ nếu chúng ta nén một phân vùng hiện hành của DOS để dành chỗ cho Linux, thì tất cả các tệp của chúng ta không thể phục hồi hết trên phân vùng DOS nhỏ hơn mới được tạo.

Sau đó chúng ta quyết định số phân vùng cần thiết và mỗi phân vùng như thế cần bao nhiêu dung lượng ổ đĩa cứng.

Ghi chú: Từ Linux, chúng ta có thể vào các phân vùng của DOS và thực hiện các lệnh di dời, sao lưu, chỉnh sửa các tệp DOS, nhưng không thể chạy các chương trình DOS bằng Linux.

Hai phần mềm Linux cho phép chúng ta phỏng tạo DOS trên Linux và cài đặt Linux trên DOS. Cả hai hệ này chủ yếu thích hợp cho các hacker. Một trong những phần mềm ấy, gọi là UMSDOS, lại không tương thích với RedHat Linux.

3.2.2. Các yêu cầu về DOS

Nếu chúng ta khởi động bằng DOS, máy sẽ vào một phân vùng sơ cấp. Một phiên bản mới được (bootable) của DOS không đòi hỏi nhiều chỗ trên ổ đĩa cứng, chỉ cần đủ chỗ cho các tệp hệ thống COMMAND.COM, CONFIG.SYS và những driver cần thiết để khởi động hệ thống. Thực tế chỉ cần 5 MB cho phân vùng sơ cấp để khởi động DOS. Một khi đã tải xong và chạy DOS, chúng ta có thể vào bất cứ phân vùng mở rộng và phân vùng logic nào của hệ thống.

Nhưng trong khi Linux có thể truy cập các tệp DOS trong một phân vùng DOS thì trái lại DOS lại không thể truy cập các tệp Linux trong phân vùng Linux.

3.2.3. Các yêu cầu về Linux

Như đã nói, Linux thao tác trên các hệ thống tệp và chúng có thể trú trên nhiều phân vùng khác nhau, chủ yếu là để phòng xa. Linux đòi hỏi một phân vùng

cho mỗi hệ thống tệp. Việc tiếp theo phải quan tâm là phân vùng hoán chuyển. Phần lớn các hệ điều hành đều cho phép tạo bộ nhớ ảo, Linux cũng lấy một phần ổ đĩa cứng làm tệp hoán chuyển hoặc phân vùng hoán chuyển để mô phỏng bộ nhớ vật lý. Kích thước phân vùng hoán chuyển tùy thuộc số lượng RAM vật lý của hệ thống máy. Một thông lệ được chấp nhận mặc nhiên là: phân vùng hoán chuyển lớn gấp đôi lượng RAM. Do đó nếu máy chúng ta có 8 MB RAM, phân vùng hoán chuyển phải là 16 MB. Nếu có từ 4 MB RAM trở xuống, chúng ta phải kích hoạt một phân vùng hoán chuyển.

Đối với RedHat Linux phiên bản 6.x trở lên, tổng dung lượng dành cho các phân vùng hoán chuyển trên Linux là tùy ý (tối thiểu là gấp đôi dung lượng bộ nhớ vật lý), do đó chúng ta chỉ cần tính đến các yêu cầu của những ứng dụng sẽ được cài trên máy để xác định dung lượng cần cho phân vùng hoán chuyển. Ví dụ như nếu chúng ta cần cài thêm ORACLE 7.x trên máy như một máy chủ cơ sở dữ liệu thì ít nhất phần phân vùng hoán chuyển phải có dung lượng là 500 MB. Vì thế nếu ngoài một phân vùng hoán chuyển mà hệ thống Linux của chúng ta lại cần hai phân vùng khác (một cho tệp hệ thống và một cho tệp người dùng) thì tổng cộng chúng ta phải phân 3 vùng cho Linux.

3.2.4. Phân vùng lại ổ DOS

Trước tiên chúng ta phải thi hành FDISK bằng cách gõ **fdisk** tại dấu nhắc DOS. Một menu gồm 4 tùy chọn FDISK sẽ hiện ra trên màn hình.

Qua các tùy chọn, chúng ta biết phân vùng nào hiện tồn tại, biết tạo phân vùng mới và xoá phân vùng cũ. Tùy vào phiên bản DOS mà chúng ta đang sử dụng, màn hình sẽ hơi khác nhau một chút.

Chọn “Display Partition Information” trên menu. Khi màn hình Display Partition Information xuất hiện, chúng ta nên chép lại các thông tin. Chúng ta sẽ cần những thông tin này nếu quyết định ngưng lại quá trình cài đặt Linux và phục hồi hệ thống nguyên thủy trên máy chúng ta.

Trong DOS 6.x, nếu muốn xem tất cả các thông tin về phân vùng hiện hữu, chúng ta cũng sử dụng tùy chọn Display Partition Information.

3.2.5. Cách tránh phân vùng đĩa cứng

Mặc dù việc phân vùng lại ổ đĩa cứng sẽ giúp Linux chạy tốt hơn, song không nhất thiết phải thực hiện như trên vì e mất dữ liệu. Có thể dùng FIPS để phân vùng mà không phá huỷ các thông tin trên ổ đĩa cứng.

FIPS (First non-destructive Interactive Partition Splitting) là một chương trình phát triển cho Linux. Như tên gọi, FIPS sẽ di chuyển các phân vùng DOS để dọn chỗ cho các phân vùng Linux mà không phá hỏng thông tin.

Muốn biết thêm, hãy tham khảo tệp `fips.doc` ở thư mục `/utils/fips` trong CD Linux. FIPS chỉ có ích khi nào ổ đĩa cứng trong máy chúng ta còn đủ khoảng trống cần thiết để cài đặt Linux, nếu không chúng ta phải xoá những tệp nào xét thấy không cần thiết.

Với bản phát hành Slackware Linux, chúng ta có thể cài đặt Linux trên cùng phân vùng với DOS (nhưng lúc này gọi là UMSDOS). UMSDOS là một dự án nhằm tạo điều kiện cho Linux hiện diện trên các phân vùng DOS. Nói cách khác, UMSDOS cho phép chúng ta tạo hệ thống tệp gốc của Linux trong một thư mục DOS đã có sẵn. Tuy nhiên chúng ta không thể sử dụng UMSDOS với RedHat Linux.

3.2.6. Xoá bỏ phân vùng

Rất tiếc là **fdisk** không cho phép chúng ta đặt lại kích thước của phân vùng một cách đơn giản. Trước tiên chúng ta phải xoá bỏ phân vùng ấy, sau đó lại tạo ra chính nó nhưng với kích thước mới. Từ màn hình tùy chọn **fdisk**, chọn tùy chọn 3, “Delete Partition” hoặc “Logical DOS Drive” để xoá phân vùng được chọn. Màn hình Delete Partition hoặc Logical DOS Drive sẽ hiện ra.

Chọn tùy chọn tương ứng với loại phân vùng chúng ta sẽ muốn xoá, thí dụ phân vùng DOS sơ cấp. Chẳng hạn tùy chọn 1 (Delete Primary DOS Partition) giúp chúng ta xoá các phân vùng sơ cấp của DOS. Chọn tùy chọn 1 để hiển thị màn hình Delete Primary DOS Partition. Màn hình sẽ hỏi tên (volume) của phân vùng và đòi hỏi xác nhận lại lần nữa trước khi chúng ta quyết định xoá hẳn phân vùng ấy, cùng với tất cả mọi thông tin trên đó.

3.2.7. Thêm phân vùng mới

Sau khi xoá các phân vùng cần xóa, chúng ta phải thêm các phân vùng thích hợp cho hệ thống DOS của chúng ta bằng tùy chọn “Create a DOS Partition”.

Hãy chọn các tùy chọn sau để được thấy hỏi đáp màn hình cho “Create a DOS Partition” hoặc “Logical DOS Drive”.

Ghi chú: Chúng ta không thể thêm phân vùng Linux vào bằng chương trình FDISK của DOS. Việc phân chia lại ổ đĩa cứng cho Linux sẽ được bàn sau ở mục “Sử dụng lệnh **fdisk** của Linux”.

Các hỏi đáp màn hình của **fdisk** bao gồm thông số khoảng trống cho phân vùng (tính bằng MB) và chỉ báo về phân vùng hiện hành (sáng rõ). Phân vùng hiện hành là phân vùng có thể khởi động được.

Để khởi động DOS, chúng ta phải chỉ định phân vùng sơ cấp là phân vùng hiện hành. Với sự chọn lựa đầu tiên như trên màn hình này, chúng ta nên chọn N (no) để sau đó có thể ấn định dung lượng dành cho phân vùng DOS.

Nếu chúng ta chọn “no”, màn hình sẽ hiển thị “Specify Disk Space for the Partition Screen”.

Chúng ta có thể chọn dung lượng dành cho phân vùng DOS bằng đơn vị MB hoặc bằng tỷ lệ phần trăm và bấm <Return>. Sau đó chúng ta chỉ định đây là phân vùng hiện hành. Từ màn hình menu của **fdisk**, chúng ta chọn tùy chọn 2, “Set Active Partition”, sau đó làm theo các hướng dẫn.

3.2.8. Định dạng phân vùng

Sau khi phân vùng lại ổ đĩa cứng, chúng ta phải chuẩn bị phân vùng mới cho DOS và phục hồi các tệp thích hợp vào phân vùng ấy. Dùng đĩa mềm đã tạo sẵn

trước đó để khởi động lại máy. Tiếp theo chúng ta định dạng (format) ổ đĩa cứng thích hợp và chuyển các tệp hệ thống lên đó bằng lệnh DOS:

format c: /s

Khi phân vùng đã được định dạng xong, chúng ta có thể phục hồi phần sao lưu vào ổ mới. Nếu do giảm kích thước của phân vùng mà hết chỗ để phục hồi, chúng ta nên chuyển các tệp dư thừa vào các ổ DOS hoặc phân vùng DOS khác.

B. CÂU HỎI VÀ BÀI TẬP

Câu 1: Trình bày yêu cầu phần cứng phù hợp để cài đặt Linux?

Câu 2: Muốn sử dụng Linux cho có hiệu quả, chúng ta phải phân vùng lại ổ đĩa cứng như thế nào?

Câu 3: Trình bày các yêu cầu về phân vùng của DOS và Linux?

Câu 4: Nêu cách tránh phân vùng đĩa cứng?

BÀI 3: CÀI ĐẶT REDHAT LINUX

Mã bài: MD37-03

❖ Mục tiêu

- Cài đặt được bản phát hành Redhat Linux với những thông tin chi tiết trong các chủ đề như:

- + Các cách cài đặt.
- + Trình tự cài đặt.
- + Thiết lập cấu hình mạng.
- + Các thiết lập khác.
- + Tạo đĩa mềm khởi động.
- + Nâng cấp và gỡ bỏ RedHat.

- Tính cẩn thận, quyết đoán trong cài đặt hệ thống.

❖ Nội dung chính

A. LÝ THUYẾT

1. Các cách cài đặt & trình tự cài đặt

Mục tiêu :

- Trình bày được cách cài đặt và trình tự cài đặt hệ điều hành Redhat Linux

1.1. Các cách cài đặt

Như đã trình bày trong chương trước, có nhiều cách thực hiện việc cài đặt Linux: qua CD, qua đĩa cứng và qua mạng (FTP, NFS, SMB). Bảng 3.1 sau đây tóm tắt các điểm mạnh yếu của từng phương pháp trên.

Cài qua	Điểm mạnh	Điểm yếu
CD	Nhanh và Ổn định	Bản phát hành sớm lỗi thời
FTP	Tiện lợi cập nhật, có thể truy cập từ mọi nơi trên thế giới	Không Ổn định và chậm nếu máy chủ nằm xa máy trạm
NFS	Tiện lợi, đặc biệt trong trường hợp máy chúng ta không có ổ CD	Chậm và đòi hỏi phải có mạng UNIX
SMB	Tiện lợi trên mạng MS Windows	Phải qua mạng và phải có hiểu biết về Samba
Đĩa cứng	Khi tất cả các phương pháp kể trên đều không áp dụng được	Đòi hỏi nhiều dung lượng phụ trội

Bảng 3.1 : Tóm tắt các cách cài đặt Linux

Nếu định cài đặt cả giao diện X Window, chúng ta cần chuẩn bị trước các thông tin về loại chipset của bìa điều khiển màn hình, cũng như về cổng nối tiếp cho chuột và modem.

Ở đây chúng ta chỉ bàn đến việc cài đặt RedHat, bản phát hành phổ biến nhất của Linux. Mặc dù chúng ta có thể cài đặt bằng nhiều cách khác nhau như

qua NFS, qua FTP, qua ảnh SMB trên một ổ đĩa cứng dùng chung, hoặc qua một ổ đĩa cứng khác, song cách phổ biến nhất là cài đặt RedHat từ CD.

Tóm lại, dù chọn phương pháp cài đặt nào, chúng ta vẫn phải nắm trước một số thông tin cần thiết về cấu hình vật lý và mục tiêu sử dụng như được nêu sau đây:

- Loại bìa điều khiển màn hình, chipset và màn hình
- Tên cổng nối tiếp của chuột
- Tên cổng nối tiếp của modem
- Địa chỉ IP, cổng gateway và tên miền (nếu nối máy với mạng)
- Loại ổ đĩa cứng, ổ CD và bìa điều khiển của chúng
- Tổ chức các thư mục mà chúng ta muốn có trên máy của mình
- Tên chúng ta định đặt cho máy của mình với hệ Linux (hostname).

Nếu có kết nối Internet, chúng ta có thể yêu cầu người quản trị mạng hoặc ISP (công ty cung cấp dịch vụ Internet) cho biết các thông tin kể trên.

Nếu dự định sử dụng nhiều hệ điều hành khác nhau trên cùng một máy (chẳng hạn như Window 95, Windows NT hoặc Windows 2000), chúng ta phải tạo ra phân vùng cho mỗi hệ điều hành ấy. Chúng ta phải sử dụng chương trình tạo phân vùng của chính hệ điều hành đó, bởi vì Linux không quản lý được mọi loại phân vùng của các hệ điều hành khác.

Sau cùng, chúng ta nên kiểm tra xem vào giờ chót bản phát hành RedHat có những chỉnh sửa nào không bởi vì Linux được nâng cấp thường xuyên, hoặc được thêm vào những tiện ích và sửa lỗi.

1.2. Trình tự cài đặt

Lưu ý: trước khi cài đặt, cần tìm hiểu các thông tin về phần cứng của hệ thống, bao gồm :

- Thông tin về ổ đĩa cứng
- Thông tin về card mạng
- Thông tin về card đồ họa
- Thông tin về màn hình
- Thông tin về giao thức và cấu hình mạng nếu kết nối mạng
- Thông tin về các thiết bị ngoài.

Có thể chọn nhiều phương án cài đặt như cài đặt từ đĩa mềm, từ đĩa cứng, từ đĩa CD Rom hoặc qua mạng. Tài liệu này chọn hướng dẫn quá trình cài đặt phiên bản 7.0 từ đĩa CDRom. Yêu cầu máy cài đặt có khả năng khởi động (boot) từ ổ đĩa CD-Rom (được hỗ trợ hầu hết trong các máy tính hiện nay).

Sau đây là các bước cài đặt cụ thể. Khi kết thúc bước trước chương trình cài đặt tự động chuyển sang bước sau. Một số bước cài đặt cho phép quay lại bước trước bằng cách chọn Back.

Đặt đĩa boot mà chúng ta đã khởi tạo vào ổ đĩa cứng rồi khởi động lại máy. Sau khi hệ thống đã kiểm tra xong phần cứng và BIOS, các thông báo như sau sẽ

hiện trên màn hình, chúng ta cũng sẽ nhận được màn hình này nếu chúng ta đưa CD RedHat 7.0 vào và cho máy khởi động từ CD.

Chọn chế độ cài text :

Boot: text

Nếu màn hình có dòng nhắc, thông thường chúng ta chỉ cần bấm phím <Enter> để tiếp tục cài đặt theo các tham số mặc định. Bảng 3.2 sau mô tả các phím chức năng có thể hỗ trợ khi chúng ta gặp khó khăn.

Phím	Mô tả
F1	Hiển thị màn hình khởi động (phím <F1> luôn hiển thị màn hình này)
F2	Hiển thị màn hình cung cấp thông tin tổng quát về tiến trình cài đặt
F3	Hiển thị màn hình giải thích chế độ chuyên gia (expert mode). Thông thường, tiến trình cài đặt Linux tự động kiểm tra phần cứng và có thể làm treo máy, khi đó cần phải vào chế độ chuyên gia để xác định từng phần cứng hệ thống
F4	Hiển thị màn hình cung cấp thông tin về việc khởi tạo và sử dụng đĩa cấp cứu
F5	Hiển thị màn hình trợ giúp. Từ ver. 5.1 trở đi, nhờ ở tệp cấu hình việc cài đặt không cần trợ giúp này nữa
F6	Hiển thị màn hình khi cài đặt không suôn sẻ và cần phải khai báo thêm vài tham số cho kernel khởi động

Bảng 3.2 : Các phím chức năng

Màn hình hiển thị các thông báo như sau, rồi bắt đầu nạp hệ thống:

Loading initrd.img...

Loading vmlinuz...

Chú ý dòng thông báo ở cuối màn hình trên: *Unable to probe*. Nó báo cho biết Linux không hiểu bìa điều khiến màn hình và màn hình của chúng ta. Chúng ta cần phải cấu hình lại bằng tay.

Sau khi khởi động xong (minh họa trên), hệ thống của chúng ta sẽ hiển thị màn hình đón chào (Welcome) như ở minh họa sau.

Lưu ý rằng nếu sử dụng chuột qua cổng USB, chúng ta nên di chuyển chuột trong khi chờ hiện ra màn hình Welcome để chương trình cài đặt có thể phát hiện ra loại chuột

1.2.1. Cấu thống

Bấm
để tiếp tục.



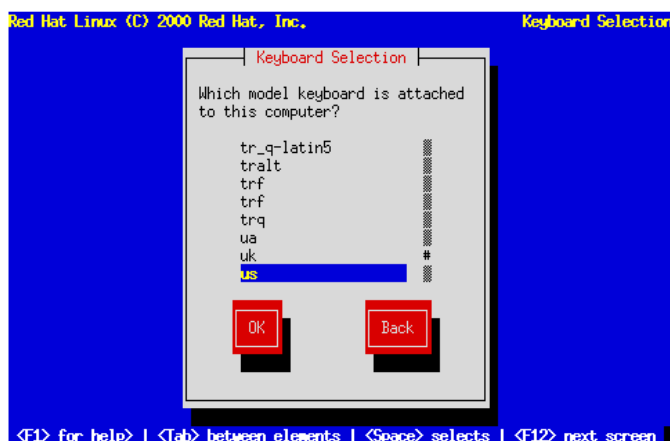
đó
hình hệ
phím <Enter>

Minh hoạ 3.4 : Màn hình chọn ngôn ngữ cài đặt

Màn hình hỏi chúng ta chọn ngôn ngữ nào (Anh, Pháp v.v. hay Việt ?) trong khi cài đặt. Ngôn ngữ được chọn sẽ là ngôn ngữ mặc định cho hệ thống của chúng ta sau khi cài đặt.

Ghi chú: Chúng ta có thể sử dụng phím để di chuyển trong hộp thoại như chương trình có nhắc nhở ở dòng cuối màn hình. Muốn di chuyển từ trường này sang trường kia, hãy bấm phím <Tab> hoặc <Alt-Tab>. Dùng Spacebar (phím ký tự trống) để chọn lựa danh sách hoặc để đánh dấu vào ô lựa chọn. Sau khi chọn ô OK hay Cancel, hãy bấm phím <Enter>. Muốn di chuyển qua danh sách liệt kê, hãy dùng phím mũi tên.

Hộp thoại như ở minh hoạ dưới yêu cầu chúng ta chọn loại bàn phím cho hệ thống. Sau này khi muốn thay đổi lựa chọn về loại bàn phím, hãy dùng `/usr/sbin/kbdconfig` (khi chúng ta đóng vai trò là *superuser* hoặc *root*).



Minh hoạ 3.5 : Màn hình chọn bàn phím

Hộp thoại kế tiếp yêu cầu chúng ta khai báo loại chuột đang dùng. Từ phiên bản RedHat 7.x, chúng ta đã có sự hỗ trợ cho loại chuột dùng cổng USB và cả loại chuột dùng cổng hồng ngoại (Infrared), còn gọi là loại chuột không dây (wireless). Cần chú ý kiểm tra cổng kết nối của chuột với máy tính xem đó là

cổng COM (Serial), USB hay PS/2. Hãy nhớ rằng cổng PS/2 có dạng hình tròn, cổng USB có dạng hình chữ nhật và cổng COM có dạng hình thang. Nếu chuột của chúng ta kết nối bằng cổng COM, chúng ta phải chú ý xem nó kết nối qua cổng số mấy: COM1 hay COM2.

Chúng ta cũng cần nghĩ xem có nên sử dụng tính chất phỏng tạo chuột 3 phím hay không, nếu chúng ta chỉ có chuột 2 phím. Mặc định là không nhưng chúng ta nên chọn có, điều này sẽ hỗ trợ cho chúng ta khi chạy trong môi trường X Window, khi đó có thể bấm cùng lúc cả 2 phím trái-phải của chuột để thay cho phím giữa, giống như có chuột 3 phím.

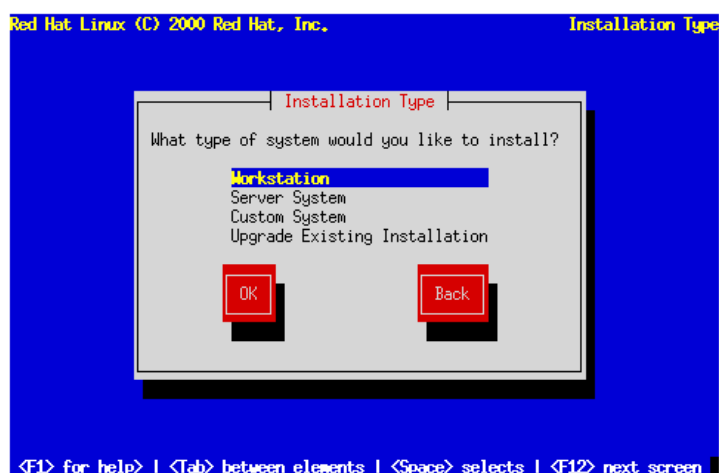
Màn hình đón chào thứ hai sẽ hướng dẫn chúng ta thêm về trình tự cài đặt cũng như đăng ký với RedHat để nhận sự giúp đỡ trực tuyến. Việc này đòi hỏi chúng ta phải mua chương trình cài đặt từ RedHat, tức là không dành cho các bản tải nạp miễn phí.



Minh hoạ 3.7 : Màn hình chào mừng

Chú ý rằng các màn hình của RedHat luôn có phần trợ giúp nằm ở phía bên trái. Chúng ta có thể tắt chúng bằng cách nhấp chuột vào ô “Hide Help”.

1.2.2. Tùy chọn cài đặt



Minh hoạ 3.8 : Màn hình “Tùy chọn cài đặt”

Màn hình “Tùy chọn cài đặt” xác định việc cài đặt là để máy chúng ta sẽ đóng vai trò gì. Chúng ta có các tùy chọn sau đây:

- **Workstation**: hãy chọn vai trò này khi chúng ta mới làm quen với Linux, muốn sử dụng máy của chúng ta một cách độc lập tại nhà hay xem nó chỉ như một trạm làm việc (Workstation) trong một mạng có sẵn. Chúng ta có thể gặp rắc rối nho nhỏ nếu cài đặt trên máy đã có dữ liệu (Windows 9x hay NT, 2000...).

- **Server**: có thể chọn vai trò này nếu chúng ta muốn thực hành như một người quản trị mạng đang tự xây dựng máy chủ (Server). Chú ý rằng với tùy chọn này tất cả dữ liệu đang có trên đĩa của chúng ta sẽ bị xóa sạch.

- **Laptop**: tương tự như Workstation, nhưng vai trò của tùy chọn này chủ yếu là để tối ưu hóa cho việc cài đặt một máy tính xách tay (laptop).

- **Custom**: có thể chọn vai trò cài đặt này khi chúng ta muốn cài đủ thứ trên máy của mình. Hãy sử dụng vai trò này khi chúng ta rất quen thuộc Linux và khi chúng ta cần một sự linh động tối đa trong việc cấu hình máy theo Linux.

- **Upgrade**: việc nâng cấp hệ thống hiện hành sẽ dễ dàng hơn khi chúng ta chọn vai trò này. Nó sẽ giữ lại dữ liệu và các cài đặt cũ của chúng ta, chỉ cập nhật các gói phần mềm và kernel của Linux một cách nhanh chóng và an toàn. Các cách cài đặt khác không đảm bảo việc bảo lưu dữ liệu cho chúng ta.

Ghi chú: Muốn xem chương trình cài đặt đang thực hiện thao tác nào, chúng ta bấm <Alt-F3> để chuyển sang terminal ảo.

1.2.3. Phân vùng đĩa cứng

Linux đưa ra cho bạn hai phần mềm để phân chia ổ đĩa dành cho Linux: đó là Disk Druid và fdisk.

1.2.3.1. Sử dụng fdisk của Linux

Khi dùng chương trình **fdisk** như một chuyên gia, trước hết chúng ta sẽ phải chọn ổ đĩa cần phân vùng (máy PC thông thường có 1 ổ đĩa cứng như thí dụ minh họa dưới đây).

Lưu ý: Chúng ta nhớ dùng chương trình **fdisk** của chính Linux và nên cẩn thận bởi vì nó khác với những **fdisk** của các hệ điều hành khác, chẳng hạn như DOS, Windows 98, OS/2 hoặc Windows 2000. Chúng ta không thể sử dụng lẫn lộn những chương trình đó, thí dụ không thể dùng **fdisk** của Linux để sắp xếp một phân vùng do DOS tạo ra.

Chúng ta có thể gõ lệnh **m** để liệt kê danh sách các lệnh của chương trình **fdisk** :

Lệnh	Command action (tiếng Anh)	Ý nghĩa
a	toggle a bootable flag	Bật, tắt cờ <i>bootable</i> (khởi động được)
c	toggle a DOS-compatible flag	Bật, tắt cờ tương thích DOS
d	delete a partition	Xoá một phân vùng
l	list known types	Liệt kê các kiểu phân vùng đã biết
m	print this menu	Hiển thị bảng này
n	add a new partition	Bổ sung một phân vùng
p	print the partition table	Hiển thị bảng phân vùng
q	quit without saving changes	Ra khỏi mà không ghi lại sự thay đổi
t	change a partition's system id	Thay đổi một <i>system id</i> của phân vùng
u	change display/entry units	Thay đổi các đơn vị hiển thị/nhập
v	verify the partition table	Kiểm tra bảng phân vùng
w	write table to disk and exit	Ghi bảng phân vùng vào đĩa và ra khỏi
x	extra functionality (expert only)	Chức năng phụ (dành cho chuyên gia)

Bảng 3.3: Danh sách các lệnh của chương trình **fdisk**

Khi bắt đầu phân vùng, chúng ta nên gõ lệnh p (bấm <p> <Enter>) để hiển thị bảng phân vùng của ổ đĩa cứng đã được thực hiện trước đó bằng chương trình **fdisk** của DOS.

Thí dụ dưới đây cho thấy màn hình sau khi thực hiện lệnh p hiển thị gì :

The number of cylinders for this disk is set to 2482

There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:

1) software that runs at boot time (e.g., old versions of LILO)

2) booting and partitioning software from other OSs

Command (m for help): p

Disk /dev/hda: 255 heads, 63 sectors, 2482 cylinders

Unit = cylinders of 16065 * 512 bytes

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1		1	6	48163+	83	Linux
/dev/hda2		7	159	1228972+	82	Linux swap
/dev/hda3		160	2482	18659497+	83	Linux

Khung trên cho thấy các phân vùng đã được xác định ở đĩa nào, cùng với vị trí khởi đầu và kết thúc của phân vùng, kích cỡ của các block và các loại phân vùng. Bảng 3.4 sau đây tóm tắt những loại phân vùng quen biết của Linux:

Loại phân vùng	Giá trị	Loại phân vùng	Giá trị
Empty	00	Novell Netware 386	65
DOS 12-bit FAT	01	PIC/IX	75
XENIX root	02	Old MINIX	80
XENIX usr	03	Linux/MINIX	81
DOS 16-bit <=32M	04	Linux swap	82
Extended	05	Linux native	83
DOS 16-bit >=32	06	Linux extended	85
OS/2 HPFS	07	Amoeba	93
AIX	08	Amoeba BBT	94
AIX bootable	09	BSD/386	a5
OS/2 Boot Manager	0a	Open BSD	a6
Win95 FAT32	0b	NEXTSTEP	a7
Win95 FAT32(LBA)	0c	BSDI fs	b7
Win95 FAT16(LBA)	0e	BSDI swap	b8
Win95 Extended (LBA)	0f	Syrinx	c7
Venix 80286	40	CP/M	db
Novell	51	DOS access	e1
Microport	52	DOS R/O	e3
GNU HURD	63	DOS secondary	f2
Novell Netware 286	64	BBT	ff

Bảng 3.4 : Các loại phân vùng đã biết của Linux

Bảng trên trình bày tất cả các loại phân vùng có thể định nghĩa bằng chương trình **fdisk** của Linux, trong đó những loại phân vùng sơ cấp thường dùng là Linux Native và Linux Swap. Nếu dùng lệnh **l** của **fdisk** thì chúng ta cũng sẽ thấy kết quả tương tự như thế. Vì không cần phân vùng lại ổ đĩa cứng cho DOS cho nên chúng ta không phải xoá bỏ bất kỳ phân vùng nào của Linux. Chúng ta chỉ phải thêm phân vùng vào. Trong thí dụ sau chúng ta cùng tạo lại phân vùng như kết quả trên. Giả sử đĩa cứng của chúng ta đã tạo được phân vùng thứ 1 (hda1), chúng ta tạo thêm phân vùng 2 mới bằng cách gõ lệnh **n** như sau:

Command Action : n

e extended

p primary (1-4)

Bấm <p> <Enter>, **fdisk** sẽ hỏi chúng ta số phân vùng. Chúng ta gõ vào theo ý mình và bấm <Enter>. Nếu chúng ta cho một số phân vùng đã được dùng, **fdisk** sẽ thông báo việc này và hỏi chúng ta xem có xoá phân vùng ấy không, trước khi thử thêm nó vào bảng phân vùng. Với thí dụ này, chúng ta nhập số 2 để thêm vào một phân vùng sơ cấp thứ hai, mang dòng tham khảo là /dev/hda2.

Partition number (1-4):2

First cylinder (7-2482, default 7):

Using default value 7

Last cylinder or +size or +sizeM or +sizeK (7-2482, default 2482): 159

Tiếp theo, máy sẽ hỏi vị trí của cylinder (trụ đĩa) đầu tiên là gì. Thông thường đó là cylinder đầu tiên có sẵn. Trên thực tế **fdisk** hiển thị một giá trị mặc định cho chúng ta chọn, trong thí dụ sau giá trị mặc định là 7:

First cylinder (7-2482, default 7):

Chúng ta thấy rằng phân vùng đầu tiên kết thúc ở cylinder 6 và đĩa có cylinder cuối cùng là 2482. Vì thế khoảng trống do **fdisk** cung cấp cho phép chúng ta khởi đầu phân vùng kế tiếp bất kỳ ở điểm nào trong tầm 7-2482. Việc không đặt phân vùng ở bất kỳ chỗ nào trong ổ đĩa cứng là một ý hay. Chúng ta nên chọn điểm kế tiếp nào còn trống. Trong trường hợp này đó là cylinder 7. Gõ 7 và bấm <Enter>.

Ghi chú: Nếu khởi động Linux từ các phân vùng khởi đầu ở các cylinder cao hơn 1024 thì có thể bị rắc rối. Trong trường hợp bất đắc dĩ phải tạo phân vùng ở những vị trí cao hơn cylinder 1024, có thể chúng ta phải khởi động Linux từ đĩa mềm. Cuối chương này sẽ có hướng dẫn chúng ta tạo đĩa mềm khởi động. Khởi động Linux từ đĩa mềm sẽ lâu hơn khởi động từ ổ đĩa cứng.

Đến đây **fdisk** muốn chúng ta chỉ định dung lượng cho phân vùng mới. Chúng ta có thể chỉ định kích thước (size) bằng số cylinder hoặc bằng số byte (+size), số kilobyte (+sizeK), hoặc megabyte (+sizeM).

Bởi vì chúng ta đã biết kích thước gần đúng của tệp hoán chuyển, cho nên chúng ta hãy chỉ định phân vùng này trước, sau đó chừa phần còn lại của ổ đĩa

cứng cho các phân vùng chương trình Linux. Ở thí dụ này, máy của chúng ta có 8 MB RAM, do đó chúng ta chỉ định phân vùng 16 MB bằng cách trả lời như sau:

```
Last cylinder or +size or +sizeM or +sizeK (42-1023): 159
```

Sau đó chúng ta dùng lệnh `p` để xem bảng phân vùng mà chúng ta vừa chỉ định. Ở thí dụ này, bảng phân vùng mới sẽ có dạng như sau:

```
root@mail linux-2.2.12]#/sbin/fdisk/dev/hda
```

```
The number of cylinders for this disk is set to 2482. There is nothing wrong with that, but this is larger than 1024,
```

```
and could in certain setups cause problems with:
```

```
1) software that runs at boot time (e.g., LILO)
```

```
2) booting and Partitioning software from other OSs (e.g., DOS FDISK, OS/2 FDISK)
```

```
Command (m for help): p
```

```
Disk /dev/hda: 255 heads, 63 sectors, 2482 cylinders
```

```
Units = cylinders of 16065*512 bytes
```

Device	Start	End	Blocks	Id	System
Boot					
/dev/hda1	1	6	48163+	83	Linux
/dev/hda2	7	2482	1228972+	83	Linux

Theo mặc định, **fdisk** sẽ định dạng phân vùng mới như là loại Linux Native. Muốn đổi nó thành phân vùng hoán chuyển (swap partition), chúng ta dùng lệnh `t`. Gõ `t`, sau đó gõ số phân vùng chúng ta muốn đổi (gõ số 2 trong thí dụ này). Chương trình **fdisk** sẽ yêu cầu chúng ta gõ giá trị thập lục phân của loại phân vùng theo bảng 2.7 (nếu không có sẵn bảng ấy, chúng ta chỉ gõ 1 và có ngay danh sách mã). Ở đây, bởi vì chúng ta muốn có một phân vùng hoán chuyển nên cần gõ số 82 tại dấu nhắc.

```
Command (m for help): t
```

```
Partition number (1-4): 2
```

```
Hex code (type L to list codes) : 82
```

```
Changed system type of Partition 2 to 82 (Linux swap)
```

Như chúng ta thấy, **fdisk** thông báo loại phân vùng mới, tuy nhiên chúng ta vẫn có thể dùng lệnh `p` để kiểm tra lại lần nữa xem giờ đây phân vùng 2 có phải là phân vùng hoán chuyển của Linux hay không.

Giờ chúng ta có thể thêm vào các phân vùng Linux của mình. Ở thí dụ này chúng ta chỉ cần thêm vào một phân vùng, song nếu muốn có nhiều hơn thì cứ thêm vào. Muốn thêm, chúng ta bấm `<n>`, chỉ định `p` cho một phân vùng sơ cấp khác và sau đó chỉ định số hiệu cho phân vùng ấy (trong thí dụ là 4). Để tránh tình trạng rải rác nhiều phân vùng khắp ổ đĩa cứng, chúng ta nên khởi đầu phân vùng sau cùng tại địa điểm mà phân vùng áp chót vừa kết thúc (trong thí dụ là cylinder 160). Đối với cylinder cuối cùng, vì chúng ta muốn sử dụng dung lượng còn lại

cho hệ Linux, cho nên chúng ta có thể chỉ định cylinder cuối thay vì phải nói chính xác là bao nhiêu byte. Do đó chúng ta gõ 2482 như sau:

Command (m for help) : n

Command action

e extended

p primary partition (1-4)

p Partition number (1-4): 3

First cylinder (160-2482, default 160):

Using default value 160

Last cylinder or +size or +sizeM or +sizeK (160-2482, default 2482):

Using default value 2482

Đến đây chúng ta nên gõ lệnh **p** để kiểm tra phân vùng mới. Nếu muốn thay đổi tiếp thì hãy thực hiện cẩn thận như trên.

Trong thí dụ về phân vùng trên ta thấy rằng khi khai báo là primary partition, số lượng tối đa chỉ là 4. Còn nếu khai báo là extended, ta có thể có số lượng phân vùng là tùy ý.

Khi đã hài lòng với tổng thể các phân vùng, chúng ta dùng lệnh **w** để ghi thông tin bảng phân vùng vào ổ đĩa cứng. Một khi chưa sử dụng lệnh **w** thì những thay đổi sẽ không được lưu trữ. Vì vậy trong khi thao tác nếu cảm thấy đã lỡ thay đổi điều gì đó không đúng ý, chúng ta dùng lệnh **q** để thoát ra mà không thay đổi gì trong bảng phân vùng. Nếu gõ **w**, Linux sẽ báo cho chúng ta biết rằng bảng phân vùng đã được đổi, kế tiếp Linux cập nhật các đĩa cho khớp với bảng phân vùng mới. Nếu làm đến đây mà máy bị treo, hãy khởi động lại bằng đĩa cài đặt và đĩa root cho đến khi chúng ta gặp dấu nhắc **#** trở lại.

Cẩn thận: Đừng dùng chương trình **fdisk** của Linux để tạo ra hoặc chỉnh sửa các phân vùng của những hệ điều hành khác. Điều này có khả năng làm cho cả hai hệ điều hành đều không sử dụng được ổ đĩa cứng nữa.

Tạo phân vùng hoán chuyển: Một vài bản phát hành Linux có cho phép việc tự động tạo ra và kích hoạt tệp swap trong tiến trình cài đặt. Song ở một vài bản phát hành khác, người sử dụng phải tạo và kích hoạt tệp ấy trước khi tiếp tục tiến trình cài đặt.

Ghi chú: Nếu trong tiến trình cài đặt chúng ta bị báo lỗi “out of memory” thì chúng ta phải tăng thêm dung lượng tệp swap.

Với RedHat 6.x trở lên, ta có thể bỏ qua mục này vì RedHat đã tự kích hoạt phân vùng hoán chuyển thay cho chúng ta.

Muốn tạo ra khoảng swap, chúng ta dùng lệnh **mkswap** sau đó cho máy biết dùng phân vùng nào và dung lượng cho RAM ảo là bao nhiêu. Thí dụ muốn tạo khoảng swap trên phân vùng/dev/hda3, tại dấu nhắc **#** chúng ta gõ lệnh như sau:

```
#mkswap -c /dev/hda3 16447
```

Số 16447 có nghĩa là 16 MB có thể nhìn thấy ở cột block trên màn hình khi gõ chương trình **fdisk**. Tùy chọn **-c** báo cho **mkswap** biết để kiểm tra những bad

sector (đoạn sector bị hỏng) trên phân vùng. Sau đó chúng ta kích hoạt hệ thống swap bằng lệnh swapon: **#swapon /dev/hda3**

Nhắc lại lần nữa rằng nếu đang sử dụng CD RedHat Linux để cài đặt, chúng ta không phải lo lắng gì về việc kích hoạt hệ thống swap khi chúng ta tạo ra phân vùng cho một người dùng. Trong khi cài đặt, chương trình sẽ phát hiện phân vùng hoán chuyển và tự động khởi động hệ thống để cài đặt.

Sau khi tạo các phân vùng trên những ổ đĩa cứng khác nhau và trở về hộp thoại phân vùng đĩa, chúng ta bấm ô chữ “Done” để tiếp tục tiến trình cài đặt.

Tạo khoảng swap xong, chương trình sẽ hiển thị hộp thoại Select Root Partition. Root Partition (phân vùng gốc) là hệ thống tệp chính của máy chúng ta dành cho Linux, nơi mà tất cả các tệp boot được lưu trữ. Từ hộp liệt kê danh sách, chúng ta hãy chọn tên ổ đĩa cứng để làm Root Partition và bấm <Enter>. Từ hộp thoại phân vùng đĩa cứng chúng ta có thể thiết lập các phân vùng khác.

Từ đây chúng ta có thể lập bất kỳ hệ thống tệp DOS hoặc OS/2 nào để sau đó có thể truy cập bằng Linux. Từ hộp liệt kê danh sách, chúng ta chọn phân vùng nào cần chỉnh sửa sau đó bấm <Enter>. Từ hộp thoại Edit Mount Point chúng ta xác định một điểm lắp ghép (**mount point**), nghĩa là một thư mục, nơi chúng ta muốn lập phân vùng ấy.

Sau khi chúng ta chọn xong **mount point** và root (điểm gốc) cho các phân vùng, chương trình sẽ tạo định dạng (format). Muốn xác định phân vùng nào cần format, chúng ta chọn tại hộp thoại Format Partition.

1.2.3.2. Sử dụng Disk Druid



Minh họa 3.12: Phân vùng đĩa bằng DiskDruid

Bấm ô Disk Druid ở hộp thoại Disk Setup, chúng ta sẽ thấy màn hình phân vùng đĩa như ở minh họa 3.12. Disk Druid giúp chúng ta tạo ra phân vùng, thiết

lập **mount point**, kích thước phân vùng, loại hệ thống tệp và biết được thông tin về thuộc tính. Dưới đây sẽ liệt kê các ô bấm khác nhau trên màn hình chính của Disk Druid, cùng với chức năng của từng ô. Các ô bấm của chương trình Disk Druid cho phép chúng ta tiến hành phân vùng một cách trực quan:

- **New**: Dùng để tạo một phân vùng mới. Một hộp thoại sẽ xuất hiện như minh họa.

- **Edit**: Dùng để điều chỉnh thuộc tính của phân vùng được chọn (phân vùng được chọn là phân vùng được đánh dấu trong vùng Partition trên màn hình). Hộp thoại chỉnh sửa sẽ xuất hiện, cấu trúc của nó cũng giống như của hộp thoại Tạo mới.

Chúng ta cũng có thể tạo một phân vùng mới bằng cách Edit vùng “Free Space” của vùng Partition.

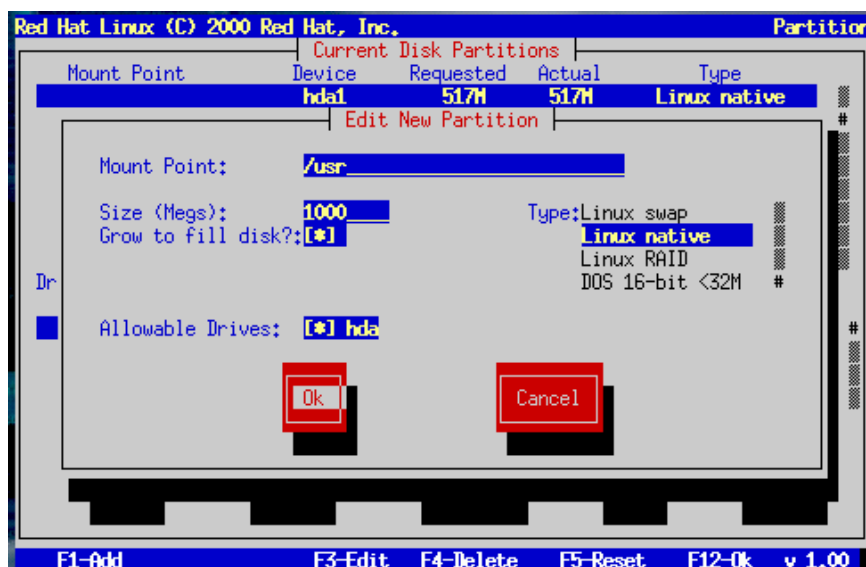
- **Delete**: Dùng để xóa phân vùng được đánh dấu trong vùng “Current Disk Partitions”.

- **Reset**: Dùng để hoàn lại tình trạng gốc trước khi sửa đổi.

- **Make RAID**: Có thể dùng để cung cấp dịch vụ RAID cho bất kỳ phân vùng nào.

Chúng tôi hy vọng là chúng ta không sử dụng chức năng RAID, trước khi hiểu rõ nó thông qua tài liệu “Official RedHat Linux Customization Guide” hay tương tự. Chức năng này hỗ trợ trong chừng mực nào đó việc bảo đảm tính toàn vẹn và bền vững của dữ liệu trên đĩa, tránh những hư hỏng có thể xảy ra.

Qua hộp thoại tạo phân vùng mới, chúng ta nhập một giá trị tương ứng vào trường **Mount Point**, chẳng hạn như giá trị phân vùng gốc (/) hoặc phân vùng var (/var). Sau đó chúng ta cho biết dung lượng của phân vùng tính bằng đơn vị megabyte, đồng thời xác định xem có muốn sau này phân vùng ấy có thể phình ra theo yêu cầu hay không (phân vùng có thể phình ra khi chúng ta thêm bớt những phân vùng khác).



Minh hoạ 3.13: Tạo phân vùng mới

Tiếp theo chúng ta chọn từ hộp liệt kê “ FileSystem Type” để quyết định xem loại hệ thống tệp nào sẽ hiện diện trên phân vùng ấy. Cuối cùng, từ danh sách “ Allowable Drive” chúng ta chọn xem ổ đĩa cứng vật lý nào sẽ chứa phân vùng.

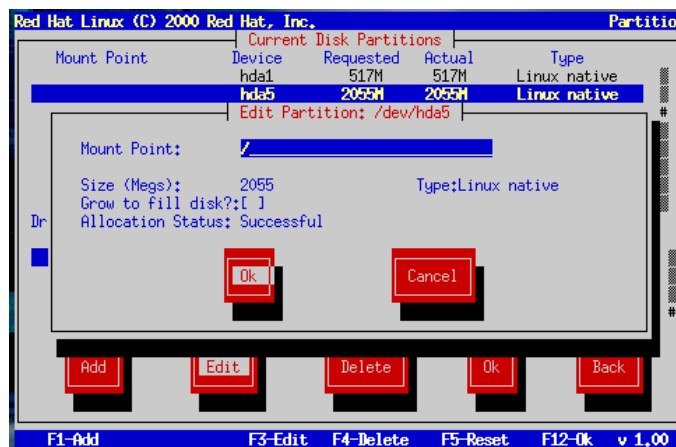
Ghi chú: Nếu chúng ta chỉ định một kích cỡ lớn hơn khoảng trống còn lại trên ổ đĩa cứng đã chọn, Disk Druid sẽ báo lỗi và đề nghị chúng ta giảm dung lượng đăng ký. Disk Druid cũng báo cho chúng ta biết những vấn đề có thể xảy ra, đồng thời cung cấp các giải pháp khả dĩ.

Một số vấn đề có thể xảy ra khi thêm một partition



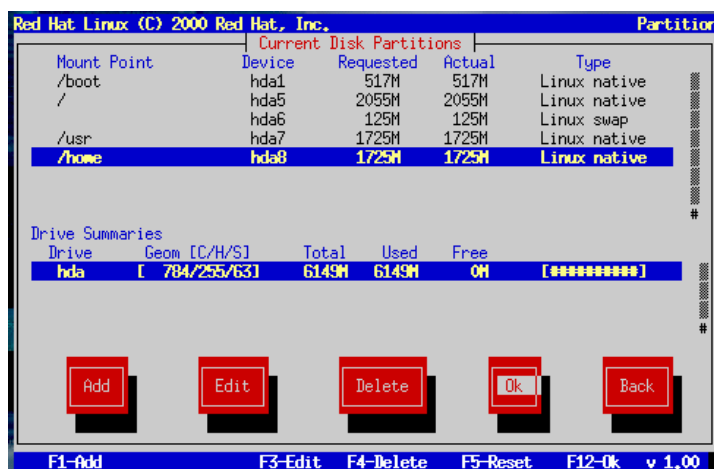
+ Hiệu chỉnh một partition

Chọn một partition cần hiệu chỉnh, nhấn Edit, màn hình mới sẽ cho phép bạn thay đổi các thông số của partition đã chọn như kích thước, kiểu, .



+ Hoàn thành việc phân chia đĩa

Chương trình cài đặt sẽ yêu cầu bạn format lại phân vùng vừa tạo, chú ý không chọn những phân vùng dữ liệu quan trọng.



1.2.3.3. Phân vùng tự động

Khi chúng ta mới làm quen với Linux, chúng ta có thể để Linux tự động phân vùng đĩa cứng. Chúng ta có thể chọn một trong 3 tùy chọn:

- Remove all Linux Partition on this system - chỉ xoá những phân vùng Linux cũ, không xoá các phân vùng khác.

- Remove all Partition on this system – xoá tất cả các phân vùng trên các đĩa cứng, kể cả các phân vùng tạo bởi hệ điều hành khác (như Windows 95/98/NT/2000).

Chú ý: Tùy chọn này cho phép chương trình cài đặt xoá hết mọi dữ liệu chúng ta đang có trên đĩa.

- Keep all Partition and use existing free space - chúng ta cần giữ lại các dữ liệu hiện hành của chúng ta. Tuy nhiên, chúng ta cũng cần đủ không gian trống cần thiết cho việc cài đặt mới.

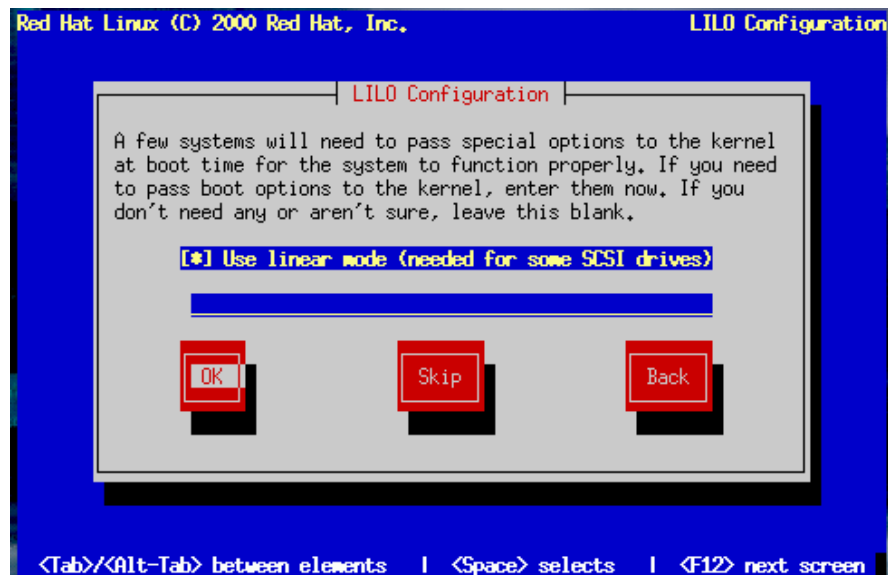
1.2.4. Cài đặt chương trình khởi động

Chúng ta cần chọn một chương trình khởi động khi muốn khởi động hệ Linux của chúng ta mà không cần đĩa mềm. Chúng ta có thể chọn GRUB (Grand Unified Bootloader), một chương trình khởi động mới của RedHat, hoặc phổ biến nhất là LILO (Linux Loader), chương trình khởi động chuẩn của Linux (minh hoạ 3.15).

Trường hợp chúng ta đã có một hệ điều hành khác trên máy, chúng ta có thể không cần cài đặt trình khởi động nào cả, nhưng thay vào đó, chúng ta không được quên việc tạo đĩa khởi động (boot disk) ở cuối chương trình cài đặt.

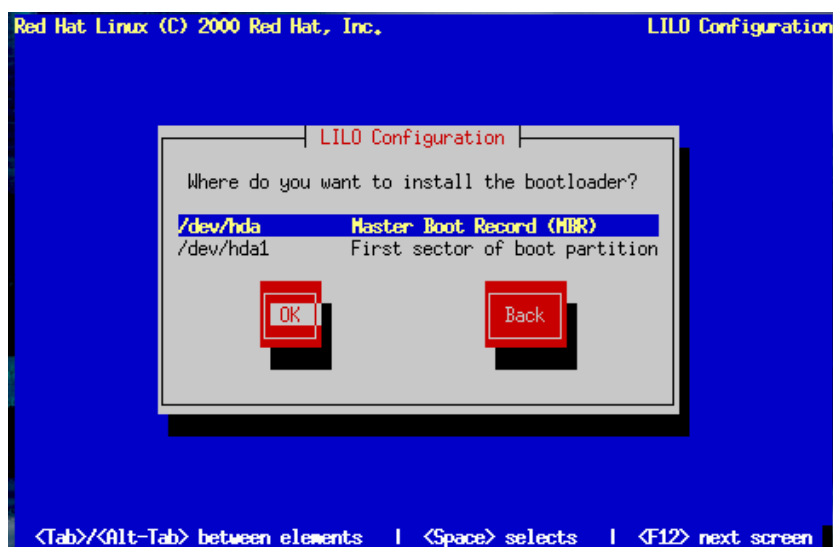
Việc xác định nơi cài đặt trình khởi động rất quan trọng: nếu như chúng ta chỉ định sai, rất có thể hệ điều hành đang hoạt động sẽ bị “biến mất” hay máy tính của chúng ta sẽ bị treo và việc chỉnh sửa trở lại khá phức tạp.

Chúng ta có thể cài đặt vào MBR (Master Boot Record), một vùng đặc biệt và duy nhất trên máy của chúng ta, được BIOS nạp vào tự động và là điểm đầu tiên mà trình khởi động tác động đến trong quá trình khởi động. Nếu chúng ta cài đặt trình GRUB hay LILO vào MRB, chúng sẽ hiện dấu nhắc khởi động. Tùy theo cách cấu hình sau này, chúng ta có thể sử dụng chúng để nạp các hệ điều hành khác nhau.



Minh hoạ 3.15: Khởi động hệ điều hành

Dùng lựa chọn đặt boot loader tại Master Boot Record để khởi tạo ngay hệ điều hành Linux khi bật máy.



Màn hình này cho phép chúng ta đặt tên cho máy tính của chúng ta đang sử dụng. Chúng ta có thể thay đổi hostname sau khi đã cài đặt xong bằng lệnh `hostname newname`, trong đó `newname` là tên mà chúng ta muốn đặt



Nếu hệ điều hành đang có trên máy chúng ta cũng đang tác động đến MRB như OS/2, Window 2000... chúng ta có thể chỉ cần cài GRUB hay LILO vào vùng “First sector of Root Partition”. Khi đó, chương trình khởi động của hệ điều hành hiện hành sẽ quản lý việc khởi động chung. Chúng ta sẽ phải định lại cấu hình cho chương trình khởi động ấy để có thể nạp GRUB hay LILO khi chúng ta muốn khởi động hệ Linux.

Nếu hệ điều hành trên máy chúng ta chỉ là Linux, hay Windows 9x, chúng ta nên cài đặt LILO hay GRUB vào MRB.

Chú ý: Chúng ta không cần lo khi lần đầu khởi động mà không thấy xuất hiện mục khởi động Windows XP hay Windows 2000 trên trình khởi động GRUB. Các phân vùng của Windows vẫn tồn tại. Nhưng chúng ta phải bổ sung tệp boot (sẽ mô tả thêm trong phần sau).

Chúng ta có thể thêm các tham số cho việc khởi động từ LILO hay GRUB bằng cách thêm vào ô “Kernel parameters”. Chúng ta sẽ trở lại vấn đề này ở các chương sau.

Tùy chọn “Force use of LBA32” cho phép chúng ta vượt qua giới hạn 1024 cylinder của phân vùng “boot”. Nếu hệ thống của chúng ta hỗ trợ LBA32 mà trình cài đặt không tự phát hiện được phần mở rộng này của BIOS, chúng ta nên đánh dấu vào tùy chọn này.

1.2.4.1. Các chương trình khởi động khác

Ta có thể dùng đĩa mềm khởi động, Loadlin, Syslinux và các phần mềm thương mại khác như System Commander, Partition Magic, Boot Start... Việc sử dụng Loadlin khá phức tạp, đòi hỏi có một bản sao của phần Linux kernel trên một DOS Partition. Ta sẽ khởi động bằng một chương trình khởi động nào đó rồi sao chép phần kernel sang một DOS Partition.

1.2.4.2. Mật khẩu cho GRUB

Nếu không chọn GRUB hay LILO, chúng ta có thể chuyển đến mục 3.3 “Thiết lập cấu hình mạng” ở dưới đây (minh họa 3.17).

Chúng ta cần cung cấp mật khẩu cho chương trình khởi động GRUB để ngăn ngừa các vụ xâm phạm tính bảo mật từ phía người dùng.

2. Thiết lập các cấu hình cho RedHat

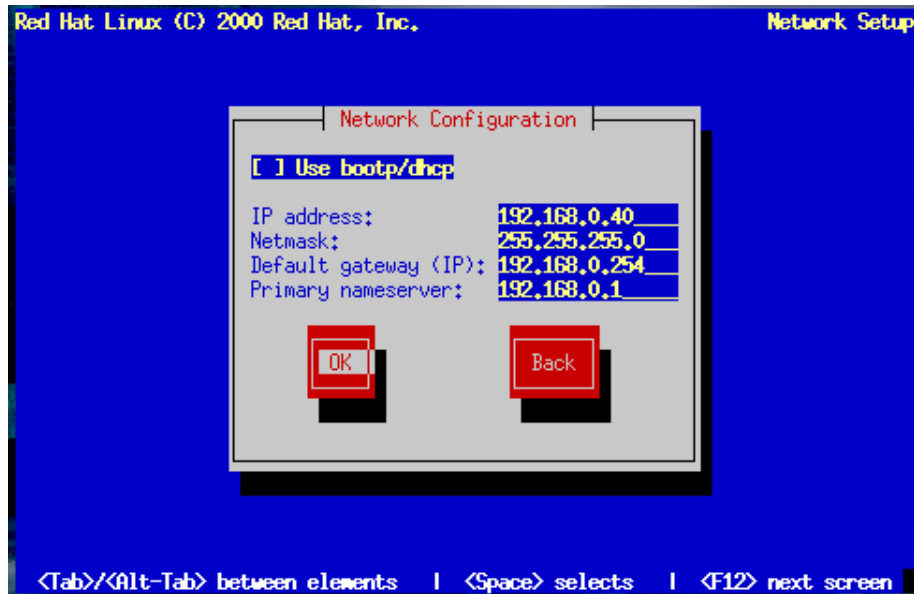
Mục tiêu :

- Thiết lập được cấu hình mạng TCP/IP
- Thiết lập được cấu hình bức tường lửa và cấu hình khác

2.1. Thiết lập cấu hình mạng TCP/IP

Chúng ta khai báo thông tin về TCP/IP qua hộp thoại Network Configuration. Quản trị viên mạng hoặc nhà cung cấp dịch vụ Internet sẽ cung cấp cho chúng ta các thông tin sau đây:

IP number, netmask, network address và broadcast address.



Minh hoạ 3.17: Thiết lập cấu hình mạng

Sau đó chương trình sẽ thiết lập cấu hình mạng máy chúng ta. Chúng ta khai báo vào hộp thoại Network Configuration các thông tin như tên miền và host name của hệ thống. Tên miền là hai phần cuối của địa chỉ Internet. Thí dụ nếu địa chỉ là www.citynet.com, thì citynet.com là tên miền, còn www là host name. Tuy nhiên ở đây chúng ta cần ghi tên đầy đủ (FQDN=Fully qualified Domain Name) vào ô “hostname”.

Tiếp theo quản trị viên mạng sẽ cho chúng ta giá trị của cổng kết nối (Gateway) mặc định và máy chủ tên miền chính (Primary Name Server). Nếu mạng của chúng ta có nhiều máy chủ tên miền, chúng ta hãy điền giá trị của các server này vào nơi thích hợp.

Ghi chú: Nên thận trọng khi đặt tên host, bởi vì tên này sẽ xuất hiện ở câu nhắc mặc định, ở các tin nhắn và ở các báo cáo ký sự (log).

Ngay cả khi máy chúng ta không thực sự nối vào mạng, chúng ta cũng có thể đặt tên cho máy của mình. Còn không thì máy của chúng ta sẽ được coi như có tên là localhost.

2.2. Cấu hình bức tường lửa

Việc tích hợp bức tường lửa là một tính chất mới của các phiên bản RedHat 7.x và cho phép ta bảo vệ hệ thống (xem minh hoạ 3.18). Chúng ta có thể chọn các mức bảo mật sau đây tuỳ theo yêu cầu của mình:

- **High:** hệ thống chúng ta sẽ không chấp nhận các kết nối mà chúng ta chưa khai báo tường minh. Một cách mặc định, chỉ có các kết nối hồi âm DNS (DNS replies) và DHCP là được cho phép (tất nhiên các kết nối sau đây sẽ không được phép: FTP, IRC, RealAudio™, Remote X Window client, v.v.).

- **Medium**: bức tường lửa của chúng ta sẽ cấm các cuộc truy cập từ xa đến tài nguyên trên máy chúng ta. Một cách mặc định, mọi kết nối đến các cổng TCP có số hiệu nhỏ hơn 1023 và các cổng NFS server (2049), cổng X Font server, v.v. đều sẽ bị cấm.

- **No Firewall**: chế độ bảo mật không được đặt ra, bất cứ kết nối nào đến máy chúng ta cũng sẽ không bị cấm. Chúng ta chỉ nên sử dụng tùy chọn này khi chúng ta đang ở trong một mạng đáng tin cậy. Tuy nhiên nếu chúng ta mới tìm hiểu về Linux, chúng ta nên chọn chế độ này để có thể sử dụng ngay một số dịch vụ. Sau đó ta sẽ xem xét lại chế độ Bức tường lửa.

- **Customize**: tùy chọn này cho phép chúng ta chủ động hơn trong việc xác định các thiết bị, kết nối, dịch vụ nào là đáng tin cậy. Thí dụ: chúng ta nối vào mạng riêng qua NIC (Ethernet bìa – eth0) và dùng modem qua giao thức PPP để kết nối với mạng Internet (ppp0). Khi đó, chúng ta có thể xem các kết nối qua eth0 là đáng tin (Trusted devices = eth0) còn giao tiếp ppp0 là cần kiểm soát.

Chúng ta cũng có thể thông qua tùy chọn “Allow Incoming” để quyết định có nên giới hạn hay không giới hạn các dịch vụ.

Chú ý rằng nếu ở RedHat 6.x, sau khi cài tự động, chúng ta có thể sử dụng ngay máy của mình làm một máy chủ Mail, Web,... Còn ở RedHat 7.x, do có Bức tường lửa cho nên các kết nối từ bên ngoài sẽ bị cấm và chúng ta cần cấu hình lại Bức tường lửa.

2.3. Các thiết lập khác

2.3.1. Hỗ trợ ngôn ngữ

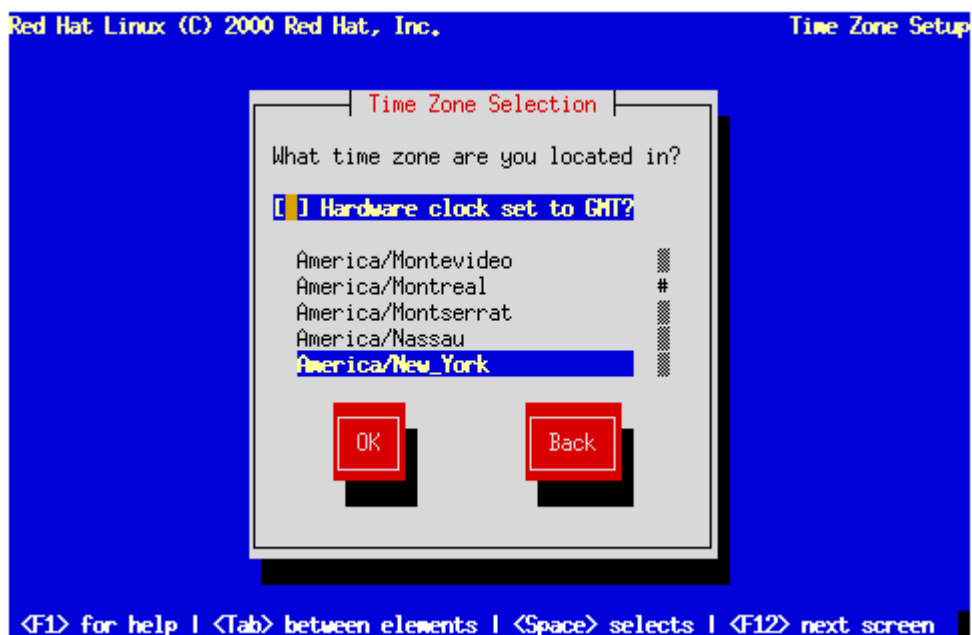
RedHat Linux cho phép cài đặt và hỗ trợ nhiều ngôn ngữ trong hệ thống của chúng ta. Chúng ta có quyền chọn cùng lúc nhiều ngôn ngữ nhưng phải quy định một trong số đó làm ngôn ngữ mặc định. Thông thường ngôn ngữ mặc định là ngôn ngữ chúng ta chọn lúc bắt đầu cài đặt, tuy nhiên khi chọn thêm nhiều ngôn ngữ thì chúng ta vẫn sẽ có khả năng thay đổi ngôn ngữ mặc định sau khi cài đặt (minh hoạ 3.19).

Việc chỉ cài một ngôn ngữ sẽ tiết kiệm một lượng đáng kể khoảng trống của đĩa cứng, nhưng chúng ta cũng chỉ có thể dùng duy nhất ngôn ngữ đó mà thôi.

2.3.2. Thiết lập cấu hình thời gian

Qua hộp thoại “Time Zone Configuration” (minh hoạ 3.20), chúng ta có thể tự quy định sẽ sử dụng giờ địa phương hay giờ GMT. Chúng ta sẽ chọn múi giờ bằng cách dùng chuột di chuyển con chạy trên bản đồ đến thành phố thích hợp và sau khi kích hoạt, một chữ X đỏ sẽ đánh dấu vị trí được chọn đó. Chúng ta cũng có thể chọn bằng cách cuộn thanh cuộn (scroll bar).

Chúng ta vẫn sẽ có thể quy định lại Time Zone sau khi cài đặt bằng cách thực hiện lệnh `/usr/sbin/timeconfig`.

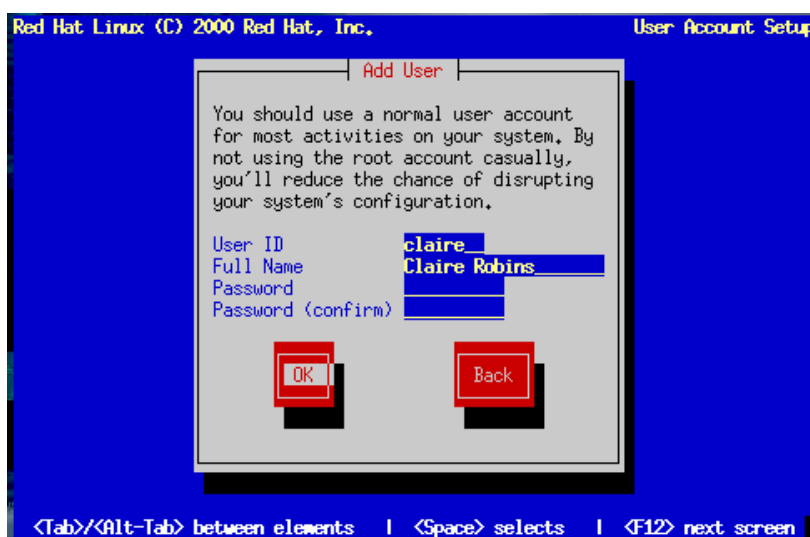


Minh hoạ 3.20: Lựa chọn múi giờ

2.4. Thiết lập trương khoản người dùng

Các hệ điều hành Linux/UNIX quy định một user có quyền hạn tối cao, do đó mang tên Superuser hay còn gọi là root. Vì nằm trong tay những quyền hạn như của quản trị viên hệ thống, superuser có thể thực hiện nhiều việc hay hoặc dở.

Mục “Account Configuration” trước hết cho phép chúng ta thiết lập mật khẩu cho trương khoản root. Mật khẩu của root là chìa khoá cuối cùng cho hệ thống của chúng ta, do đó hãy chọn một mật khẩu dễ nhớ nhưng đảm bảo sự an toàn. Hộp thoại Root Password sẽ mời chúng ta gõ vào hai lần để xác nhận mật khẩu.



Minh hoạ 3.21: Thiết lập trương khoản người dùng

+ Chúng ta có thể tạo tiếp nhiều user theo cửa sổ sau:



Nếu các user quên mật khẩu của họ thì chúng ta còn giúp được bằng quyền hạn của root. Trong đa số trường hợp quên mật khẩu root thì chúng ta buộc phải cài đặt lại toàn bộ hệ thống ; tuy nhiên chúng ta vẫn có khả năng khởi động lại từ đĩa mềm và chỉnh sửa tệp mật khẩu để phục hồi, hoặc khởi động Linux để vào chế độ Single, sau đó sẽ đặt lại mật khẩu mới cho root.

+ Màn hình thiết lập mật khẩu root :

Hộp thoại Root Password buộc bạn phải thiết lập một mật khẩu root cho hệ thống của bạn. Bạn sẽ sử dụng mật khẩu này để log vào hệ thống và thực hiện các chức năng quản trị hệ thống của mình.



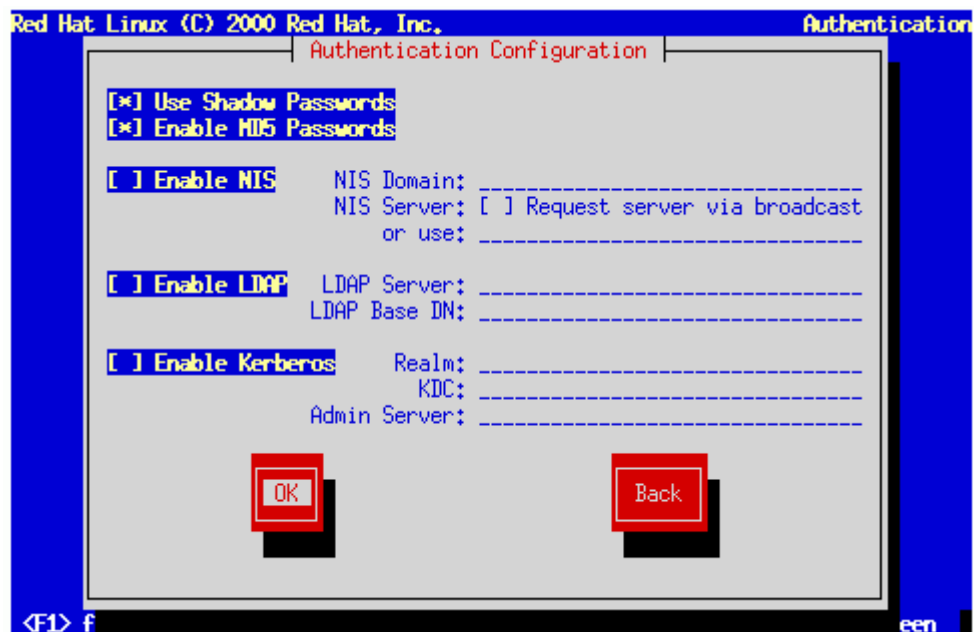
Cũng trong màn hình này, chúng ta có thể tạo các trường khoản khác cho các user cần thiết, bằng cách bấm vào ô <Add>. Một hộp thoại (minh hoạ 3.22) sẽ hiện ra cho phép nhập tên user (User name – 8 ký tự) mật khẩu (password) và nhắc chúng ta khẳng định lại mật khẩu (Confirm).

Chúng ta cũng có thể bỏ qua, không tạo thêm user ngay mà để sau khi cài đặt xong sẽ thực hiện.

2.5. Thiết lập cấu hình xác thực

Màn hình “Authentication Configuration” (cấu hình xác thực) sẽ giúp chúng ta thiết lập quy tắc xác thực trên máy của mình. Nhưng nếu chúng ta đã quy định từ khi cài đặt rằng máy của mình là máy trạm hay máy chủ thì màn hình nói trên sẽ không hiện ra.

Chúng ta có thể chọn sử dụng các tiêu chuẩn chứng thực trên mạng như NIS, LDAP, Kerberos, hay SMB, bằng cách đánh dấu vào ô tương ứng và mở thẻ (tag) liên quan để điền những thông tin thích hợp.



Minh hoạ 3.23: Thiết lập cấu hình xác thực

- Enable MD5 password: cho phép chúng ta tạo các mật khẩu dài (tối đa 256 ký tự) thay cho mật khẩu chuẩn chỉ có 8 ký tự.

- Enable shadow password: cung cấp cho chúng ta một phương pháp lưu trữ mật khẩu an toàn. Mật khẩu thật sẽ không lưu trữ trong tệp /etc/password như các hệ thống chuẩn, mà lưu trữ trong tệp /etc/shadow để chỉ có root mới đọc được.

- Enable NIS: cho phép chúng ta sử dụng chung một mật khẩu trên một nhóm máy tính trong cùng một vùng NIS (Network Information Service). Chúng ta cần cung cấp thêm các thông tin về tên của vùng NIS và tên của một máy chủ

NIS cụ thể. Chúng ta cũng có thể chọn để kích hoạt khả năng phổ biến thông điệp đến mọi máy trong mạng cục bộ hầu tìm một máy chủ NIS thích hợp.

- Enable LDAP (Lightweight Directory Access Protocol): cho phép sử dụng LDAP cho một vài hay toàn bộ giao dịch chứng thực. LDAP là một giao thức truy cập thông tin danh bạ điện tử trong hệ thống mạng. Chúng ta cần cung cấp địa chỉ IP của máy chủ LDAP (LDAP server) trong hệ thống, chọn cách nhận biết thông tin về user qua tên riêng DN (LDAP Base DN – Distinguished Name) và cho phép chúng ta gửi thông tin mật khẩu của user đã được mã hoá đến một máy chủ LDAP trước khi xác thực thông qua TLS (Transfer Layer Security).

- Enable Kerberos: Kerberos là một hệ thống bảo mật cung cấp các dịch vụ chứng thực qua mạng. Có 3 tùy chọn để chúng ta lựa, bao gồm: Realm, KDC, Admin Server.

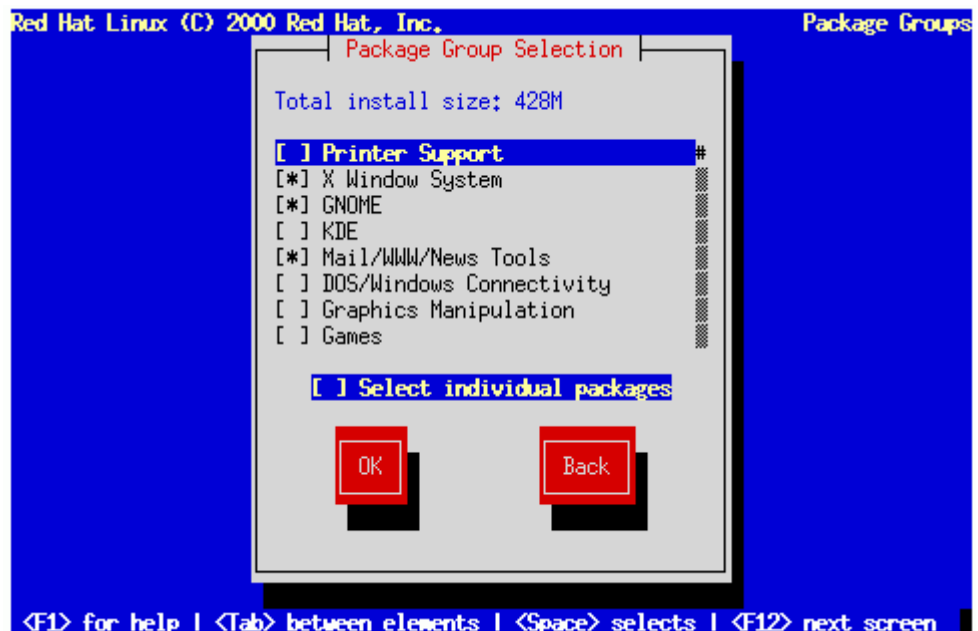
- Enable SMB Authentication: cài đặt PAM để sử dụng một máy chủ SMB cho việc xác nhận các user. Cần cung cấp các thông tin về máy chủ SMB và về Nhóm SMB.

3. Chọn các gói phần mềm & cài đặt

Mục tiêu :

- Trình bày được cách chọn các gói phần mềm để cài đặt
- Nắm vững được các gói phần mềm cài đặt chủ yếu

Đến đây thì hệ thống về căn bản đã sẵn sàng cài đặt. Tuy nhiên chúng ta còn phải chọn những gói phần mềm (package) muốn cài đặt, sau đó thiết lập cấu hình cho chúng.



Minh hoạ 3.24: Chọn các gói phần mềm cài đặt

Chương trình cài đặt sẽ liệt kê trong hộp thoại “Components to Install” các thành tố có thể cài đặt và bảng 3.5 giới thiệu một số gói phần mềm như vậy.

Thành phần	Mô tả
Printer Support	Cho phép in từ hệ Linux
Classic X Window System	Cung cấp môi trường đồ họa GUI cho mọi máy UNIX/Linux, tương tự như MS Windows
X Window System	Cung cấp hệ thống X Window (quản lý cửa sổ đồ họa)
Laptop Support	Cung cấp các ứng dụng chạy trên Laptop và hỗ trợ cho các Laptop chạy Linux
GNOME	Cung cấp các ứng dụng theo giao diện kiểu GNOME
KDE	Cung cấp các ứng dụng theo giao diện kiểu KDE
Sound and Multimedia Support	Cung cấp hỗ trợ âm thanh và multimedia trong môi trường X Window
Network Support	Cung cấp các ứng dụng và công cụ để gỡ lỗi và quản trị mạng, bao gồm các dịch vụ SNMP
Dialup Support	Giúp truy cập Internet qua modem điện thoại
Messaging and Web Tools	Cung cấp chương trình sử dụng thư điện tử, lướt duyệt Web, đọc và đưa tin lên Usenet
Graphics and Image Manipulation	Cung cấp các chương trình đồ họa chẳng hạn như các phần mềm GhostView, Screen Snapshot và GIMP.
News Server	Cho phép hệ thống hoạt động như một server Diễn đàn thảo luận, có thể nhận và phổ biến ý kiến trao đổi của những người sử dụng.
NFS Server	Giúp hệ thống có thêm hai chức năng là xuất khẩu và ghép kèm vào các hệ thống tệp khác trên mạng.
Windows File System	Cung cấp các dịch vụ SMB cho cả client và server MS Windows
Anonymous FTP Server	Cung cấp phần mềm máy chủ FTP vô danh (wu-ftp)
SQL Database Server	Quản trị các Cơ sở dữ liệu dùng SQL (MySQL, PostgreSQL...)
Web Server	Apache, phần mềm Web server rất phổ biến
Router/FireWall	Phần mềm quản lý định tuyến và bức tường lửa.
DNS Name Server	Cung cấp phần mềm cần thiết để chạy server tên miền riêng trên mạng với hệ điều hành Linux
Network Managed Workstation	Cung cấp công cụ quản lý máy trạm
Authoring and Publishing	Cung cấp các ứng dụng cho việc sáng tác và xuất bản

Emacs	Cài đặt phần mềm soạn thảo văn bản emacs, cung cấp X Window front-end cho emacs
Utilities	Cung cấp các công cụ tiện ích
Legacy Application Support	Hỗ trợ các ứng dụng bằng tương thích ngược (kế thừa các ứng dụng cũ)
Software Development	Cung cấp các trình biên dịch và công cụ lập trình (GNU gcc, các trình dịch Python, Perl...)
Kernel Development	Cung cấp các công cụ phát triển kernel
Windows Compatibility/ Interoperability	Cung cấp ứng dụng phỏng tạo/liên tác môi trường MS Windows
Games and Entertainment	Cung cấp các trò chơi chạy dưới X Window hoặc chế độ văn bản.
Everything	Cài đặt tất cả những gì có trên CD: Cần có khoảng 3.0 GB trống, chưa kể khoảng trống dành cho các tệp dữ liệu.

Bảng 3.5: Các gói phần mềm cài đặt chủ yếu

Ghi chú: Chúng ta có thể chọn từng gói phần mềm bằng cách đánh dấu mỗi ô tương ứng trên hộp thoại, hoặc cài đặt toàn bộ khi đánh dấu tùy chọn ấy. Muốn chọn một gói phần mềm để cài đặt, chúng ta chỉ cần di chuyển đến mục mình muốn và nhấn thanh Spacebar. Sau khi chọn xong tất cả những thành tố phần mềm muốn cài đặt, chúng ta bấm OK và bấm <Enter>.

Lưu ý: Sau này chúng ta có thể dùng chương trình RPM được mô tả trong giáo trình để cài đặt các gói phần mềm.

GNOME và KDE là 2 kiểu giao diện GUI của Linux (tương tự giao diện Windows 9x, 2000, XP). Chúng ta có thể chọn cả hai với điều kiện là đĩa cứng của chúng ta đủ lớn (khoảng gần 2 GB).



Minh hoạ 3.25: Chọn chi tiết cài đặt các phần mềm

Sau khi cài đặt, một hộp thoại khác báo cho chúng ta biết nơi chứa thông tin về các tệp sẽ cài đặt là /tmp/install.log. Bấm <Enter> để tiếp tục cài đặt.

Nếu chúng ta đánh dấu vào ô “Select individual package”, chúng ta có thể chọn lựa lại từng gói phần mềm mà chúng ta muốn hay không muốn cài đặt.

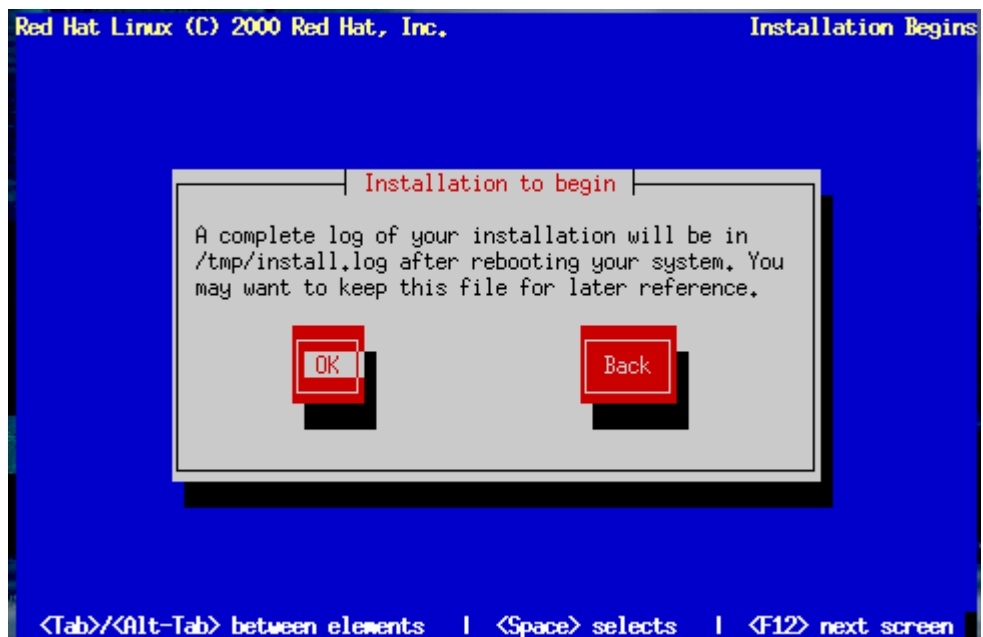
Có thể chọn một trong 2 cách liệt kê các gói phần mềm theo hình cây hoặc dàn ngang (Tree View hoặc Flat View) để dễ theo dõi và lựa chọn. Khi nhấp chuột lên một gói phần mềm bất kỳ, chúng ta sẽ đọc được các giới thiệu về nó ở phần dưới của màn hình. Chúng ta cũng có thể theo dõi mối quan hệ giữa các phần mềm sẽ được cài đặt khi bật chế độ “Dependencies” (liên quan). Nếu chúng ta loại bỏ một số phần mềm liên quan cần thiết cho một số phần mềm khác thì RedHat sẽ thông báo ngay rằng chúng ta cần cài đặt lại các phần mềm đó để bảo đảm các phần mềm đang chọn hoạt động tốt.

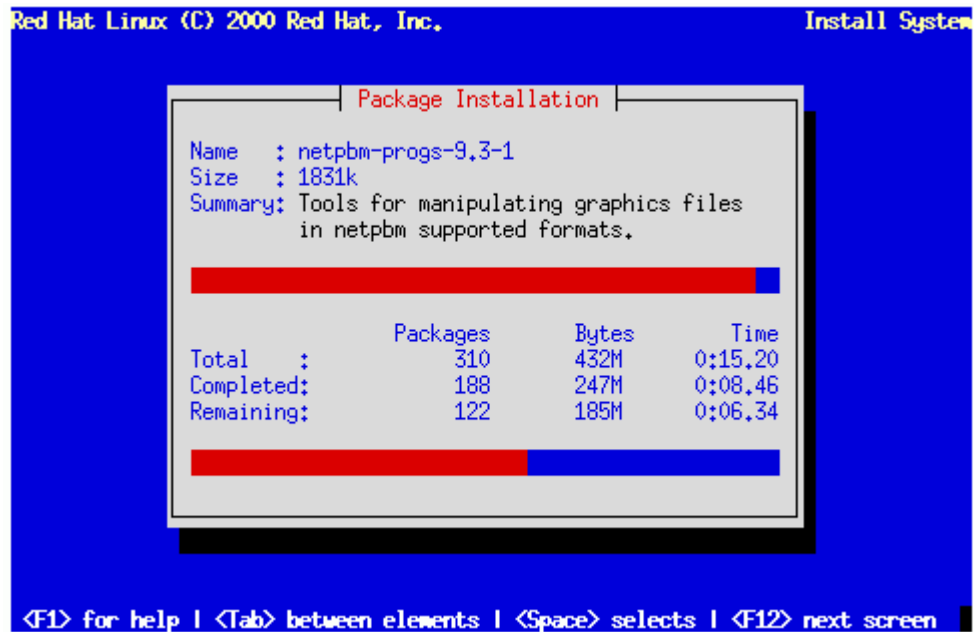
+ Cấu hình Video Adapter :

Chương trình cài đặt sẽ tự phát hiện video card khởi tạo. Nhấn OK để tiếp tục



+ Bắt đầu khởi tạo các gói tin:
Quá trình khởi tạo sẽ được ghi vào tệp /tmp/install.log. Nhấn OK để tiếp tục.





Tạo đĩa khởi tạo cho hệ thống (boot disk): Chọn No và tiếp tục.



+ Hoàn thành cài đặt :

Như vậy là bạn đã hoàn thành xong công việc cài đặt hệ điều hành RedHat 7.0. Bạn hãy rút đĩa ra khỏi ổ CD và nhấn OK để khởi động lại hệ thống.



4. Thiết lập cấu hình & Kiểm tra cấu hình X Window

Mục tiêu :

- Thiết lập được cấu hình & Kiểm tra cấu hình X Window

4.1. Thiết lập cấu hình X Window

Lưu ý: Cố gắng chọn bìa điều khiển màn hình cho thật phù hợp, bởi vì trong số những thiết bị ngoại vi thì bìa điều khiển màn hình và màn hình là những thứ mà phần mềm có thể dễ dàng làm hỏng nhất. Nếu chọn sai loại bìa điều khiển màn hình, màn hình của chúng ta có rủi ro bị cháy.

Đến đây, hệ thống sẽ cài đặt server XFree86 thích hợp cho phần cứng máy chúng ta.

Ở bước tiếp theo màn hình sẽ hỏi chúng ta về các chip đồng hồ (clockchip) trên bìa điều khiển màn hình. Các chip này dùng để điều khiển tín hiệu video thông qua bìa màn hình. Nếu không được đồng bộ hoá, các tín hiệu có khả năng làm cháy màn hình của chúng ta. Chúng ta phải cung cấp tham số chính xác và nếu không có những thông tin ấy chúng ta nên dùng tham số mặc định cho Clockchip, nghĩa là No Clockchip Setting, sau đó bấm OK.

Tiếp theo hệ thống sẽ tự động trắc nghiệm (autoprobe) và thiết lập cấu hình X Window. Máy chúng ta có thể bị treo trong tiến trình autoprobe. Nếu máy treo, chúng ta chỉ cần khởi động lại máy và tiếp tục cài đặt. Chúng ta có thể bỏ qua phần tự động trắc nghiệm và tiếp tục cài đặt.

Trường hợp autoprobe thành công, máy sẽ yêu cầu chúng ta chọn lựa độ phân giải. Chúng ta có thể chọn lựa nhiều độ phân giải tùy vào khả năng xử lý của bìa điều khiển màn hình và màn hình. Cuối cùng chương trình cài đặt sẽ hướng dẫn chúng ta cách khởi động và đình chỉ hệ thống X Window.

4.2. Kiểm tra cấu hình X Window

Như đã giới thiệu, các hệ Linux và UNIX đời mới đã có một giao diện đồ họa gọi là X Window, gọi tắt là X.

Việc cấu hình giao diện X để chúng ta làm việc thoải mái như trong môi trường MS Windows lại không dễ dàng lắm. Chúng ta cần nắm được chính xác các tham số về hệ thống đồ họa của mình, bao gồm: bìa đồ họa (Video Card), màn hình (Monitor), hoặc các tần số tín hiệu đồng bộ dọc/ngang (Vertical/Horizontal Sync). Các tham số này thường có sẵn trong tài liệu đi kèm màn hình của chúng ta.

Trong trường hợp không thấy loại màn hình của mình có tên trong danh sách, chúng ta nên chọn loại màn hình “phổ quát” (Generic) tương đương và chỉnh lại tần số đồng bộ dọc/ngang.

Chúng ta cũng cần chọn thêm các tham số về độ phân giải kích thước và độ phân giải màu trong màn hình “Custom configuration”. Khi đó, chúng ta nên chạy kiểm tra (**test**) để xem cấu hình có hoạt động được không. Nếu máy bị treo, nhiều lúc chúng ta có thể dùng tổ hợp 3 phím Ctrl+Alt+Backspace để thoát khỏi chế độ kiểm tra. Tuy nhiên không phải lúc nào cũng thoát được.

Chúng ta cũng cần chọn giao diện GNOME hoặc KDE làm giao diện mặc định.

5. Tạo đĩa mềm khởi động & khởi động lại

Mục tiêu :

- Tạo được đĩa mềm khởi động và khởi động lại

5.1. Tạo đĩa mềm khởi động

Để tạo đĩa mềm khởi động, chúng ta cần chuẩn bị sẵn một đĩa mềm trắng và đưa nó vào trong ổ đĩa mềm của máy tính.

Đây là một việc rất cần thiết. Đĩa mềm này sẽ giúp chúng ta khởi động hệ Linux của mình khi mà các chương trình GRUB, LILO và các chương trình quản lý khởi động của các nhà sản xuất khác không hoạt động được.

Chúng ta cũng có thể tạo đĩa khởi động sau khi cài đặt xong, nghĩa là chúng ta có thể bỏ qua bước này.

5.2. Khởi động lại

Sau khi hoàn tất phần thiết lập và đặt xong cấu hình cho hệ thống Linux, chúng ta cần khởi động lại. Việc này do chương trình cài đặt tự thông báo, trước hết chúng ta chỉ cần lấy ra các đĩa mềm và đĩa CD đang còn nằm trong máy mình.

Khởi động lại Linux không giống như khởi động lại DOS bằng cách tắt rồi mở công tắc điện. Nếu làm như thế với Linux, hệ thống và kết cấu tệp có thể bị hỏng, vì vậy trong khi chạy Linux chúng ta đừng đột ngột tắt điện.

6. Cài đặt Linux ở chế độ văn bản

Mục tiêu :

- Cài đặt được Linux ở chế độ văn bản từ đĩa CD

6.1. Cấu hình phần cứng cơ bản

Việc cơ bản cần chuẩn bị là nắm rõ các tham số phần cứng của máy tính:

- **Đĩa cứng:** các tham số về loại, kích thước, kiểu đĩa. Chúng ta cần phân biệt rõ các đĩa cứng đã có trong máy đang được cài đặt như đĩa chính (Master) hay đĩa phụ (Slave) và với giao diện nào: IDE 0, IDE 1 hay SCSI...

- **Bộ nhớ:** tổng lượng RAM đã cài trên máy tính.

- **Ổ CD:** loại ổ CD với giao diện IDE, SCSI hay một loại khác.

- **SCSI:** tên của bìa giao diện SCSI và đời (model) của nó.

- **NIC (Network Interface Card – bìa giao diện mạng):** tên nhà sản xuất và đời. Ngoài ra cũng cần biết một số chi tiết về cấu hình mạng của nơi chúng ta đặt máy. Trong trường hợp máy không nối mạng, chúng ta có thể tự chọn một cấu hình nào đó hợp lý.

- **Chuột:** loại chuột (3 phím hay 2 phím) và cổng giao tiếp (PS/2, cổng COMx, USB)...

Thông thường chương trình cài đặt có thể nhận ra được các phần cứng trong máy, nhưng việc chuẩn bị sẵn các thông tin trên sẽ giúp chúng ta điều chỉnh lại các nhận biết tự động của chương trình cài đặt.

6.2. Các màn hình ở chế độ văn bản

Dù Linux đang chạy ở chế độ văn bản, các cấu hình mà chúng ta cần cài đặt được hiển thị trên các màn hình có giao diện trực quan và cũng gần giống các màn hình sử dụng môi trường đồ họa.

6.3. Dùng bàn phím để di chuyển

Nên chú ý rằng trong các màn hình ở chế độ văn bản, chúng ta chỉ sử dụng bàn phím để di chuyển từ phím nhấn này đến phím nhấn khác nhằm chọn cấu hình hay hành động thích hợp. Chúng ta sẽ sử dụng các phím mũi tên, Tab và Alt+Tab để di chuyển. Cần thật cẩn thận không bấm nhầm phím.

6.4. Cài đặt ở chế độ văn bản từ đĩa CD

Chúng ta có thể cài đặt chế độ văn bản từ đĩa CD nếu máy của chúng ta hỗ trợ việc này và chọn bất kỳ một phím chức năng nào đó để chấm dứt việc cài đặt chế độ đồ họa như mặc định. Chúng ta chọn tiếp:

Boot: text

Hay :

boot: text expert

để bắt đầu.

Trong một vài trường hợp, chúng ta có thể cần nhập thêm tham số cho việc khởi động:

boot: text mem=128M

Vì theo mặc định. Linux thường chỉ tận dụng được 64 MB RAM (trong đa số trường hợp với các bản RedHat 7.0 trở về trước).

B. CÂU HỎI VÀ BÀI TẬP

Câu 1: Trình bày cách cài đặt và trình tự cài đặt Linux?

Câu 2: Trình bày cách thiết lập các cấu hình cho RedHat?

Câu 3: Nêu cách tạo đĩa mềm khởi động và khởi động lại?

Câu 4: Trình bày cách cài đặt Linux ở chế độ văn bản?

BÀI 4: BẮT ĐẦU SỬ DỤNG LINUX

Mã bài: MĐ37-04

❖ Giới thiệu

Bài này sẽ giúp các em học sinh thiết lập trương khoản người sử dụng (user account), giới thiệu một vài lệnh căn bản và tiện ích của hệ điều hành Linux mà các em mới cài đặt xong lên máy PC. Chúng ta có một hệ điều hành đa nhiệm và đa người dùng của riêng mình nên cứ mạnh dạn bắt tay vào thao tác để có kinh nghiệm thực tế, vì có thể các em không được gặp dịp may như thế trên những hệ UNIX thuần túy. Ngoài ra, với Internet, chúng sẽ có cơ hội tải về hàng ngàn ứng dụng nguồn mở và miễn phí từ thế giới Linux.

❖ Mục tiêu

- Thiết lập tài khoản.
- Quản lý người sử dụng.
- Những lệnh căn bản.
- Làm việc với các tệp DOS.
- Đóng chương trình Linux.
- Chạy các chương trình Linux.
- Chơi trò chơi trên Linux.
- Chạy các chương trình DOS trên Linux.
- Chạy các chương trình Windows trên Linux.
- Tự tin trong việc sử dụng Hệ điều hành Linux

❖ Nội dung chính

A. LÝ THUYẾT

1. Thiết lập tài khoản

Mục tiêu :

- Nêu được các bước thiết lập tài khoản
- Thiết lập được một tài khoản bằng các dòng lệnh

Việc khởi động Linux mất gần một phút, cuối cùng dấu nhắc hiện lên màn hình và mời chúng ta đăng nhập vào hệ thống. Toàn bộ thông báo đại loại có thể như sau:

RedHat Linux Release 7.3 (Valhala)

Kernel 2.4.18-3 on an I686

Web login:

Chúng ta sẽ thấy các số hiệu khác nhau tùy theo phiên bản Linux được cài đặt.

Đến đây chúng ta phải nhập vào tên người sử dụng (user name) và mật khẩu (password). Trường khoản người sử dụng giúp Linux phân biệt chúng ta với nhiều người khác mà nó phải phục vụ cùng lúc hoặc ở các thời điểm khác nhau.

Linux chấp nhận nhiều trường khoản, mỗi trường khoản cung cấp cho từng người sử dụng một thư mục mặc định, gọi là thư mục “nhà” (home directory).

Việc thiết lập trường khoản làm cho người sử dụng chỉ được thao tác trong phạm vi vài thư mục nào đó của hệ thống và với một số các câu lệnh nhất định mà thôi, bởi vì mục tiêu đầu tiên của trường khoản là để bảo vệ sự riêng tư của từng cá nhân.

1.1. Giao tiếp qua dòng lệnh

Chúng ta nhập dòng lệnh cho Linux cũng giống như cho DOS hoặc cho những hệ điều hành giao tiếp qua dòng lệnh với người dùng. Linux đòi hỏi sự chính xác với từng ký tự trong câu lệnh, kể cả việc phân biệt chữ thường với chữ hoa.

Nếu xảy ra trường hợp Linux không hiểu một câu lệnh nào đó, chúng ta nên kiểm tra xem mình viết đúng hay chưa. Đa số các câu lệnh sẽ được thực thi ngay sau khi chúng ta bấm phím <Enter>.

1.2. Lịch trình nhập lệnh

Nhiều shell Linux có lệnh history để xem lại lịch trình các câu lệnh được gõ vào. Lịch trình nếu không bị xoá sẽ được dùng như một thứ nhật ký của phiên sử dụng. Chúng ta có thể bấm phím < > để hiện lịch trình, rồi di chuyển con chạy trong đó và bấm <Enter> để kích hoạt một lệnh tự chọn mà không phải gõ lại nó.

Trong thí dụ sau, người sử dụng tên là Lan Anh dùng lệnh history và màn hình hiển thị một lịch trình các câu lệnh đã nhập, đại loại như:

```
[lan_anh@web~]$history
1 clear
2 adduser
3 history
```

Sau khi đã có lịch trình, chúng ta có thể chọn lại một lệnh trong đó bằng cách bấm phím < > và lướt con chạy cho đến khi gặp lệnh thích hợp. Hoặc chúng ta có thể bấm phím < ! > rồi nhập số của lệnh mà chúng ta muốn Linux thi hành lại. Thí dụ khi muốn máy thi hành lại lệnh **adduser** trong lịch trình trên, chúng ta gõ như sau:

```
[lan_anh@web~]$!2
```

Chúng ta có thể định nghĩa số của từng dòng trong lịch trình ra lệnh tại tệp cấu hình profile ở trường khoản của người sử dụng.

Ghi chú: Linux có nhiều loại shell và một số shell không có tiện ích lịch trình.

1.3. Nhập lệnh bằng sao ghép

Nếu chúng ta đã cài đặt chuột và một chương trình tiện ích mang tên “selection”, chúng ta có thể sao vài đoạn chữ từ các vùng khác nhau của màn hình để ghép thành một câu lệnh.

Muốn chọn một đoạn chữ, chúng ta bấm và giữ phím trái của chuột rồi kéo lê con chạy đi hết đoạn chữ (để nó đổi màu thành âm bản), sau đó bấm phím phải của chuột để sao đoạn chữ sang dòng lệnh.

Thao tác này rất có ích khi chúng ta muốn nhập một câu lệnh dài.

1.4. Tự động điền lệnh

Linux còn một cách khác để nhập lệnh. Chúng ta gõ vài ký tự đầu của tên lệnh rồi bấm phím <Tab>. Linux sẽ tìm trong thư mục lệnh xem tệp nào có tên bắt đầu giống như những ký tự mà chúng ta vừa gõ vào, rồi tự động điền đủ tên tệp vừa được phát hiện. Trong trường hợp có vài lệnh mang các ký tự khởi đầu giống nhau, Linux sẽ phát ra âm thanh “bip” và tự động viết đủ tên lệnh cho đến ký tự chung cuối cùng mà những lệnh đó đều có.

Thí dụ chúng ta muốn chép một lệnh mang tên `todo_Monday` sang tệp `todo_today`. Chúng ta gõ `cp to` tại dấu nhắc rồi bấm <Tab>, Linux sẽ bip và tự động viết đủ phần chung của tên tệp tại dòng lệnh như sau:

```
[lan_anh@web~]$sp todo_
```

Nếu vào lúc này chúng ta gõ `M` (nghĩa là ký tự đầu tiên của chữ `Monday`) và bấm <Tab>, Linux sẽ tự động điền đủ `todo_Monday` vào dòng lệnh.

2. Quản lý người sử dụng

Mục tiêu :

- Đăng nhập và đăng xuất được một user
- Thêm người sử dụng trong Slackware
- Thêm người sử dụng mới trong RedHat Linux

Người chịu trách nhiệm gìn giữ các trương khoản của người sử dụng trong hệ thống được gọi là quản giá trị viên hệ thống. Quản giá trị viên hệ thống có nhiệm vụ thiết lập các trương khoản và một số các công việc khác mà chúng ta có thể tìm hiểu tại phần “Quản giá trị hệ thống” của giáo trình này.

Trên hệ Linux của chúng ta, chúng ta là quản giá trị viên hệ thống, do đó chúng ta chịu trách nhiệm thiết lập trương khoản cho chính mình, cho gia đình và bè chúng ta.

Muốn thêm vào trương khoản cho chính mình, chúng ta phải đóng vai trò quản giá trị viên hệ thống để tạo nó ra. Đôi lúc quản giá trị viên hệ thống còn được gọi là superuser bởi vì người này nắm quyền kiểm soát trên toàn bộ hệ thống. Để bắt đầu hành trình Linux, thoạt tiên chúng ta phải đăng nhập với tư cách là superuser qua trương khoản root.

2.1. Đăng nhập và đăng xuất

Muốn đăng nhập (log in) với quyền hạn của superuser, chúng ta gõ `root` tại dấu nhắc đăng nhập và Linux sẽ hỏi mật khẩu.

Mật khẩu ngăn cấm những người không thẩm quyền lên đăng nhập vào những trường khoản mà họ không có quyền dùng. Linux bảo vệ mật khẩu chúng ta gõ vào bằng cách không cho hiện lại (echo) các ký tự, nghĩa là không hiển thị những gì đang gõ vào, do đó chúng ta cố gắng gõ đúng mật khẩu.

Nếu gõ vào tên hoặc mật khẩu không hợp lệ, Linux sẽ báo lỗi như sau:
 web login: lan_anh
 Password: password
 Login incorrect
 web login:

Để đăng xuất (log out), chúng ta gõ logout. Lệnh này sẽ đưa chúng ta trở về dấu nhắc đăng nhập. Nếu lệnh này không hoạt động, chúng ta cần gõ lệnh **exit**.

2.2. Thêm người sử dụng trong Slackware

Trong Linux Slackware, sau khi đăng nhập với tư cách là root, chúng ta bổ sung người sử dụng mới vào hệ thống hiện hành bằng cách gõ lệnh **adduser**:

```
[root@web~] # adduser
```

Adding a new user. The user name should be not exceed 8 characters in length, or you may run into problems later.
 Enter login name **for** new account (^C to quit):

Chúng ta cần chú ý đến dấu nhắc, tức là chỗ mà ngay sau nó chúng ta phải gõ lệnh vào. Dấu nhắc bắt đầu bằng host name là cái tên mà chúng ta đã chọn trong khi cài đặt hệ thống và tiếp theo là ký tự ~ tức dấu sóng (tilde), kết thúc bằng dấu thăng #. Linux sử dụng ký tự ~ để biểu thị thư mục nhà (/home) của trường khoản hiện hành (sẽ giải thích ở phần sau). Tại thời điểm này host name là [root@web](#) và dấu ~ tượng trưng cho thư mục hiện hành, nghĩa là chúng ta đang ở trong thư mục ấy. Nếu chúng ta ra lệnh **adduser** từ thư mục /usr/bin, thì dấu nhắc sẽ có dạng:

```
[root@web /usr/bin] #
```

Ký tự cuối là dấu thăng (#), được Linux quy ước dùng để chỉ trường khoản của superuser. Các user bình thường thì có ký tự dollar (\$) ở cuối dấu nhắc.

Tiếp theo, chúng ta có thể thấy cảnh báo nếu có sai chính tả và ngữ pháp tại các dấu nhắc ("should be not" hoặc "you may run" ...). Những sai sót này không ảnh hưởng đến an toàn máy; chỉ nhắc chúng ta nhớ rằng Linux là một hệ thống chạy được và chạy tốt, song cũng có khác các thương phẩm.

Chúng ta hãy gõ vào tên một người sử dụng dài không quá tám (8) ký tự và bấm <Enter>. Sau đây là một thí dụ khi cô Lan Anh tạo lập trường khoản:

```
Enter login name for new account (^C to quit): lan_anh
Editing information for new user [lan_anh]
Full Name: Phan Lan Anh
GID[100]: <Enter>
Checking for an available UID after 500
501...
```

```

First unused uid is 502
UID[502]:<Enter>
Home directory: [/home/lan_anh] :<Enter>
Shell [/bin/bash]:<Enter>
Password: xxxxxxxx
Information for new user [lan_anh]:
Home directory: [/home/lan_anh] Shell: [/bin/bash]
Password: [xxxxxxx] uid: [502] gid: [100]
Is this correct? [y/N]:y
Adding login [lan_anh] and making directory [/home/lan_anh]
Adding the files from the /etc/skel directory:
./kermc -> /home/lan_anh/./kermc
./less -> /home/lan_anh/./less
./lessrc -> /home/lan_anh/./lessrc
./term/ -> /home/lan_anh/./term
./term/.termrc ->/home/lan_anh/./termrc
./emacs -> /home/lan_anh/./emacs
[root@web~] #

```

Ghi chú: Chúng ta cần gõ tên đầy đủ của người sử dụng để về sau nhận diện được trương khoản ấy, rồi nhập số định danh nhóm (group ID) và số định danh người sử dụng (user ID), nhưng vào thời điểm này chúng ta đừng lo âu về những chi tiết đó. Linux dùng chúng để xác định các thư mục và các tệp mà chúng ta có quyền truy cập. Chúng ta có thể chấp nhận các giá trị mặc định (nằm bên trong ngoặc vuông) bằng cách bấm phím <Enter> sau mỗi lần máy hỏi.

Tiếp theo Linux nhắc chúng ta nhập tên home directory cho người sử dụng, tức thư mục sẽ dành cho người sử dụng khi đăng nhập. Đây là vùng để trương khoản của người sử dụng ấy dùng để lưu các tệp và làm việc. Linux cung cấp một thư mục mặc định dựa vào tên của người sử dụng. Nếu thư mục mặc định ấy được chúng ta chấp nhận thì bấm <Enter>. Nếu không, chúng ta gõ vào một tên khác và bấm <Enter>. Tạm thời chúng ta nên chấp nhận các mặc định do lệnh **adduser** đề nghị.

Đến đây Linux yêu cầu chúng ta xác định loại shell cho người sử dụng. Shell là một chương trình diễn dịch (interpreter) tiếp nhận dòng lệnh được gõ vào và thực hiện một số lệnh nhất định. Kể từ đầu đến giờ, chúng ta đang dùng một shell gọi là bash. Vào thời điểm này, chúng ta chỉ cần chấp nhận tùy chọn mặc định bash.

Tham số quan trọng cuối cùng là mật khẩu của trương khoản. Tốt nhất, đối với mỗi trương khoản chúng ta nên có một mật khẩu khác nhau.

Tiếp theo, Linux sẽ hiển thị tất cả mọi thông tin được nhập vào và hỏi chúng ta xem đã chính xác chưa. Nếu chưa, chúng ta gõ n (hoặc bấm <Enter>, bởi

vì No là lựa chọn mặc định) sau đó trở lại chỉnh sửa những chỗ cần thiết. Chỉ khi mọi thứ đã chính xác, chúng ta mới gõ y.

Linux hiển thị một loạt tệp chép từ thư mục /etc/skel sang thư mục “nhà” của người sử dụng. Đây là những tệp cấu hình cho một số mục như terminal của người sử dụng và cách hoạt động của một số chương trình chẳng hạn như **emacs** và **less**.

Sau khi thêm trường khoản vào hệ thống, chúng ta kiểm tra sự hiện diện của nó bằng một trong hai cách. Cách nhanh nhất là dùng tiện ích **finger** để xem người sử dụng có trường khoản hay chưa. Dạng tổng quát của câu lệnh này là :

```
finger tên_user
Thí dụ:
[root@web~] # finger lan_anh
Login: lan_anh
Name: Phan Lan Anh
Directory: /home/lan_anh
Shell:/bin/bash
Never logged in.
No Mail.
No Plan.
[root@web~] #
```

Nếu người sử dụng có một trường khoản, Linux sẽ hiển thị tình trạng ấy; nếu không thì sẽ thông báo là không có trường khoản.

Cách kiểm tra thứ hai một trường khoản là đăng nhập thật sự vào trường khoản ấy để thử xem Linux có cho vào hay không. Chúng ta có thể thử bằng nhiều cách:

- Đăng xuất sau đó lại đăng nhập bằng tên người sử dụng mới.
- Sử dụng lệnh **su**.
- Dùng lệnh **login**.
- Sử dụng một trong sáu terminal ảo do Linux cung cấp để đăng nhập vào trường khoản mới (nên nhớ là Linux có đặc tính đa người dùng).

Lệnh	Mô tả
Logout	Đăng xuất khỏi trường khoản đang dùng và trở về dấu nhắc đăng nhập. Không thể truy cập vào trường khoản nói trên nếu không đăng nhập lại.
su tên_user	Chuyển sang trường khoản của người khác và sẽ nhập mật khẩu của trường khoản mới này. Nếu không khai tên_user, thì mặc định là root. Người sử dụng cũ không bị đăng xuất và vẫn ở trường khoản cũ.
Login tên_user	Cũng là chuyển người sử dụng, nhưng người sử dụng cũ bị đăng xuất. Nếu để trống tên_user thì hệ thống sẽ trở về dấu nhắc đăng nhập bình thường.

<Alt-Fx>	Cho phép sử dụng 1 trong 6 terminal ảo bằng cách bấm <Alt> và một phím chức năng từ F1 đến F6. Sẽ hiển thị một màn hình để có thể đăng nhập trường khoản mới. Tức là dù ở một trường khoản song vẫn có thể vào trường khoản khác và tổ hợp <Alt-Fx> giúp di chuyển qua lại giữa hai trường khoản ấy.
----------	--

Bảng 4.1: Các cách kiểm tra một trường khoản

Ghi chú: Nếu định thêm người sử dụng mới vào một trường khoản mà chúng ta đã tạo ra từ trước và không phải là root, chúng ta sẽ không có quyền sử dụng lệnh **adduser**, một câu lệnh chỉ dành riêng cho superuser. Khi đó chúng ta hãy đăng nhập lại với tư cách root.

2.3. Thêm người sử dụng mới trong RedHat Linux

Phiên bản RedHat Linux có tự động hoá chức năng tạo thêm người sử dụng mới; từ dòng lệnh, chúng ta gõ:

```
[root@web/root] #adduser lan_anh
```

Thực chất, lệnh trên là một shell script nằm ở /usr/sbin; muốn ra lệnh đó, chúng ta phải là superuser. Giá trị của các thuộc tính sẽ có của người sử dụng mới đã được khai báo sẵn trong tệp /etc/default/useradd hay /home/etc/default/useradd.

Trình script trên chỉ là một tệp văn bản, nó tạo ra các thư mục và tệp cần thiết cho người sử dụng mới. Điều còn lại phải làm là thiết lập mật khẩu người sử dụng khi người ấy đăng nhập. Việc thay đổi mật khẩu sẽ được bàn tới tại mục “Thay đổi mật khẩu”.

2.4. Dùng bảng điều khiển RedHat để quản lý người sử dụng

Nếu cài đặt XFree86 chung với RedHat Linux, chúng ta có thể mở cửa sổ thiết lập cấu hình Quản lý người sử dụng và nhóm người sử dụng (User/Group Manager) trong bảng điều khiển (Control Panel) như minh hoạ 5.2 để thêm người sử dụng mới, xoá bỏ hoặc cho nghỉ một người sử dụng, hoặc thay đổi những thiết lập cho một người sử dụng. Chúng ta có thể dùng lệnh sau : từ cửa sổ shell (hay xterm, hay Run), gõ:

```
redhat-config-users
```

Muốn thao tác một trường khoản của người sử dụng, chúng ta chọn người sử dụng từ hộp thoại sau đó chọn nút thích hợp. Bảng 5.2 mô tả chức năng từng nút.

Nút	Mô tả
Add	Hiện hộp thoại Add User, giúp thiết lập các thuộc tính từng người sử dụng, chẳng hạn như thư mục “nhà” và mật khẩu.
Deactivate	Tạm treo trường khoản của một user để sau này còn sử dụng. Lý do treo có thể là user ấy nghỉ phép hoặc bị kỷ luật. Có thể nén các tệp của user ấy để tiết kiệm chỗ trống trên ổ đĩa cứng và chờ đến khi tái kích hoạt.
Reactivate	Tái kích hoạt trường khoản của người sử dụng.

Remove	Xoá một user ra khỏi hệ thống. Các thư mục và tệp của user ấy sẽ bị xoá sạch. Linux cho phép sao lưu chúng trước khi xoá hẳn.
Edit	Chỉnh sửa các chi tiết trương khoản của user như mật khẩu (trong trường hợp quên mật khẩu), shell và nhóm của user.
Exit	Thoát khỏi RH Linux User/Group Manager.

Bảng 4.2: Các nút quản lý user/group của RH Linux

Nhấp chuột vào nút Add sẽ hiển thị hộp thoại Add User như ở minh hoạ 5.2, chúng ta thiết lập cấu hình trương khoản cho từng người sử dụng bằng cách điền vào các trường của hộp thoại. Bảng sau mô tả các trường cùng với chức năng tương ứng.

Trường	Mô tả
Username	Tên của người sử dụng dùng để đăng nhập hệ thống
Password	Mật khẩu của người sử dụng đang đăng nhập. Muốn phân phối mật khẩu người sử dụng, phải dùng lệnh Edit từ menu; màn hình sẽ hiển thị một hộp thoại khác để nhập mật khẩu mới cho người sử dụng. Tổ hợp này cũng cho phép bỏ trống trường mật khẩu bằng cách chọn giá trị “none” hoặc khoá mật khẩu.
UID	Đây là một trường do hệ thống sinh ra. Xem thêm chi tiết tại chương “Quản lý trương khoản người sử dụng”.
Primary Group	Nhóm sơ cấp người sử dụng. Trường này giúp đưa những người sử dụng vào thành từng nhóm, với từng quyền hạn riêng biệt.
Full Name	Tên họ đầy đủ của người sử dụng.
Home	Thư mục “nhà” của người sử dụng. Thông thường vị trí này ở bên trong thư mục /home.
Shell	Shell mặc định, vị trí làm việc ban đầu của một trương khoản. Một hộp thoại cho phép chọn bất kỳ shell nào mà RedHat Linux sẵn có.

Bảng 4.3: Các tùy chọn của hộp thoại Add User

2.5. Thay đổi mật khẩu

Chúng ta thường phải đổi mật khẩu (hoặc đặt mật khẩu cho một trương khoản mới), nhất là cho trương khoản gốc (root) vì nó cần được bảo vệ thật cẩn thận. Để thay đổi mật khẩu, bất kỳ phiên bản Linux hoặc UNIX nào đều dùng lệnh passwd, nó đòi nhập cả mật khẩu cũ lẫn mới và kiểm tra mật khẩu mới vừa nhập vào. Nếu chúng ta chưa có mật khẩu cho trương khoản (thậm chí quên mất mật khẩu), hãy sử dụng lệnh passwd để thay đổi. Việc thay đổi mật khẩu diễn ra như sau:

```
[lan_anh@web~]$passwd lan_anh
Changing password for lan_anh
```

Enter old password: old-password

Enter new password: new-password

Re-type new password: new-password

Nếu chúng ta làm không đúng, Linux sẽ thông báo là mật khẩu chưa được thay. Linux cũng yêu cầu mật khẩu phải bao gồm ít nhất sáu ký tự.

Cẩn thận: Đừng quên mật khẩu! Nếu quên mật khẩu của người sử dụng, chúng ta phải thay đổi thông tin về trường khoản ấy. Nếu quên mật khẩu trường khoản root, chúng ta phải sử dụng đĩa mềm khởi động (được tạo ra trong tiến trình cài đặt) để khởi động lại hệ thống và đổi mật khẩu. Chúng ta cũng có thể thay đổi mật khẩu của root bằng cách khởi động lại hệ thống ở chế độ Single.

Chúng ta có thể đổi mật khẩu trống bằng cách chọn “none” trong hộp thoại RH Add/Edit User, sau đó cho người sử dụng tự lập mật khẩu mới bằng lệnh passwd. Chúng ta cũng có thể chỉnh sửa tệp /etc/passwd và gỡ bỏ mật khẩu đã mã hoá từ bản ghi của người sử dụng (trong trường hợp không có khai báo shadow). Thông thường RedHat xác định số ký tự tối thiểu của password là 6, tuy thế ta có thể thay đổi được, thí dụ ta dùng công cụ linuxconf như ở minh hoạ 5.4.

3. Sử dụng các lệnh cơ bản

Mục tiêu :

- Biết cách dùng man để tìm trợ giúp cho câu lệnh
- Sử dụng được các lệnh can thiệp vào thư mục
- Sử dụng được các lệnh thao tác tệp

Những mục sau đây sẽ hướng dẫn chúng ta sử dụng các lệnh cơ bản để điều khiển hệ thống, trong đó có một số lệnh thực sự là những trình tiện ích mà Linux dùng để mở rộng tập hợp các câu lệnh của mình. Những chương trình ấy nằm trong các thư mục /bin, /sbin và /usr/bin.

3.1. Dùng man để tìm trợ giúp cho câu lệnh

Muốn nhận được trợ giúp trực tiếp cho các lệnh Linux, chúng ta gõ:
man xyz*

Linux sẽ hiển thị từng trang thông tin liên quan đến từng lệnh bắt đầu bằng các ký tự xyz. Chúng ta có thể xem lần lượt các lệnh này bằng cách bấm phím q, khi đó nội dung từng lệnh sẽ hiện ra. Đặc điểm này của **man** chỉ có từ RedHat 7.2.

Nếu chưa biết chắc chắn mình sẽ dùng lệnh nào, chúng ta thử gõ tham số -k và nhập một từ khoá liên quan đến chủ đề chúng ta đang quan tâm. Lúc ấy **man** sẽ tìm trong các tệp trợ giúp (được gọi là trang **man**, hoặc trang manual) có chủ đề chứa từ khoá ấy. Linux cũng cung cấp một **alias** (bí danh) cho lệnh ấy, gọi là **apropos**.

Nếu chúng ta nhập lệnh **man ls**, Linux hiển thị trợ giúp cho lệnh **ls**, bao gồm tất cả các tham số. Lệnh **man -k cls** cung cấp danh sách các câu lệnh có chữ **cls** trong tệp trợ giúp. Lệnh **apropos cls** cũng giống như **man -k cls**.

3.2. Sử dụng các lệnh can thiệp vào thư mục

Cũng giống như những hệ điều hành khác mà chúng ta có dịp dùng qua, Linux có nhiều câu lệnh để tạo ra, xoá bỏ, di chuyển thư mục và hiển thị thông tin của thư mục.

3.2.1. Chuyển đổi thư mục hiện hành bằng lệnh `cd`

Cũng như DOS và các hệ điều hành khác, Linux chứa các tệp trong một cấu trúc cây gọi là thư mục. Chúng ta đi đến một tệp qua đường dẫn từ thư mục gốc bằng ký tự `/`. Do đó tệp cấu hình emacs cho người sử dụng tên lan_anh có thể được xác định như sau:

```
/home/lan_anh/.emacs
```

Nếu trước nay bạn từng quen thuộc với hạn chế của DOS là tám ký tự cho tên tệp và ba ký tự cho cái đuôi, nay chúng ta sẽ thích thú hơn vì Linux không hạn chế số ký tự cho tên tệp.

Linux cũng sử dụng khái niệm về một home directory (thư mục nhà), thư mục này được xác định khi thêm một trương khoản vào hệ thống. Thông thường, thư mục “nhà” của một người sử dụng được xác định bằng ký tự tilde (ký tự sóng ~). Khi người sử dụng muốn chép một tệp từ thư mục hiện hành `/usr/home/lan_anh` vào thư mục “nhà” của mình, chúng ta có thể dùng ký tự tilde thay vì tên của thư mục.

```
cp.emacs ~
```

Muốn di chuyển trong cấu trúc Linux, chúng ta dùng lệnh chuyển thư mục `cd`. Nếu chúng ta gõ `cd` vào mà không kèm tham số nào, Linux đưa chúng ta về thư mục “nhà” của chúng ta. Muốn chuyển từ thư mục này sang thư mục khác, chúng ta dùng lệnh `cd` như khi chúng ta sử dụng DOS, nghĩa là `cd thư_mục_mới`. Linux cũng dùng dấu chấm (`.`) để đại diện thư mục hiện hành và dấu chấm chấm (`..`) đại diện cho thư mục mẹ. Thực ra chính DOS mới phỏng tạo UNIX, chứ UNIX/Linux không phỏng tạo DOS.

Ghi chú: Hãy cẩn thận khi dùng dấu số ngược ở đường dẫn thư mục. Trong khi DOS dùng dấu số ngược xuống (`\`) thì Linux dùng dấu chéo lên (`/`). Dấu chéo xuống được Linux dùng để nối tiếp một câu lệnh trên một dòng khác.

Và ngược lại với DOS, Linux quan tâm đến việc chúng ta phải dùng khoảng trắng khi xác định các tham số `“.”` và `“..”` Linux không hiểu lệnh `cd...`, nhưng nếu chúng ta viết `cd ..` thì Linux sẽ hiểu. Nói tóm lại, Linux cần khoảng trắng ở giữa câu lệnh và tham số.

3.2.2. Liệt kê các tệp và thư mục bằng lệnh `ls`

`ls` viết tắt cho list (danh sách) và Linux dùng lệnh này để liệt kê danh sách tệp. Lệnh này cũng tương tự như `DIR` của DOS. (Linux chấp nhận lệnh `dir` để liệt kê danh sách tệp trong một thư mục). Lệnh `ls` của Linux liệt kê tất cả các tệp chính bằng màu sắc. Theo mặc định, màu xanh lơ biểu thị thư mục và xanh lục biểu thị các chương trình thi hành được. Muốn thay các màu mặc định, chúng ta chỉnh sửa tệp `/etc/DIR_COLORS`.

ls sử dụng nhiều tham số để thay đổi cách liệt kê tệp và loại tệp phải liệt kê. Tham số phổ biến nhất là `-la`, ra lệnh cho máy liệt kê thông tin của tệp theo dạng dài. Lệnh `ls-la` liệt kê tất cả thông tin của từng tệp trong thư mục hiện hành. Lệnh `ls.emacs` liệt kê các tệp bắt đầu bằng `.emacs`, trong khi `ls-l.emacs` liệt kê tất cả thông tin của các tệp bắt đầu bằng `.emacs`.

Tuỳ chọn `-ltar` (được dùng như là `ls-ltar`) liệt kê thông tin giống như lệnh `ls` vừa kể, có khác chăng là các thông tin được trình bày theo thứ tự từ lâu nhất đến mới nhất.

3.2.3. Tạo thư mục mới bằng lệnh `mkdir`

Tương tự như lệnh `MD` của DOS, `mkdir` của Linux tạo thư mục mới và chúng ta cung cấp tên của thư mục ấy như ở thí dụ sau:

```
mkdir backup
```

Ghi chú: Trong trường hợp chúng ta quá quen với lệnh `MD` của DOS và không thích dùng `mkdir`, Linux có một cách để tạo bí danh cho tên các lệnh.

3.2.4. Xoá bỏ thư mục bằng lệnh `rmdir`

Lệnh `rmdir` xoá các thư mục Linux với điều kiện thư mục ấy phải rỗng. Thí dụ nếu thư mục `/backup` có thư mục thứ cấp thì lệnh:

```
rmdir /backup
```

không hoàn thành công tác được.

Lệnh:

```
rmdir /backup/lan_anh/*
```

xoá tất cả các tệp trong thư mục `/backup/lan_anh` và sau đó

```
rmdir/backup/lan_anh
```

mới có thể xoá bỏ thư mục `/backup/lan_anh` bây giờ đã rỗng.

Cẩn thận: Vì lệnh `rmdir` không thể xoá thư mục chưa rỗng, chúng ta có thể dùng tham số `-r` với lệnh `rm`. Thí dụ: `rm -r *` sẽ xoá sạch mọi thứ từ thư mục hiện hành và tất cả các thư mục thứ cấp bên dưới. Do đó chúng ta phải cẩn thận khi ra lệnh này, vì một khi đã xoá là không phục hồi được. Nhớ sao lưu trước.

3.3. Sử dụng các lệnh thao tác tệp

Linux xử lý tệp và thư mục cũng như nhau.

3.3.1. Chép các tệp bằng lệnh `cp`

Lệnh `cp` tương tự như `copy` của DOS. Chúng ta dùng lệnh này để chép một hoặc nhiều tệp từ thư mục này sang thư mục khác. Cú pháp của `cp` như sau:

```
cp tệp_nguồn tệp_đích
```

Chúng ta thay thế hai tham số `tệp_nguồn` `tệp_đích` bằng tên hai tệp mà chúng ta chọn. Nếu muốn giữ nguyên tên tệp, hãy dùng thư mục sẽ chứa tệp thay vào chỗ tham số `tệp_đích`. Ở DOS, chúng ta có thể bỏ trống `tệp_đích` nếu chép về thư mục hiện hành.

Lệnh **cp** lananh1 lananh1.old sẽ chép tệp lananh1 sang một tệp sao lưu mang tên lananh1.old, trong khi lệnh **cp** ~/lananh1.old /backup/lan_anh sẽ chép tệp lananh1.old từ thư mục “nhà” sang thư mục /backup/lan_anh. (Ký tự ~ đại diện cho thư mục “nhà” của người sử dụng).

3.3.2. Chuyển tệp bằng lệnh mv

Lệnh này tương tự như MOVE của DOS để chúng ta di chuyển tệp từ thư mục này sang thư mục khác. Khi ra lệnh di chuyển tệp, nghĩa là chúng ta đã chép tệp ấy sang chỗ mới và xoá tệp ở chỗ cũ. Lệnh **mv** không sao chép tệp.

Cú pháp của lệnh **mv** giống như lệnh **cp**:

mv tệp_nguồn tệp_đích

Lệnh **mv** lananh1 lananh1.old chép tệp lananh1 sang một tệp sao lưu mang tên lananh1.old, sau đó huỷ tệp lananh1 cũ, trong khi lệnh **mv** ~/lananh1.old /backup/lan_anh di chuyển tệp lananh1.old từ thư mục “nhà” sang thư mục /backup/lan_anh.

3.3.3. Xoá tệp bằng lệnh rm

Lệnh này nguy hiểm bởi vì khi đã huỷ thì chúng ta không thể khôi phục tệp được. Do đó để an toàn, chúng ta nên sử dụng hình thức sau đây của lệnh **rm**:

rm -i tên tệp

Tham số -i bắt máy phải hỏi lại người sử dụng xem có thực sự muốn xoá bỏ tệp hay không. Thí dụ lệnh **rm** lananh1 sẽ xoá tệp lananh1, trong khi **rm** -i lananh1 sẽ mời chúng ta khẳng định việc huỷ tệp.

Cẩn thận: Với Linux, một khi tệp bị huỷ thì coi như mất luôn chứ không thể lấy lại được (undelete) như với DOS. Khi xoá một tệp, hy vọng duy nhất của chúng ta là bản sao lưu của tệp ấy.

3.3.4. Hiện thị nội dung tệp bằng lệnh more

Lệnh **more** hiện thị tệp văn bản qua từng màn hình một. Chúng ta có thể xem một tệp văn bản mà không nhất thiết phải dùng đến phần mềm soạn thảo (edit), không cần in tệp ấy ra và cũng không phải tạm dừng thiết bị cuối (terminal) trong khi thiết bị ấy hiện thị tệp. Thí dụ, muốn hiện thị nội dung tệp .emacs, chúng ta gõ lệnh như sau:

more .emacs

Trong các phiên bản trước 6.0 của RH Linux, lệnh **more** khá bất tiện khi không cho chúng ta trở lui xem lại một trang màn hình đã qua. Tuy nhiên các tùy chọn dưới đây mới được bổ sung từ phiên bản 6.x cho phép trở lui k trang màn hình (mặc định là 1 trang): b hay <Ctrl-b>. Lệnh **less** sau đây có thể giải quyết vấn đề này tiện lợi hơn.

Ghi chú: Nếu dùng **more** với một tệp dữ liệu nhị phân, chúng ta có thể bực mình vì chẳng hạn máy sẽ bị treo. Trong trường hợp này, chúng ta thoát ra bằng cách bấm <Ctrl-q> hoặc <Ctrl-s>.

3.3.5. Sử dụng lệnh less

less hiển thị từng màn hình một. Giống như **more**, **less** có thể hiển thị màn hình thông tin của một tệp văn bản, song điểm khác biệt là chúng ta có thể di chuyển tới lui. Chúng ta thử dùng lệnh sau đây để duyệt tệp readme trong thư mục /info:

```
less /info/readme
```

Ghi chú: Ở DOS, chúng ta dùng lệnh CLS để xoá màn hình ; Ở Linux, đó là lệnh **clear**.

4. Xử lý các tệp DOS trong Linux

Mục tiêu :

- Sử dụng được các lệnh để xử lý các tệp DOS trong Linux

Trong khi cài đặt, chúng ta đã có dịp khai báo các partition (phân vùng) DOS để cho Linux hiểu được. Các partition này được đặt ở một thư mục mà chúng ta đã chọn trong khi thiết lập cấu hình, thí dụ như /dos.

Nếu dùng lệnh **cp** để chép các tệp ấy sang đĩa mềm, chúng ta có thể gặp rắc rối vì UNIX và Linux xử lý tệp văn bản khác với DOS, nhất là khi phải xử lý phím xuống dòng. Mặt khác khi muốn tương tác với đĩa mềm, nếu dùng lệnh **cp**, chúng ta cần phải **mount** (lắp ghép) đĩa mềm trước và sau khi thực hiện lệnh sao chép **cp** xong, chúng ta còn phải **unmount** (tháo gỡ) đĩa mềm đó để lấy đĩa mềm ra. Để giải quyết vấn đề này, người ta đã soạn thảo một nhóm chương trình để xử lý các tệp DOS chạy ở môi trường UNIX. Đó là các câu lệnh thuộc họ m-, trong đó có những lệnh như **mcop**y và **mdir**. Lệnh **mcop**y hoạt động như lệnh COPY của DOS và **mdir** liệt kê thư mục. Như chúng ta đã nhận thấy, các lệnh này giống như đồng loại của chúng ở DOS, chỉ khác là chúng bắt đầu bằng ký tự m, do đó mới gọi là họ lệnh m-. Họ lệnh m- là thành phần của phần mềm chương trình **mtools**, vốn là bộ sưu tập các chương trình công cộng (phân phối tự do miễn phí) giúp cho UNIX tương tác với các tệp DOS dễ dàng hơn.

Các câu lệnh nói trên cũng giúp chúng ta chép tệp sang đĩa mềm dễ dàng hơn. vì lúc ấy chúng ta có thể sử dụng cách gọi tên của DOS, chẳng hạn như ổ đĩa A, thay vì phải gọi theo Linux là /dev/fd0. Muốn biết thêm chi tiết về họ lệnh m-, chúng ta gõ:

```
man mtools
```

Tuy nhiên trong lúc cài đặt, chúng ta cần chọn phần mềm “DOS emulation” thì mới sử dụng được **mtools**.

Lệnh	Mô tả
mattrib	Hiển thị thuộc tính của tệp được lựa chọn
mcd	Chuyển sang thư mục được nêu ở đường dẫn
mcop	Chép các tệp sang đường dẫn được nêu
mdel	Xoá bỏ các tệp được chọn
mdir	Liệt kê thư mục

mformat	Format đĩa mềm
mlabel	Đặt tên nhãn cho file system DOS
mmd	Tạo thư mục mới
mrd	Xoá bỏ một thư mục (thư mục này phải rỗng, như đối với DOS)
mren	Đặt lại tên cho tệp DOS
mtype	Hiển thị nội dung văn bản của tệp DOS

Bảng 4.4: Các lệnh họ m-

5. Đóng tắt Linux và chạy các chương trình Linux

Mục tiêu :

- Đóng tắt được Linux
- Chạy được các chương trình Linux

5.1. Đóng tắt Linux

Khi chúng ta dùng máy PC chạy với DOS, làm việc xong chúng ta chỉ cần tắt công tắc điện. Nhưng với Windows làm như thế sẽ tổn hại đến các tệp.

Với Linux, mức độ rủi ro này còn cao hơn nữa, đặc biệt sẽ tổn hại cả phần cứng lẫn hệ thống tệp. Chúng ta phải đóng tắt Linux theo quy định, nếu không muốn lần làm việc tiếp theo sẽ gặp khó khăn khi khởi động máy.

Trước khi ghi dữ liệu vào đĩa cứng, Linux lưu rất nhiều thông tin về chính mình và thông tin của những tệp đang nằm trong bộ nhớ, tại những vùng đệm, còn gọi là bộ nhớ trung gian (buffer). Tiến trình này giúp cải thiện năng suất của cả hệ thống, đồng thời kiểm soát việc sử dụng phần cứng. Đây là việc mà các hệ điều hành đa nhiệm phải làm để một người sử dụng này không thể sử dụng một thiết bị nào đó đang phục vụ một người sử dụng khác. Nếu đột ngột tắt máy, các thông tin vừa kể sẽ mất đi và hệ thống tệp bị hỏng.

Là một hệ điều hành đa nhiệm và dành cho nhiều người sử dụng, Linux phải đảm bảo rằng từng thành viên phải ngưng phiên làm việc đúng quy cách và lưu lại công việc đang tiến hành trước khi đóng tắt cả hệ, như thế sẽ tránh được trường hợp mất dữ liệu và hỏng hệ thống tệp. Điều này cũng giúp cho những người khác đang cùng sử dụng hệ Linux này có đủ thời gian đăng xuất. Muốn đóng tắt Linux cho đúng, chúng ta phải dùng lệnh **shutdown** với cú pháp như sau: `shutdown [-r] thời_gian_đóng_tắt [lời_nhắc]`

Tùy chọn `-r` có nghĩa là Linux phải lập tức khởi động lại sau khi đóng tắt. Điều này có ích khi chúng ta muốn thoát khỏi Linux để khởi động một hệ điều hành khác.

Thời_gian_đóng_tắt báo là tham số báo cho hệ điều hành biết khi nào có thể đóng tắt. Thời gian được tính theo hình thức 24 giờ, thí dụ nếu muốn hệ điều hành đóng tắt vào lúc 11 giờ đêm, chúng ta cần gõ lệnh:

```
shutdown 23:00
```

Tham số [lời nhắc] là thông báo chung cho tất cả những người sử dụng đang ở trong mạng. Từng người sẽ thấy lời nhắc này trên màn hình của mình.

Thí dụ khi muốn ngưng sử dụng máy để thực hiện công tác sao lưu hàng tuần, chúng ta gõ lệnh như sau để mọi người lo mà đăng xuất:

```
[root@web] /root] # shutdown -r 23:00
```

 Đóng tắt hệ thống vào lúc 11:00 pm để bảo trì hệ thống.

Lưu ý: Trên một số hệ thống đôi khi Linux có thể hiểu được nhóm phím khởi động lại <Ctrl-Alt-Del> và sẽ thực hiện thao tác đóng tắt đúng quy trình như khi chúng ta gõ lệnh **shutdown**. Tuy nhiên ở một số hệ thống khác, Linux không hiểu được nhóm phím ấy.

Nếu lỡ tắt hệ thống không đúng cách và làm hỏng kết cấu tệp, chúng ta có thể dùng lệnh **fsck** để thử sửa lại hệ thống tệp.

5.2. Chạy các chương trình Linux

Khi đã quen các thao tác mở, tắt trong Linux và vài lệnh cơ bản, chúng ta có thể bắt đầu thử một số ứng dụng đã cài đặt khi thiết lập hệ thống. Những ứng dụng này bao gồm các tiện ích, từ một cái máy tính bỏ túi đơn giản cho đến những bộ biên dịch C và C++. Một vài ứng dụng như thế có giá trị lớn; may thay nhờ vào giấy phép GNU, nhiều ứng dụng đã trở thành miễn phí.

Nhiều chương trình khác dành cho Linux cũng hiện diện miễn phí trên Internet và chúng ta có thể lấy được chúng nhờ vào một chương trình tải nạp đi kèm trong bản phát hành Slackware và RedHat. Ngoài ra nhiều cửa hàng có thể cung cấp cho chúng ta những đĩa CD-ROM với hàng trăm chương trình UNIX dưới dạng mã nguồn. Chúng ta có thể chọn lọc vài chương trình yêu thích từ CD-ROM rồi dùng các chương trình **gcc** và **g++** để biên dịch các chương trình ấy.

Những chương trình này chủ yếu làm việc ở chế độ văn bản, do đó không cần phải chạy hệ X Window.

5.2.1. Sử dụng chương trình CD Player

CD Player là một chương trình chơi nhạc từ đĩa CD, được cài đặt sẵn trong bản phát hành RedHat Linux. Chúng ta nên thử nó một lần xem, nếu máy chúng ta có một ổ CD-ROM chấp nhận đĩa CD audio. Thực tế có thể CD Player không tương thích với tất cả các loại ổ CD-ROM có trên thị trường.

Chương trình này cho phép dùng bàn phím điều khiển CD, vì vậy chúng ta nhớ bật bàn phím sang chế độ <Num Lock>.

5.2.2. Sử dụng Gnumeric và KSpread

Trước khi phần mềm VisiCalc ra đời, các nhà lập kế hoạch phải dùng những tờ giấy kẻ ô gọi là spreadsheet. Là phiên bản điện tử của tờ spreadsheet, VisiCalc đã làm một cuộc cách mạng ở khâu tính toán và lập kế hoạch.

Ngày nay các chương trình như Microsoft Excel hoặc Lotus 1-2-3 đang tiếp nối truyền thống của VisiCalc. Trong Linux, Gnumeric và KSpread cũng làm chức năng đó.

Gnumeric và KSpread đều có dạng bảng tính (spreadsheet calculator) gồm nhiều dòng và cột. Mỗi ô chứa một giá trị số học, một chuỗi ký tự, hoặc một

biểu thức. Các chuỗi ký tự có thể dựa vào những ô khác để lập thành nhiều mối liên hệ phức hợp.

Nếu đã từng làm việc với các chương trình bảng tính khác, chúng ta sẽ không gặp khó khăn với các lệnh của Gnumeric và KSpread.

5.2.3. Sử dụng bc Calculator

bc là một chương trình tính toán theo câu lệnh vì bản thân nó có một ngôn ngữ lập trình cho phép chúng ta thực hiện tương tác.

Sau khi chúng ta gõ lệnh, bc hiển thị vài dòng lưu ý về bản quyền tác giả và dấu nhắc sẽ nhấp nháy chờ lệnh. Chúng ta có thể ra lệnh làm hai phép tính cộng và trừ. Chúng ta cũng có thể ra lệnh nhân và chia, song phiên bản bc phát hành kèm với RH Linux lại xén bớt kết quả hai phép tính này.

bc rất tiện lợi cho các phép tính đơn giản. Một điều tiện lợi nữa là bc có khả năng lưu giá trị từ một phép tính này cho phép tính sau chỉ bằng một cú pháp đơn giản, đó là tên-của-biến = biểu-thức.

Thí dụ sau đây tính giá trị của $125*5$, sau đó lưu kết quả vào biến var1. Kết quả phép tính được hiển thị bằng cách gõ tên của biến var1 và sẽ in giá trị (625) ở hàng kế tiếp. Thí dụ này còn lập biến var2 làm nơi chứa kết quả (125) của var1 chia cho 5.

```
var1 = 125*5
```

```
var1
```

```
625
```

```
var2 = var1/5
```

```
var2
```

```
125
```

5.2.4. Sử dụng chương trình minicom

Chúng ta có thể nối kết với thế giới nếu chúng ta có một modem và một phần mềm viễn thông. Linux cung cấp một phần mềm mang tên **minicom** và chúng ta chỉ cần nối modem với cổng nối tiếp COM trên máy PC.

Cũng giống như nhiều phần mềm Linux khác, **minicom** do một người viết ra nhưng được nhiều người khác trên Internet giúp đỡ hoàn chỉnh và phần mềm này chạy rất tốt, có khả năng cạnh tranh với nhiều ứng dụng thương mại khác. Để biết thêm các chức năng chi tiết của **minicom**, chúng ta xem trợ giúp bằng lệnh **man**.

Điều đầu tiên nên ghi nhớ là **minicom** dùng nhóm phím <Ctrl-Shift-a> cho một số chức năng, chẳng hạn như auto-dial (tự động quay số điện thoại) và file downloading (tải tệp xuống). Đang ở trong **minicom**, nếu cần trợ giúp chúng ta bấm <Ctrl-a><z> để hiển thị màn hình tóm tắt các câu lệnh.

Phím	Mô tả
D	Truy cập thư mục quay số gọi
S	Gửi tệp đi
P	Liệt kê các tham số liên lạc

L	Bật/tắt việc ghi biên bản phiên làm việc vào một tệp
F	Gửi tín hiệu BREAK cho thiết bị bên kia đầu dây
T	Phỏng tạo terminal giữa các chế độ vt100, Minix, hoặc ANSI
W	Bật/tắt chế độ làm tròn dòng
G	Chạy một tệp script của minicom
R	Nhận một tệp
A	Thêm vào cuối dòng một ký tự chuyển dòng
H	Dừng liên lạc đường điện thoại
M	Khởi động modem
K	Chạy giao thức kermit
E	Bật/tắt chế độ bán song công
C	Xoá màn hình tại chỗ
O	Giúp chúng ta lập cấu hình minicom
J	Nhảy tắt đến một command shell khác
X	Thoát khỏi modem và khởi động lại modem
I	Chế độ cursor key
Z	Hiển thị màn hình trợ giúp
B	Cuốn ngược trở về cửa sổ terminal

Khi đang ở trong cửa sổ trợ giúp, chúng ta chỉ việc gõ ký tự tương ứng để thi hành lệnh. Tuy nhiên nếu đang ở chương trình **minicom**, chúng ta phải bấm <Ctrl-a> trước ký tự đã chọn. **minicom** cho phép dùng bốn giao thức chuyển tệp: zmodem, ymodem, xmodem và kermit. Chúng ta nên thử chạy zmodem trước hết vì khả năng vượt trội của giao thức này về việc phục hồi khi có lỗi. Nếu các thiết bị cuối mà chúng ta định liên lạc lại không có zmodem, chúng ta nên thử dùng giao thức theo thứ tự vừa được kê ra. Điều này không có nghĩa rằng kermit là dở. Kermit không dở, nhưng chỉ chậm hơn các giao thức khác. Điểm mạnh của kermit là hầu hết các hệ thống khác đều chấp nhận giao thức này.

Điều thứ hai phải ghi nhớ khi sử dụng **minicom** là phần mềm viễn thông này có vài câu lệnh đặc biệt nguy hiểm. Đó là những câu lệnh giúp người sử dụng có quyền hạn như một superuser. Do vậy ai đó khi chạy **minicom** sẽ làm được một vài việc mà biết đâu chúng ta không muốn người ấy làm.

6. Chạy các chương trình DOS trong Linux

Mục tiêu :

- Cài đặt được phần mềm DOSEMU
- Lập được cấu hình DOSEMU

Sau khi chạy các ứng dụng Linux một thời gian, có thể chúng ta sẽ muốn chạy vài chương trình DOS hoặc Windows. Một số phần mềm như DOSEMU và Wine cho phép làm việc này bằng cách mô phỏng nhiều hệ điều hành khác nhau.

DOSEMU giúp các chương trình dựa vào DOS (và những biến thể khác như PC-DOS) chạy được với Linux. DOSEMU có nghĩa là Mô phỏng DOS (DOS

EMUlator). Còn phần mềm Wine thì mô phỏng Windows và sẽ được giới thiệu ở mục “Chạy các chương trình Windows với Linux”.

Ghi chú: Một vài bản phát hành Linux có lệnh “simply dos” để khởi động một chương trình chỉnh sửa theo chế độ DOS. Những bản phát hành mang tính thương mại như RedHat Linux hoặc tương tự đều có lệnh này.

6.1. Cài đặt DOSEMU

Phần mềm DOSEMU (phiên bản thường dùng là 1.0.2) được lưu ở dạng nén trong thư mục /contrib/ củá đĩa CD-ROM với tên dosemu_1.0.2.tgz hoặc dosemu_1.0.2.tar. Chúng ta sao tệp này vào thư mục /usr/src rồi bung ra bằng những lệnh giải nén thích hợp như sau đây:

```
[root@web src] # gzip -d dosemu_1.0.2.tgz
```

```
[root@web src] # tar -xvf dosemu_1.0.2.tar
```

Sau đó chúng ta tạo ra một số tệp bằng những lệnh khác:

```
[root@web src] # make config
```

```
[root@web src] # make depend
```

```
[root@web src] # make most
```

Từ đó các tệp DOSEMU sẽ hiện diện trong thư mục /var/lib/dosemu. Để thực hiện thao tác vừa qua, chúng ta phải đăng nhập như là root và máy chúng ta phải còn trống ít nhất 10 MB bộ nhớ ảo.

Ghi chú: Máy chúng ta phải được cài đặt gói phần mềm Development, một số công cụ và trình biên dịch để xây dựng ứng dụng mô phỏng DOS.

Chúng ta cũng có thể dùng dạng RPM của phiên bản này, gọi là dosemu_1.0.1-1.i386.rpm. Khi đó chúng ta chỉ cần gõ lệnh sau:

```
[root@web src] # rpm -i dosemu_1.0.1-1.i386.rpm
```

6.2. Lập cấu hình DOSEMU

DOSEMU được xây dựng để chạy một số lệnh DOS và chương trình DOS trong môi trường DOS phỏng tạo.

Sau khi xây dựng xong phần mô phỏng, chúng ta phải lập cấu hình cho hệ thống. Đầu tiên chúng ta tạo ra một đĩa mềm DOS khởi động, sau đó chép các tệp DOS sau đây vào đĩa: command.com, **fdisk.exe** và sys.com.

Tiếp theo, chúng ta chép các tệp DOSEMU từ thư mục dosemu vào đĩa: emufs.sys, ems.sys, cdrom.sys và exitemu.com. Chúng ta có thể dùng các lệnh -m được đề cập trước đây tại mục “Xử lý các tệp DOS với Linux” để sao chép các tệp từ phân vùng Linux sang đĩa mềm.

Ghi chú: Nếu khó tìm các tệp Linux, chúng ta hãy sử dụng lệnh find, thí dụ như lệnh sau đây sẽ hiển thị vị trí tệp ấy trên hệ thống máy của chúng ta:

```
find / -name emufs.sys-print
```

DOSEMU cần phải có tệp cấu hình dosemu.conf và global.conf để chạy được hoàn hảo. Tệp global.conf là tệp cấu hình chính của dosemu luôn được thực thi khi khởi động dosemu. Tệp dosemu.conf là tệp chứa các giá trị cho những cấu

hình đã khai báo trong `global.conf`. Có thể sửa chữa chúng thông qua công cụ ‘`setup-dosemu`’.

Chúng ta phải tùy chỉnh tệp `dosemu.conf` cho hợp với hệ thống máy mình nhưng không nên chỉnh sửa gì ở tệp `global.conf`. Để có một ý niệm ban đầu, chúng ta xem các tệp mẫu như danh sách 5.1 dưới đây hiển thị toàn bộ `dosemu.conf` và `global.conf` trong bộ cài đặt. Những nhận xét được đánh dấu bằng dấu thăng (pound #) và hầu hết các tùy chọn đều mang hình thức giá trị tham số. Nếu một tham số nào đó có hơn một giá trị, thì các giá trị ấy được đặt trong ngoặc móc ({}).

Các giá trị sẽ được xem là một chuỗi (string) hay một giá trị số (numeric) hoặc logic (boolean) tùy theo việc chúng được đặt giữa 2 dấu nháy kép hay giữa 2 ngoặc đơn.

Nội dung tệp `dosemu.conf` và `global.conf` điển hình (xem phần phụ lục)

Sau đó chúng ta phải dùng một chương trình soạn thảo văn bản để thay đổi mọi thiết lập sẵn có trong tệp thí dụ sao cho phù hợp với hệ thống máy chúng ta. Một số thiết bị như bộ xử lý card video cũng phải tương thích.

Ghi chú: Chúng ta cũng có thể khởi động DOSEMU từ một phân vùng (partition) trên ổ đĩa cứng thay vì từ đĩa mềm. Muốn truy cập ổ đĩa cứng, chúng ta chỉ cần lập cấu hình cho một ổ đĩa cứng hoặc một phân vùng trong tệp `dosemu.conf`.

6.3. Chạy DOSEMU

Muốn chạy DOSEMU chúng ta chỉ cần gõ lệnh tại dấu nhắc Linux. Muốn thoát ra, chúng ta gõ `exitemu`. Bảng 5.9 liệt kê các tùy chọn từ dòng lệnh cho DOSEMU. Chúng ta dùng tùy chọn `-?` để hiển thị toàn bộ các tham số.

Bảng 5.9 Các tham số DOSEMU tại dòng lệnh

Tham số	Mô tả
-A	Khởi động từ ổ đĩa mềm A
-C	Khởi động từ ổ đĩa cứng
-c	Tối ưu hoá hiệu năng video từ các terminal ảo
-D	Lập các tùy chọn gỡ lỗi
-e	Xác định bộ nhớ EMS
-F#	Xác định số lượng (#) đĩa mềm để sử dụng từ <code>dosemu.conf</code>
-f	Hoán chuyển định nghĩa của các ổ đĩa mềm A và B
-H#	Xác định số lượng (#) đĩa cứng để sử dụng từ <code>dosemu.conf</code>
-k	Sử dụng bàn phím được định nghĩa bằng các tham số <code>rawkeyboard</code> trong tệp <code>dosemu.conf</code>
-p	Chép các thông số gỡ lỗi vào một tệp
-t	Phát ngắt thời gian 9
-V	Kích hoạt mô phỏng màn hình VGA
-x	Xác định bộ nhớ XMS
-?	Hiển thị trợ giúp tóm tắt cho từng lệnh

-2	Mô phỏng máy 286
-3	Mô phỏng máy 386
-4	Mô phỏng máy 486

Từ dấu nhắc DOS của DOSEMU, chúng ta có thể chạy hầu hết các chương trình DOS ngoại trừ chương trình nào đòi hỏi DPMI (DOS Protected Mode Interface: Giao diện theo chế độ DOS có bảo vệ). Chúng ta chỉ cần gõ tên của chương trình, sau đó DOSEMU sẽ căn cứ theo đường dẫn mà chúng ta đã cung cấp để nạp và chạy, với điều kiện là DOSEMU tìm được chương trình ấy.

Chạy chương trình với DOSEMU có nhiều rắc rối, đa phần bởi vì máy đang mô phỏng DOS thay vì chạy DOS thật. Việc mô phỏng sẽ làm giảm tốc độ hệ thống và tốc độ sẽ đặc biệt chậm nếu máy phải chạy các chương trình Linux khác ở các terminal.

Nhiều chương trình của DOS sẽ chiếm bộ xử lý CPU, khiến các chương trình Linux khác khó mà chen vào. Để giải toả bớt vấn đề, người ta đã viết một chương trình mang tên garrot giúp cho Linux có điều kiện chiếm CPU. Chúng ta có thể tải garrot xuống từ website FTP mang tên sunsisite.unc.edu tại thư mục /pub/linux/alpha/dosemu.

7. Chạy các chương trình Windows với Linux

Mục tiêu :

- Trình bày được cách chạy các chương trình DOS trong Linux và chạy các chương trình Windows với Linux

Vì DOSEMU không thể thực hiện các chương trình của Windows nên người ta đã làm ra phần mềm mô phỏng gọi là Wine. Wine viết tắt từ WINDows Emulator, hoặc từ Wine Is Not a Windows Emulator. Cả hai lối viết tắt này đều được giải thích trong mục FAQ của Windows (FAQ: những câu thường xuyên được hỏi).

Nếu muốn dùng thử Wine, chúng ta nên đọc qua Windows FAQ. Wine chưa được triển khai rộng như DOSEMU, chính vì thế mà nó còn bỏ sót nhiều lỗi và nhiều chương trình Windows cũng chưa được nó chấp nhận. Thật ra muốn dùng Wine chúng ta phải cài đặt Windows trên một phân vùng nào đó mà Linux truy cập được, bởi vì Wine còn phụ thuộc nhiều vào Windows. Wine cũng đòi hỏi hệ thống đồ họa X Window phải được cài đặt trước.

Muốn chạy thử Wine, chúng ta cần những thứ sau đây:

- Bản kernel của Linux, phiên bản 0.99.13 hoặc mới hơn (thí dụ 2.4.18-3).
- Mã nguồn của Wine, bởi vì Wine chỉ có ở dạng này.
- Ít nhất 16 MB bộ nhớ RAM, nếu chạy với 64 MB hoặc nhiều hơn thì càng tốt.
- X Window đã cài đặt và thiết lập cấu hình xong.
- Một thiết bị điều khiển con chạy (cursor) trên màn hình, chẳng hạn như chuột.

- Microsoft Windows cài đặt ở phân vùng nào mà Linux có thể truy cập được.

Vì Wine đang được tiếp tục phát triển cho nên có thể còn có phiên bản mới. Chúng ta nên coi thư mục /pub/Linux/ALPHA/wine/development, ở địa chỉ sunsite.unc.edu để có những thông tin cập nhật. Tên sẽ được đặt tên tùy theo ngày phát hành, chẳng hạn như Wine-20020509.tar.gz. Muốn tìm hiểu thêm về Wine, chúng ta có thể tải các tệp FAQ và HOWTO mới nhất xuống rồi đọc chúng. Những tệp này sẽ giúp chúng ta biên soạn, cài đặt, thiết lập cấu hình và sử dụng Wine.

Cài đặt Wine giống với cài đặt DOSEMU, chỉ khác là chúng ta có thể đặt tệp nén ở bất cứ thư mục nào. Chúng ta dùng lệnh **tar** để bung tệp ra như ở thí dụ sau:

```
[root@web wine] # gzip -d wine - 20020509.tar.gz
```

```
[root@web wine] # tar -xvf wine - 20020509.tar
```

Việc biên dịch mã nguồn của Wine yêu cầu tỉ mỉ hơn so với DOSEMU, tựa như biên dịch một kernel mới vậy. Chúng ta phải trả lời nhiều câu hỏi trong tiến trình xây dựng. HOWTO về Wine giải thích cặn kẽ việc này. Tiếp theo chúng ta sẽ cung cấp tham số về thời gian chạy máy. Những tham số này được lưu tại tệp /usr/local/etc/wine.conf (chú ý: tùy theo phiên bản của wine hay Linux mà vị trí này có thể thay đổi). Có thể chỉnh sửa tệp này bằng cách thủ công, song tốt hơn chúng ta nên dùng chương trình cấu hình kèm theo để thực hiện. Sau khi lập xong cấu hình cho các tệp biên dịch và tệp tham số thời gian chạy máy, chúng ta chỉ cần ra lệnh **make** để xây dựng Wine. Để bắt đầu sử dụng Wine, chúng ta gọi phần mô phỏng và cung cấp đường dẫn đến một tệp thi hành của Windows.

```
[lan_anh@web~] $wine /dos/windows/winmine.exe
```

hay:

```
[lan_anh@web~] $wine C:\\dos\\windows\\winmine.exe
```

B. CÂU HỎI VÀ BÀI TẬP

Câu 1: Trình bày cách thiết lập tài khoản của Linux?

Câu 2: Nêu cách sử dụng các lệnh cơ bản trong Linux?

Câu 3: Nêu cách xử lý các tệp DOS trong Linux?

Câu 4: Trình bày cách chạy các chương trình DOS trong Linux và chạy các chương trình Windows với Linux

Câu 5: Thực hành đăng nhập hệ thống Linux?

Hướng dẫn thực hành

1. Truy cập vào máy tính đã cài đặt hệ điều hành Linux

Khởi động máy đã cài đặt Linux, xuất hiện dấu nhắc khởi động hệ điều hành

Boot : linux

Khi HĐH Linux khởi động, xuất hiện dấu nhắc truy cập hệ thống :

login : password :

Người dùng nhập vào username và password tương ứng, trên màn hình xuất hiện dấu nhắc của hệ thống như sau :

```
[user12@linux user12]
```

2. Sử dụng Telnet để truy cập vào máy Linux từ xa

Truy cập vào Server LINUX từ máy Windows. Yêu cầu máy Windows đã cài đặt mạng. Để kiểm tra hệ thống mạng, từ dấu nhắc cửa lệnh trên Windows, gõ lệnh :

```
C:\>ping 200.201.202.180
```

Nếu trên màn hình xuất hiện : Reply from 200.201.202.180 ... thì nghĩa là máy tính có khả năng truy cập vào Server LINUX, ngược lại, nếu có thông báo nào khác thông báo như trên thì nên kiểm tra lại cấu hình mạng trên máy. Tiếp theo, ta gõ lệnh :

```
telnet 200.201.202.180
```

Sau một khoảng thời gian thiết lập liên kết, trên cửa sổ telnet xuất hiện :

```
login : password :
```

Người dùng nhập vào username và password tương ứng.

Ví dụ : Đăng nhập vào với tài khoản user12, trên màn hình xuất hiện như sau

```
login: user12
```

```
Password:
```

```
Last login: Wed Apr 7 08:35:50 from 131.16.16.21 [user12@linux user12]$
```

3. Thoát khỏi hệ thống

Thoát khỏi phiên làm việc : #exit hoặc #logout

Chấm dứt hoạt động của hệ thống : #shutdown -h now

BÀI 5: NÂNG CẤP VÀ CÀI ĐẶT PHẦN MỀM VỚI RPM

Mã bài: MD37-05

❖ Giới thiệu

Các phần mềm dùng cho Linux có thể được nâng cấp hoặc tạo mới. Việc cài đặt phần mềm có thể là thủ công (tự biên dịch từ các tệp nguồn) hay bằng RPM.

Trong bài này chúng ta sẽ cùng nghiên cứu về RPM với các nội dung chính như sau:

- Các thuật ngữ liên quan

- Chính sách nâng cấp phần mềm
- Cài đặt phần mềm
- Sử dụng RPM
- Nâng cấp kernel

❖ Mục tiêu

- Hiểu được những kiến thức cơ bản về RPM để cài đặt các phần mềm cần thiết dùng cho Linux.
- Hiểu vững các chính sách nâng cấp phần mềm, cài đặt được một số phần mềm cơ bản cũng như cài đặt được những phiên bản sửa lỗi Kernel Linux.
- Nâng cao nhận thức chia sẻ công đồng.
- Tự tin cài đặt và sử dụng các ứng dụng trong Linux.

❖ Nội dung chính

A. LÝ THUYẾT

1. Chính sách nâng cấp phần mềm

Mục tiêu :

- Trình bày được các chính sách nâng cấp phần mềm

Nên nâng cấp những phần mềm nào và bao lâu thì thực hiện một lần? Câu trả lời tùy thuộc vào mục đích sử dụng của hệ thống – cá nhân hay cơ quan – và tùy theo yêu cầu của các user. Phần mềm thường thay đổi phiên bản nhanh chóng, chưa kể việc nhiều thành phần khác nhau của Linux được cập nhật từ mọi nơi. Do đó chúng ta sẽ không kịp sử dụng hệ thống một cách thành thạo nếu cứ gắng sức chạy theo những bản nâng cấp liên tục ra đời.

Mỗi khi nâng cấp phần mềm cho hệ thống, chúng ta không nhất thiết phải cài đặt lại toàn bộ Linux. Thông thường chỉ một phần nhỏ của phần mềm hệ thống phải thay đổi khi cài đặt phiên bản nâng cấp. Có thể chúng ta sẽ phải nâng cấp phần kernel hoặc thư viện hệ thống. Tuy nhiên khi nâng cấp gói phần mềm ứng dụng thì chúng ta phải cài đặt một phiên bản hoàn toàn mới.

Nhìn chung, chúng ta nên nâng cấp hệ thống nếu phiên bản mới của hệ thống hoặc của ứng dụng có thể giải quyết được những vấn đề quan trọng, hoặc tăng thêm chức năng nào đó mà chúng ta cần có. Vấn đề như thế nào được gọi là quan trọng thì tùy chúng ta quyết định.

Ghi chú: Nên sao lưu hệ thống hiện hành trước khi nâng cấp phần mềm, để phòng trường hợp hỏng hóc chúng ta vẫn có thể trở lại sử dụng hệ thống cũ.

2. Cài đặt phần mềm

Mục tiêu :

- Nêu được các công việc và trách nhiệm của quản trị viên hệ thống

2.1. Giới thiệu

Cài đặt các chương trình trọng yếu vào hệ thống Linux thường phức tạp hơn là cài vào một hệ điều hành đơn nhiệm như DOS. Bản chất multiuser của Linux cho phép mỗi ứng dụng trên hệ thống cùng lúc thỏa mãn được yêu cầu truy cập từ nhiều phía khác nhau. Rắc rối hơn nữa, hầu hết các chương trình đều đòi

hỏi phải được lập cấu hình cho hợp với hệ thống trước khi sử dụng. Do đó quản trị viên phải xác định từng mục cho hợp với cấu hình hệ thống trong tiến trình lập cấu hình.

Thí dụ user này dùng terminal loại cũ chạy ở chế độ văn bản, trong khi user khác đang sử dụng một thiết bị đời mới nhất với X Window. Lúc ấy superuser phải đảm bảo rằng ứng dụng đáp ứng được thiết bị cũ chỉ gửi đi các ký tự ASCII – nghĩa là chữ và số - trong khi thiết bị X Window phải nhận được đồ họa và màu sắc.

Cài đặt chương trình vào Linux phức tạp hơn, bởi vì quản trị viên phải tạo ra thư mục mới để chứa các tệp liên kết với chương trình mới. Một vài gói phần mềm mới yêu cầu lập lại cấu hình cho các thiết bị hệ thống.

Một user bình thường chỉ phải bỏ công tìm hiểu về các chức năng của ứng dụng mới và nhớ thêm vài câu lệnh mới, trong lúc đó quản trị viên có trách nhiệm đảm bảo rằng tài nguyên hệ thống phải được phân bổ, lập cấu hình và duy trì đúng đắn. Và đương nhiên ứng dụng mới không được xung đột với các chương trình sẵn có trên hệ thống.

Nhìn từ bên ngoài, việc cài đặt phần mềm bằng cách sử dụng menu và câu lệnh có vẻ đơn giản, song đối với hệ thống thì đó là công việc phức tạp. Những ứng dụng dành cho hệ điều hành một người sử dụng (chẳng hạn như DOS) thường chỉ chạy một mình mà không gặp cạnh tranh. Trên hệ thống Linux, ngay cả khi chỉ có một người đăng nhập, vẫn có nhiều tiến trình đang làm việc cùng lúc. Mức độ phức tạp sẽ tỷ lệ thuận với số người sử dụng, chưa kể đến việc nhiều người dùng một ứng dụng cùng lúc.

Sở trường của Linux là điều phối được nhiều tiến trình, nhiều chương trình, thiết bị và user cùng lúc. Muốn tồn tại trong môi trường chặt chẽ này, ứng dụng phải được nạp vào đúng cách, nếu không toàn bộ hệ thống sẽ treo, mọi chương trình sẽ ngưng và hàng loạt user sẽ bị thiệt thòi. Do đó khi nạp ứng dụng mới vào hệ thống, quản trị viên hoặc superuser phải thử nghiệm ứng dụng sau khi cài đặt và làm sao cho ứng dụng ấy thích hợp với cả hệ thống. Muốn hiểu tiến trình nạp phần mềm vào hệ thống Linux, chúng ta phải biết rõ trách nhiệm và quyền hạn của quản trị viên hệ thống là gì.

2.2. Công việc của quản trị viên hệ thống

Nếu sử dụng Linux trên hệ thống nhỏ, có khả năng chúng ta là quản trị viên hệ thống của chính chúng ta. Chúng ta chỉ cài đặt và chạy các chương trình của chúng ta. Trách nhiệm của chúng ta là sao lưu tệp, duy trì một khoảng trống thích hợp trên ổ cứng, quản lý bộ nhớ, cùng với một số việc khác đảm bảo cho hệ thống chạy hữu hiệu và có năng suất.

Quản trị viên của một hệ thống lớn phải có thêm những trách nhiệm như sau:

- Khởi động và đóng tắt hệ thống.
- Bảo đảm còn đủ khoảng trống trên ổ cứng và hệ thống tệp không bị lỗi.

- Bảo đảm số lượng tối đa các user truy cập được phần cứng và phần mềm hệ thống.
- Bảo vệ hệ thống chống lại các hành động xâm nhập bất hợp pháp và phá hoại.
- Thiết lập liên lạc với các hệ thống tin học khác.
- Tạo ra và xoá bỏ các trương khoản của hệ thống.
- Làm việc với các hãng cung cấp phần cứng và phần mềm, với các chuyên gia và những người có trách nhiệm hỗ trợ hệ thống.
- Cài đặt, lắp đặt và gỡ lỗi cho terminal, máy in, ổ đĩa, cùng với các cấu kiện khác.
- Cài đặt và duy trì phần mềm, kể cả các ứng dụng mới và bản cập nhật hệ điều hành.

Rất nhiều khi các user thích đăng nhập với tư cách là root và gõ đủ thứ lệnh, thậm chí họ có thể gây ra mọi loại vấn đề. Vì vậy quản trị viên cần dành quyền sử dụng trương khoản root chỉ cho công việc quản trị. Còn để làm các công việc hàng ngày thì mỗi người chỉ nên dùng trương khoản cá nhân của mình mà thôi.

3. Sử dụng RPM

Mục tiêu :

- *Nêu được vị trí của các gói phần mềm RPM*
- *Cài đặt được gói phần mềm bằng RPM*
- *Gỡ bỏ được cài đặt gói phần mềm bằng RPM*

Hai bản phát hành Red Hat Linux và Caldera OpenLinux đều sử dụng gói phần mềm (Package) để quản lý việc cài đặt. Gói phần mềm là một chương trình đầy đủ đã được thử nghiệm và lập sẵn cấu hình cài đặt. Gói phần mềm được xây dựng từ các tệp mã nguồn mở, làm cho người sử dụng lẫn người triển khai đều biết được mình đang có cái gì trong tay. Để quản lý những phần mềm ấy, Red Hat

Software đã phát triển công cụ RedHat Package Manager (RPM).

RPM làm việc theo sáu chế độ khác nhau, trong đó chúng ta có thể sử dụng năm chế độ từ dòng lệnh hoặc bằng gnorpm, một công cụ căn cứ trên X Window. Chúng ta cũng có thể sử dụng GnomeRPM (gnorpm) hay KDERPM. Các chế độ đó là cài đặt, tháo bỏ cài đặt, cập nhật hoá, tìm, kiểm sát và xây dựng. Chúng ta chỉ có thể xây dựng một gói phần mềm RPM từ dòng lệnh với cú pháp sau đây:

`rpm [tuỳ chọn] tên_gói phần mềm`

Với tuỳ chọn là một trong những flag (cờ hiệu) mà RPM dùng để thao tác các gói phần mềm và tên_gói phần mềm xác định tên của gói phần mềm được sử dụng.

Tên gói phần mềm thường có dạng quota-1.55-4.i386.rpm. Tên gói phần mềm bao gồm những thành phần như sau:

Tên	quota
phiên bản	1.55
ấn bản	4
cấu trúc máy tính	i386
phần mở rộng	.rpm

Tuy nhiên tệp của gói phần mềm mang tên gì cũng được bởi vì bản thân thông tin về gói phần mềm đã có sẵn bên trong tệp.

3.1. Vị trí của các gói phần mềm

Đa phần các gói phần mềm được cung cấp kèm theo bản phát hành này nằm ở thư mục /RedHat/RPMS trên CD-ROM. Muốn cài CD-ROM và liệt kê các gói phần mềm khác nhau, chúng ta dùng những lệnh sau:

```
cd /mnt
```

```
mount /mnt/cdrom
```

```
cd cdrom/RedHat /RPMS
```

```
ls | more
```

Hầu hết các gói phần mềm đã được mặc định cài đặt trong quá trình cài đặt Linux. Nếu lúc ấy chúng ta chọn không cài đặt và hiện giờ lại muốn cài đặt thì cứ tiến hành tùy ý.

RPM cũng giúp cài đặt các gói phần mềm nằm trên những máy tính khác bằng FTP mà chúng ta sẽ xem ở đoạn tiếp theo.

3.2. Cài đặt gói phần mềm bằng RPM

Từ dòng lệnh, chúng ta dùng tùy chọn `-i` như sau:

```
rpm -i quota -1.55-4.i386.rpm
```

Lệnh này sẽ cài đặt gói phần mềm quota vào hệ thống máy chúng ta. Tùy chọn `-i` cho phép cài đặt gói phần mềm quota-1.55-4.i386.rpm vào hệ thống cục bộ. RPM thực hiện quá trình cài đặt qua các bước sau:

- Kiểm tra tính phụ thuộc. Mỗi gói phần mềm có khả năng phụ thuộc vào phần mềm khác đã cài sẵn trên máy.

- Kiểm tra tiềm năng xung đột. RPM xét xem một thành phần đã cài đặt sẵn hay chưa, hoặc thành phần ấy có cũ hơn thành phần đang trong quá trình cài đặt hay không.

- Xử lý các tệp cấu hình. RPM sẽ thử cung cấp một tệp cấu hình thích hợp và nếu phát hiện một tệp cấu hình có sẵn, RPM sẽ lưu tệp này lại để đối chiếu trong tương lai.

- Cài đặt tệp. RPM mở các gói phần mềm thành phần và đặt chúng vào thư mục thích hợp.

- Xử lý sau khi cài đặt. Sau khi cài đặt các thành phần xong rồi, RPM tiến hành các công việc cần thiết để lập cấu hình hệ thống cho đúng đắn.

- Cập nhật cơ sở dữ liệu. RPM ghi chép lại lộ trình của mình vào một cơ sở dữ liệu.

Trong khi cài đặt, câu lệnh nêu trên sẽ không thông báo phản hồi, song chúng ta có thể dùng tùy chọn `-v` (verbose) để lấy thông báo. Bảng 6.2 liệt kê các tùy chọn trong khi cài đặt.

Tùy chọn	Mô tả
<code>-vv</code>	Cung cấp thông tin đầy đủ.
<code>-h</code>	Thỉnh thoảng hiển thị dấu # trong khi cài đặt, nhờ vào đó biết rằng RPM đang thực sự làm việc chứ máy không bị treo.
<code>--percent</code>	Thay vì hiển thị dấu #, tùy chọn này sẽ hiển thị tỷ lệ phần trăm khối lượng công việc đã thực hiện trong quá trình cài đặt.
<code>--test</code>	Không cài đặt gói phần mềm, nhưng thử nghiệm cài đặt thử và báo cáo lỗi.
<code>--replacefiles</code>	Thay thế các tệp từ những gói phần mềm khác.
<code>--force</code>	Lệnh cho RPM không quan tâm đến các lỗi do xung đột và cứ tiếp tục cài đặt.

Bảng 5.1: Các tùy chọn khi cài đặt

Muốn cài đặt một gói phần mềm từ máy khác, chúng ta có thể dùng giao thức tải FTP với một địa chỉ URL để xác định gói phần mềm ấy.

```
rpm-i ftp://ftp.netwharf.com/pub/RPMS/quota-1.55-4.i386.rpm
```

Câu lệnh này giả sử rằng chiếc máy từ xa chấp nhận chế độ FTP vô danh.

Ghi chú: Chúng ta có thể cùng lúc nhập username và mật khẩu vào dòng lệnh như sau:

```
rpm -i ftp://mark@ftp.netwharf.com/pub/RPMS/quota-1.55-4.i386.rpm
```

```
Password for mark@ftp.netwharf.com: <enter your password here>
```

Tuy nhiên đây không phải là cách an toàn để nhập lệnh, bởi vì có thể ai đó sẽ nhìn trộm, hoặc truy lục lệnh này từ tệp ghi chép lịch trình các câu lệnh của chúng ta.

3.3. Gỡ bỏ cài đặt gói phần mềm bằng RPM

Một trong những tiện lợi khi sử dụng RMP là việc cài đặt các chương trình mới rất dễ. Việc gỡ bỏ cài đặt cũng không có gì khó khăn. Chúng ta dùng tùy chọn `-e` như sau:

```
rpm -e quota -1.55.i386.rpm
```

Muốn gỡ bỏ một gói phần mềm ra khỏi hệ thống máy chúng ta, RPM phải qua các bước như sau:

-Kiểm tra tính phụ thuộc. RPM kiểm tra cơ sở dữ liệu xem có gói phần mềm nào khác phụ thuộc vào gói phần mềm ấy hay không. Nếu có, RPM sẽ không xoá, trừ khi chúng ta khẳng định là phải xoá.

- Chuẩn bị gỡ bỏ. RPM thi hành một script chuẩn bị cho việc gỡ bỏ cài đặt.

- Kiểm tra các tệp cấu hình. RPM lưu bản sao của mọi tệp cấu hình đã thay đổi.
- Xoá các tệp. RPM xoá tất cả các tệp kết hợp với gói phần mềm được xác định.
- Dọn dẹp. RPM thi hành một script dọn dẹp sau khi gỡ bỏ cài đặt.
- Cập nhật cơ sở dữ liệu. RPM gỡ bỏ tất cả mọi chỉ mục tham chiếu đến gói phần mềm đã tháo bỏ.

Cũng như đối với tùy chọn `-i`, chúng ta có thể sử dụng các tùy chọn `-v` và `-vv` để lấy thông báo đầy đủ từ lệnh `erase`. Chúng ta cũng có thể dùng tùy chọn `-test` để xem thử việc gì sẽ xảy ra khi gói phần mềm được gỡ bỏ cài đặt. Cuối cùng, tùy chọn `-nodeps` báo cho RPM cứ tiến hành gỡ bỏ cài đặt mà không cần quan tâm đến mọi phụ thuộc. Chúng ta nên cẩn thận với tùy chọn này, vì nếu chúng ta gỡ bỏ cài đặt một gói phần mềm đang có chương trình nào khác phụ thuộc vào nó thì sau này chương trình ấy sẽ không làm việc trơn tru được.

3.4. Cập nhật gói phần mềm bằng RPM

Với RPM, chúng ta được thuận lợi khi nâng cấp phần mềm bằng tùy chọn `-U` (viết hoa). Giả sử chương trình mang tên quota được một tác giả thêm vào nhiều chức năng và sau đó phổ biến lại với tên `quota-1.55-4.i386.rpm`. Muốn nâng cấp phiên bản cũ, chúng ta ra lệnh như sau:

```
rpm-U quota-1.55.4.i386.rpm
```

Trong khi nâng cấp, RPM cài đặt gói phần mềm đã xác định, sau đó xoá tất cả các phiên bản cũ. RPM cũng bỏ ra một khoảng thời gian khá lớn để xử lý các tệp kết hợp với gói phần mềm. Do đó trong khi RPM đang nâng cấp, có thể máy sẽ có thông báo như sau, cho chúng ta biết rằng một tệp cấu hình được lưu vào tệp mới:

```
Saving syslog.conf to syslog.conf.rpmsave
```

Điều này có nghĩa là RPM đã tạo ra một tệp cấu hình mới có khả năng tương thích với hệ thống máy chúng ta. Sau khi nâng cấp xong, chúng ta nên đối chiếu hai tệp cấu hình để chỉnh sửa tệp mới, nếu thấy cần thiết.

3.5. Tìm các gói phần mềm

Muốn biết những gói phần mềm nào đã cài đặt vào hệ thống, chúng ta dùng lệnh như sau:

```
rpm -qa
```

Lệnh sẽ hiển thị danh sách các gói phần mềm hiện có trên hệ thống. Muốn lấy thông tin từ một gói phần mềm nhất định, chúng ta cần gõ tùy chọn `-q`. Bảng sau liệt kê các tùy chọn mà chúng ta có thể sử dụng với nhóm lệnh `rpm -q` để tìm các gói phần mềm.

Tùy chọn	Mô tả
<code>-q</code> tên	Cung cấp tên, phiên bản và số phát hành của gói phần mềm.

-qa	Liệt kê tất cả các gói phần mềm đã cài đặt trên hệ.
-qf tệp	Tìm gói phần mềm liên kết với tệp.
-qp gói_phần_mềm	Tìm gói_phần_mềm.
-qi gói_phần_mềm	Cung cấp tên, mô tả, bản phát hành, kích cỡ, ngày tạo ra, ngày cài đặt và các thông tin khác về gói phần mềm.
-ql gói_phần_mềm	Liệt kê tất cả các tệp liên kết với gói phần mềm.

Bảng 5.2: Các tùy chọn tìm RPM

Cẩn thận: Các tùy chọn `-q` không chạy trơn tru khi chúng ta xác định một số kết nối tượng trưng (symbolic link). Để có kết quả tốt, chúng ta hãy chuyển đến thư mục thực sự của tệp trước khi sử dụng các tùy chọn `-q`.

Thí dụ chúng ta tìm thấy một gói phần mềm mới và muốn biết thêm thông tin, chúng ta gõ lệnh sau:

```
rpm -qip quota -1.55-4.i386.rpm
```

Lệnh sẽ hiển thị đại loại như:

Name: quota Distribution: Manhattan

Version: 1.55 Vendor: Red Hat software

Release: 9 Build Date: Thu May 7 22:45:481998

Install date: (not installed) Build Host: porky.redhat.com

Group: Utilities/SystemSourceRPM: quota-1.55-9.src.rpm

Size: 82232 Packager: RedHat Software bugs@redhat.com

Summary: Quota administration package

Description:

Quotas allow the system administrator to limit disk usage by a user and /or group per filesystem. This package contains the tools which are needed to enable, modify, and update quotas.

3.6. Kiểm tra gói phần mềm

Chế độ làm việc cuối cùng của RPM là kiểm tra xem xét lại gói phần mềm. Đến một lúc nào đó chúng ta sẽ muốn thử nghiệm tính nhất quán của một tệp trên hệ thống. Giả sử chúng ta nghi ngờ là có một tệp bị hỏng do một chương trình hoặc một user. Chúng ta muốn so sánh với những tệp nguyên thủy mà chúng ta đã cài đặt. RPM giúp chúng ta thực hiện việc này bằng tùy chọn `-V` (V viết hoa). Lệnh kiểm tra sẽ so sánh kích cỡ, tổng kiểm tra (checksum) MD5, lập cấu hình nhóm, loại tệp, người sở hữu tệp và permission (quyền truy cập hay phạm vi tác động đến tệp). Muốn biết xem từ khi cài đặt đến nay, một tệp nào đó của gói phần mềm có bị chỉnh sửa gì chưa, chúng ta gõ lệnh `rpm -V` với tên gói phần mềm. Thí dụ muốn kiểm tra gói phần mềm có tên quota, chúng ta gõ lệnh như sau:

```
rpm -V quota
```

Nếu không có gì thay đổi, RPM sẽ không hiển thị gì hết. Ngược lại RPM sẽ trình bày một dãy 8 ký tự cho thấy điều gì đã thay đổi và tên tệp đã thay đổi.

Lúc ấy chúng ta có thể xem lại một số tệp trong gói phần mềm để quyết định xem có phải cài đặt lại gói phần mềm ấy hay không. Bảng 6.4 liệt kê các mã báo lỗi hiển thị sau khi gõ lệnh.

Mã	Ý nghĩa
c	Tệp này là tệp cấu hình
5	Tệp này không qua được thử nghiệm tổng kiểm MD5
S	Kích cỡ tệp đã thay đổi sau thời điểm cài đặt
L	Có vấn đề với các mối liên kết tương trưng
T	Thời gian chỉnh sửa tệp không trùng với nguyên thủy
D	Thuộc tính thiết bị
U	Các thiết lập cho user đã thay đổi
G	Các thiết lập cho nhóm đã thay đổi
M	Chế độ

Bảng 5.3: Mã báo lỗi khi kiểm tra

3.7. Cài đặt phần mềm không của Linux

Rất tiếc là đa phần các chương trình không của Linux thì không ở dạng gói phần mềm RPM, nhất là các chương trình được tải từ cơ sở dữ liệu xuống qua cổng FTP vô danh.

Quá trình cài đặt những phần mềm đó rất khác nhau, đi từ mức độ hết sức đơn giản cho đến vô cùng khó khăn, hầu như không thể cài đặt được. Việc này tùy thuộc vào cách các tác giả viết ra các script cài đặt và tùy thuộc vào tư liệu về cài đặt của họ.

3.7.1. Các định dạng của gói phần mềm

Các gói phần mềm tải qua cổng FTP vô danh được ép thành tệp nén. Những tệp này được tạo ra bằng nhiều cách khác nhau. Thông thường thì cây thư mục chứa tệp nguồn, tư liệu, tệp thi hành và các tệp khác gộp chung thành tệp gộp (**tar file**) bằng chương trình **tar**. Sau đó tệp gộp được nén để chiếm ít chỗ.

Thường thì gói phần mềm có đuôi mở rộng giúp chúng ta biết tệp theo định dạng (format) nào. Nếu đó là .gz thì tệp được nén bằng chương trình **gzip** của GNU. Đây là định dạng tệp nén phổ biến nhất cho các gói phần mềm Linux. Nếu đuôi mở rộng là .Z, đó là tệp nén bằng chương trình **compress**. Thí dụ gói phần mềm foo.tar.gz là tệp gộp **tar**, được nén bằng **gzip**.

Ghi chú: Đôi khi một tệp **tar** được nén bằng **gzip** sẽ mang đuôi .tgz thay vì .tar.gz.

3.7.2. Cài đặt phần mềm

Sau khi xem qua định dạng của gói phần mềm, chúng ta quyết định đặt các tệp nguồn ở đâu để bắt đầu xây dựng gói phần mềm. Có những gói phần mềm rất lớn do đó chúng ta nên đặt chúng vào hệ thống tệp nào đó còn dư nhiều chỗ. Có người tạo ra hệ thống tệp riêng cho nguồn, sau đó **mount** (lắp đặt) vào một thư mục, chẳng hạn như /urs/local/src, hoặc /src. Tùy chúng ta muốn đặt vào đâu

cũng được, song hãy nhớ chừa đủ chỗ để cho phần mềm sau đó được biên dịch thành công.

Chúng ta di chuyển gói phần mềm đến cây nguồn đã thiết lập, sau đó bung ra toàn bộ. Đối với những tệp được nén bằng lệnh **gzip**, chúng ta bung ra bằng lệnh **gzip**, thí dụ:

```
gzip -d foo.tar.gz
```

sẽ bung ra tệp nén foo.tar.gz và thay thế bằng tệp gộp mang tên foo.tar. Chúng ta xem bảng 6.5 về các tùy chọn của lệnh **gzip**.

Flag	Tên Flag	Mô tả
-a	ascii	Văn bản dạng ASCII; chuyển đổi các ký tự cuối dòng bằng cách sử dụng những quy ước cục bộ.
-c	stdout	Ghi ra stdout (đầu xuất chuẩn, tức màn hình; sẽ giải thích sau), giữ các tệp nguyên thủy không thay đổi.
-d	decompress	Giải nén (bung ra)
-f	force	Khắc định ghi chồng lên tệp xuất và nén các mối liên kết.
-h	help	Liệt kê phần trợ giúp.
-l	list	Liệt kê nội dung tệp nén.
-L	license	Hiển thị bản quyền phần mềm
-n	no-name	Không lưu và không phục hồi tên và ngày giờ nguyên thủy.
-N	name	Lưu và phục hồi tên và ngày giờ nguyên thủy.
-q	quiet	Bỏ qua tất cả các nhắc nhở cảnh báo.
-s suf	suffix.suf	Sử dụng hậu tố suffix.suf vào các tệp nén.
-t	test	Thử nghiệm tính toàn vẹn của tệp nén.
-v	verbose	Chuyển sang chế độ có thông báo.
-V	version	Hiển thị số phiên bản
-1	fast	Nén nhanh hơn.
-9	best	Nén chặt hơn (nghĩa là tệp sẽ nhỏ hơn)
file		Xác định tệp nào cần thao tác; nếu bỏ trống, máy sẽ sử dụng stdin (đầu nhập chuẩn, tức bàn phím; sẽ giải thích sau)

*Bảng 5.4: Các tùy chọn của lệnh **gzip***

Đối với những tệp được nén bằng lệnh **compress**, chúng ta bung ra bằng lệnh **uncompress**, thí dụ:

```
uncompress foo.tar.z
```

sẽ bung ra tệp nén foo.tar.Z và thay thế bằng tệp gộp mang tên foo.tar.

Sau khi bung ra tệp nén xong, chúng ta chuyển tệp **tar** vào cây thư mục. Chúng ta sẽ đặt nguồn của từng gói phần mềm riêng rẽ vào thư mục của chúng trên cây thư mục.

3.7.3. Sử dụng lệnh tar

Chú ý rằng đối với lệnh **tar**, chúng ta có thể có 3 cách ghi tùy chọn:

- Gọi nhớ với 2 dấu trừ và tên tùy chọn. Thí dụ: **tar --help**
- Viết tắt với một dấu trừ và tùy chọn viết tắt. Thí dụ: **tar -h**
- Kiểu cũ: khi có nhiều tùy chọn và quen dùng chúng ta có thể viết gộp tất cả các tùy chọn cần sử dụng:. Thí dụ: **tar zxvf ttt**.

Tuy nhiên cách này dễ gây hiểu lầm và cho kết quả sai lệch. Thí dụ:

```
tar cvbf 20 /dev/rmt0
```

khác

```
tar -c -v -b 20 -f /dev/rmt0
```

bởi vì ở lệnh thứ hai thì 20 là trị số của tùy chọn -b và /dev/rmt0 là trị số của -f.

Nếu ta viết lại lệnh như sau: **tar cvbf /dev/rmt0 20**, kết quả sẽ khác nữa.

Trước khi giải nén tệp **tar**, chúng ta phải xem lại tệp đó có được tạo ra với một thư mục hay không và thư mục ấy có được coi là mục ghi đầu tiên hay không. Chúng ta dùng lệnh:

```
tar tvf tên_tệp_tar|more
```

để xem thư mục ghi đầu tiên trong tệp **tar** có phải là thư mục hay không. Nếu phải, tệp **tar** sẽ tạo ra thư mục khi được bung ra. Nếu không có thư mục ở cấp cao nhất của tệp **tar**, tất cả các tệp ở cấp cao nhất sẽ được giải nén vào thư mục hiện hành.

Trong trường hợp này, chúng ta phải tạo ra một thư mục và chuyển tệp **tar** vào đấy trước khi bung ra.

Ghi chú: Trước khi bung ra tệp **tar**, chúng ta nên kiểm tra xem có thư mục ở cấp cao nhất hay không. Trong trường hợp tệp **tar** bung ra vài trăm tệp vào thư mục hiện hành thì rắc rối lắm.

Một khi đã đặt tệp **tar** vào chỗ mà chúng ta muốn bung ra nó ra, chúng ta dùng lệnh như sau để bung ra cây nguồn vào tệp **tar**:

```
tar xvf tên_tệp_tar
```

Bước tiếp theo tùy thuộc vào cách viết chương trình của gói phần mềm mà chúng ta đang cài đặt. Thường thì chúng ta sẽ chuyển sang thư mục cấp cao nhất của nguồn phần mềm và tìm một tệp nào đó có dạng như README.1ST. Ở thư mục nguồn cấp cao thường có vài tệp tư liệu giải thích quá trình cài đặt.

Ghi chú: Với hầu hết các phiên bản Linux, chúng ta có thể giải nén một tệp **tar** luôn một thể khi khai thác nó. Chúng ta chỉ cần thêm flag z vào lệnh **tar**, chẳng hạn như:

```
tar zxvf foo.tar.gz
```

Quá trình cài đặt thường bao gồm việc hiệu chỉnh tệp Makefile để chỉnh sửa các thư mục đích. Phần mềm sẽ đặt các tệp nhị phân đã biên dịch vào những thư mục này. Thông thường chúng ta chạy lệnh **make** và tiếp theo đó là lệnh **make install**.

Quá trình thực hiện lệnh **make** có thể thay đổi (gọi các chương trình biên dịch hay cài đặt) theo từng gói phần mềm mà chúng ta cài đặt. Đối với vài gói phần mềm, một dạng shell script chuyên lập cấu hình sẽ yêu cầu chúng ta trả lời vài câu hỏi trước khi biên dịch phần mềm cho chúng ta. Chúng ta nên đọc trước các tệp tư liệu đi chung với gói phần mềm (nhất là tệp INSTALL).

3.8. Xem lại các quyền truy cập

Thông thường, việc thiết lập các quyền truy cập cho gói phần mềm sẽ diễn ra trong quá trình cài đặt. Mỗi ứng dụng sẽ có một script cài đặt đi kèm và script này sẽ cài đặt từng tệp với phần sở hữu và quyền truy cập tương ứng. Chỉ khi có trục trặc và khi có một user không thực hiện được những việc được quyền, thì lúc ấy chúng ta mới phải tìm ra thư mục nơi mà ứng dụng được chép vào và xem lại các quyền truy cập. Trên nguyên tắc thì tệp mà chúng ta dùng để kích hoạt ứng dụng đã bao gồm những quyền truy cập cho phép tất cả các user sử dụng ứng dụng đó. Chỉ superuser mới có khả năng xoá bỏ hoặc ghi đè lên. Thường thì ứng dụng được cài đặt vào một thư mục cho phép đọc và thi hành, chứ không ghi đè lên.

3.9. Giải quyết vấn đề

Một ứng dụng được viết tốt và hỗ trợ tốt sẽ được cài đặt vào hệ thống máy chúng ta mà không đòi hỏi chúng ta cung cấp nhiều thông tin. Ứng dụng tốt sẽ thiết lập các quyền truy cập đúng cách, sao cho công việc của chúng ta được đơn giản hoá, nghĩa là chúng ta chỉ cần chạy thử và báo cho user khác biết rằng ứng dụng mới đã khả thi rồi. Tuy nhiên không phải mọi thứ đều trơn tru. Nếu vì lý do nào đó mà chương trình không thể hoàn thành việc nạp tư liệu hoặc sau khi cài đặt xong mà vẫn không chạy tốt, chúng ta phải có trách nhiệm xác định lý do rồi sau đó đưa ra giải pháp.

Nếu một chương trình không được cài đặt hoàn hảo, công việc gỡ lỗi của quản trị viên thường chỉ là đọc tư liệu trợ giúp và các tệp README đi kèm với ứng dụng để xem qua danh sách các vấn đề thường gặp và giải pháp của chúng. Tuy nhiên chúng ta khó biết hết tất cả các phần mềm viết cho Linux, do đó phải cầu viện ở ngoài.

Nếu không thể gỡ lỗi bằng thông tin đi kèm gói phần mềm, chúng ta thử liên lạc với các nhóm tin Usenet để tham khảo những gì mọi người bàn bạc về phần mềm đó. Nếu gửi câu hỏi đến nhóm tin đúng, nhiều vấn đề của chúng ta sẽ được giải quyết. Nếu mạng Internet cũng không giúp gì được, chúng ta vẫn có thể liên lạc qua e-mail với người phát triển ứng dụng ấy. Xin nhắc chúng ta rằng Linux là phần mềm miễn phí, do đó hầu hết các gói phần mềm cho Linux cũng

không mất tiền. Vì vậy có lẽ chúng ta cũng không nên hy vọng luôn luôn nhận được sự trợ giúp dễ dàng.

3.10. Gỡ bỏ các ứng dụng

Nếu có gói phần mềm nào đó không còn được sử dụng hoặc đã có phiên bản tốt hơn, chúng ta nên gỡ bỏ nó khỏi hệ thống.

Cũng như khi cài đặt, gỡ bỏ một chương trình trên hệ Linux rắc rối hơn so với thao tác trên những hệ điều hành chỉ dành cho một user. Nhiều khi việc gỡ bỏ không đơn giản chỉ là xoá hết tệp của ứng dụng và sau đó xoá luôn thư mục. Các driver và những liên kết phần mềm khác cũng phải tháo ra để sau này không gặp rắc rối. Mỗi khi máy hiển thị lời nhắc hay lời cảnh báo, chúng ta nên ghi chúng vào một tệp nhật ký (logfile) để sau này tổng kết xem những gì đã đổi thay sau khi cài đặt. Từ đó chúng ta suy ra nên gỡ bỏ và thay đổi những tệp nào để cho việc gỡ bỏ một gói phần mềm được hoàn tất tốt.

4. Nâng cấp Kernel

Mục tiêu :

- Trình bày được các bước nâng cấp Kernel
- Nâng cấp được các phiên bản cập nhật của Kernel

Cùng với những bản nâng cấp và gói phần mềm khác, những phiên bản kernel Linux mới cũng được phát hành đều đặn. Những phiên bản này đã được sửa lỗi và tăng thêm chức năng. Chúng ta nâng cấp kernel để lập lại cấu hình hoặc để bổ sung driver cho các thiết bị mới.

Trước khi nâng cấp kernel, chúng ta nên sao lưu phần mềm hệ thống và dành một đĩa mềm khởi động Linux, phòng khi lỡ tay làm hỏng. Mô tả chi tiết về các thao tác xây dựng lại kernel cho Linux nằm ở chương “Lập cấu hình kernel Linux”.

Tiến trình nâng cấp kernel được mô tả chi tiết trong phần tư liệu HOWTO về Kernel. Tư liệu này được niêm yết đều đặn trên Internet, tại các website FTP Linux hoặc các nhóm thông tin Linux. Chúng ta nên sao chép một bản HOWTO để đọc kỹ trước khi nâng cấp kernel.

Bước đầu của tiến trình cơ bản nâng cấp kernel là truy cập các nguồn kernel mới qua FTP vô danh tại các website về Linux. Khi đã có nguồn mới, chúng ta cần lưu lại nguồn cũ. Chúng ta chuyển thư mục /usr/src/linux sang tên khác, chẳng hạn như /usr/src/linux.old. Chúng ta bung ra gói phần mềm nguồn kernel vào thư mục /usr/src và thư mục Linux sẽ được tự động tạo ra. Đến đây, chúng ta chuyển sang thư mục Linux để đọc các tư liệu và các tệp README.

Từ đây trở đi, tiến trình có thay đổi chút ít. Chúng ta gõ lệnh **make** config để chạy phần script về cấu hình và cung cấp thông tin hệ thống. Nếu phần này hoàn tất, chúng ta gõ lệnh đại loại như **make** dep (dependencies) để kiểm tra tình trạng phụ thuộc các tệp, nhằm đảm bảo kernel mới sẽ tìm ra tất cả các tệp cần thiết để biên dịch.

Sau công đoạn kiểm tra tính phụ thuộc, chúng ta gõ lệnh **make clean** để xoá mọi tệp đối tượng cũ rải rác ở thư mục kernel nguồn. Nếu đến thời điểm này mọi việc vẫn tốt, chúng ta yên tâm gõ **make** để biên dịch kernel mới. Biên dịch xong, chúng ta dùng chương trình quản lý môi LILO (LILO Boot Manager) để cài đặt kernel mới.

Xin nhắc chúng ta lần nữa là hãy đọc kỹ phần HOWTO của kernel trước khi bắt tay vào việc, như thế chúng ta sẽ tránh mọi phiền phức bực bội. Đồng thời cũng giúp chúng ta tránh tình trạng bỏ rác vào hệ Linux hiện hành.

5. Cài đặt trong môi trường X bằng RPM

Mục tiêu :

- Khởi động được phần mềm GNOME-RPM
- Chọn được gói phần mềm để cài đặt
- Lập được cấu hình mặc định cho trình cài đặt

Nếu như chúng ta không muốn nhớ các câu lệnh phức tạp ở trên mà muốn thực hiện các công việc một cách dễ dàng như trong môi trường Windows, chúng ta có thể sử dụng trình RPM của GNOME.

Tất cả các chức năng của RPM dạng dòng lệnh đều được đưa vào trong GNOME-RPM (gnorpm) dưới dạng các cửa sổ, menu, thanh công cụ hay hộp thoại.

5.1. Khởi động GNOME-RPM

Có thể khởi động Gnome-RPM theo một trong các cách sau:

- Trong môi trường GNOME, chọn lần lượt các mục sau Main Menu Button => Programs => System => GnoRPM
- Trong môi trường KDE, chọn lần lượt các mục sau Main Menu Button => Programs => System => GnoRPM
- Tại một terminal/console chúng ta gõ:

```
# gnor &
```

Ta sẽ làm việc với cửa sổ như sau:

Các gói phần mềm sẽ được trình bày thành nhóm/cây bên bảng bên trái và các gói phần mềm cụ thể trong từng nhóm được liệt kê trên bảng bên phải. Có thể chọn các tác động trong menu pop-up Operation hay trên thanh công cụ.

5.2. Chọn gói phần mềm

Khi chọn một nhóm phần mềm, danh sách các gói phần mềm trong nhóm hiện ra cho phép chúng ta chọn một hay nhiều gói phần mềm để tác động. Chúng ta giữ phím Ctrl (Control) và nhấp chuột trái lên biểu tượng (biểu tượng) của phần mềm cần chọn. Chúng ta cũng có thể giữ phím Shift và nhấp lên biểu tượng của phần mềm thứ nhất và biểu tượng của phần mềm cuối cùng nếu chúng ta chọn một nhóm các phần mềm nằm liên tiếp nhau.

Dòng thông báo trên thanh trạng thái ở bên dưới cửa sổ cho biết số lượng phần mềm mà chúng ta định chọn.

5.3. Cài đặt phần mềm mới

Chúng ta chọn Install trên thanh công cụ để cài đặt phần mềm mới.

Những gì chúng ta thấy trong cửa sổ Install sẽ phụ thuộc vào sự lựa chọn khi dùng Filter.

Theo mặc định, Red Hat sẽ tìm trong đường dẫn /mnt/cdrom/RedHat/RPMS để tách ra các gói phần mềm có thể được cài đặt. Chúng ta có thể thay đổi đường dẫn mặc định này trong Operations => Preferences.

Một thông tin ngắn về phần mềm chúng ta chọn sẽ hiện lên trong bảng Package Info.

Chọn Add nếu chúng ta muốn cài đặt phần mềm chúng ta đang chỉ định (highlight).

Chọn Query để biết thông tin về phần mềm đã chọn một cách chi tiết hơn. Chúng ta cũng có thể chọn Upgrade khi phần mềm được chỉ định đã được cài đặt trước (với phiên bản cũ hơn).

5.4. Lập cấu hình mặc định cho trình cài đặt

Khi nhập vào Operations => Preferences, chúng ta được phép xác định lại cách cài đặt mặc định cho trình GNOME-RPM.

Chúng ta không nên bật tùy chọn “no dependency checks”. Trong Linux, cũng như trong Windows, hiện nay, các phần mềm thường được xây dựng dựa trên các hàm thư viện có sẵn của Hệ điều hành hay của một số phần mềm khác.

Chúng ta cũng có thể định các cấu hình về Network với việc dùng Webfind, Rpmfind hay Distributions để cài đặt các gói phần mềm thông qua mạng hay Internet.

5.5. Gỡ bỏ phần mềm

Khi chọn Uninstall, chúng ta quyết định gỡ bỏ phần mềm đã chọn.

Màn hình Remove Packages sẽ nhắc chúng ta kiểm tra lại các phần mềm sẽ bị tháo bỏ.

Nếu bấm YES, tiến trình gỡ bỏ sẽ bắt đầu.

Khi chấm dứt, tên của các gói phần mềm sẽ không còn hiện lên ở bất kỳ cửa sổ nào.

B. CÂU HỎI VÀ BÀI TẬP

Câu 1: Trình bày những hiểu biết cơ bản của em về RPM để cài đặt các phần mềm cần thiết dùng cho Linux?

Câu 2: Nêu các chính sách nâng cấp phần mềm?

Câu 3: Trình bày cách cài đặt một số phần mềm cơ bản và cài đặt được những phiên bản sửa lỗi Kernel Linux?

Câu 4: Trình bày cách cài đặt trong môi trường X bằng RPM?

BÀI 6: QUẢN TRỊ HỆ THỐNG LINUX

Mã bài: MĐ37-06

❖ Giới thiệu

Bài này sẽ bàn về một số công việc và vấn đề chủ chốt của một quản trị viên hệ thống trong môi trường multiuser của Linux. Nếu đọc giáo trình này để tìm hiểu và cài đặt Linux, thì tức khắc gần như chúng ta đã trở thành quản trị viên hệ thống. Nhiều mục trong chương này sẽ thiên về quản trị hệ thống cho các mạng cơ quan. Tuy nhiên, cho dù chỉ là user duy nhất dùng Linux trên máy gia đình, chúng ta cũng nên làm quen với việc quản trị mạng lớn để mở rộng nhận thức về những vấn đề tổng quát hơn.

Những chủ đề chính sẽ được đề cập trong bài này bao gồm:

- *Tầm quan trọng của quản trị hệ thống*
- *Khái niệm multiuser*
- *Các hệ thống xử lý tập trung*
- *Các hệ thống xử lý phân tán*
- *Mô hình khách/chủ*
- *Quản trị trong môi trường mạng*
- *Xác định vai trò quản trị viên mạng.*

❖ Mục tiêu

- Mở rộng nhận thức tổng quát hơn những công việc của một quản trị viên hệ thống.
- Hiểu được các kiến thức cơ bản về các hệ thống xử lý, mô hình clien/sever, xác định vai trò của một quản trị viên mạng.
- Thiết lập được hệ thống mạng, thao tác các thiết bị ngoại vi và giám sát hệ thống cũng như nâng cấp phần mềm ở mức độ cơ bản nhất.

❖ Nội dung chính

A. LÝ THUYẾT

1. Các hệ thống và các thành phần xử lý

Mục tiêu :

- *Trình bày được các thành phần của hệ thống xử lý tập trung*
- *Trình bày được các thành phần của hệ thống xử lý phân tán*

1.1. Các hệ thống xử lý tập trung

Trên đà phát triển của công nghệ thông tin, trong những thập niên 1950 và 1960, các hệ điều hành multiuser đã ra đời, cho phép nhiều user chia sẻ tài nguyên từ các terminal riêng lẻ. Sử dụng batch-processing sequence (trình tự xử lý theo lô), hai user có thể thực thi hai chương trình khác nhau trong khi vẫn dùng chung một bộ xử lý trung tâm, một thiết bị lưu trữ và một thiết bị kết xuất. Với sự phổ biến của mạng điện thoại chuyển mạch (PSTN), máy tính bắt đầu sử dụng được nhiều tài nguyên tin học ở xa mà không phụ thuộc khoảng cách địa lý. Trong mô hình này, từng bộ xử lý sẽ sử dụng tài nguyên xử lý liên lạc để kết nối với các terminal ở xa. Từ đó đã phát triển ra mô hình front-end processing (xử lý mặt trước) về mặt liên lạc và mô hình xử lý tập trung về mặt dữ liệu.

Trước khi các máy tính được giảm giá mạnh mẽ và trở nên phổ biến khắp nơi thì hầu hết các hệ UNIX đều dùng mô hình xử lý tập trung. Với cách xử lý này, một máy tính lớn (mainframe) có thể phụ trách mọi xử lý. Các user chỉ việc kết nối tới đó và dùng chung tài nguyên của máy lớn. Hiện nay mô hình này càng ngày càng ít được dùng, mặc dù nó vẫn thích hợp cho những trường hợp các user ở cách nhau quá xa. Thí dụ cơ quan của chúng ta có một trung tâm xử lý chính và tất cả các chi nhánh từ xa đều có thể truy cập trung tâm này. Trên bàn của mỗi user là một terminal, bao gồm bàn phím và màn hình được kết nối với máy lớn và dùng chung những tài nguyên như máy in, bộ lưu trữ, vv...

Mô hình xử lý tập trung thường gồm nhiều thành phần như server, bộ xử lý mặt trước (front processor), terminal, modem và những bộ ghép nối nhiều cổng. Khi một user cần truy vấn từ xa, yêu cầu này được gửi về trung tâm và máy tính tại đây sẽ xử lý, sau đó trung tâm sẽ gửi kết quả về nơi yêu cầu. Mọi dữ liệu đều được xử lý và lưu trữ bởi máy lớn.

1.2. Các thành phần của mô hình xử lý tập trung

Muốn làm việc theo mô hình xử lý tập trung, chúng ta phải có một số thành phần như server, bộ xử lý mặt trước, terminal, modem và những bộ ghép nối nhiều cổng.

Có thể định nghĩa server như là một máy tính được thiết lập cấu hình nhằm chia sẻ tài nguyên với những máy khác. Thí dụ chúng ta có thể dùng một máy tính tương thích họ IBM PC làm server với điều kiện máy phải có đủ chỗ trên ổ cứng và đủ RAM.

Bộ xử lý mặt trước kết nối các kênh liên lạc với server và giữ vai trò thao tác các chi tiết liên lạc để server rảnh rang mà xử lý dữ liệu.

Terminal gồm có hai loại phổ biến, đó là terminal thường (dumb) và terminal thông minh (smart). Trước nay UNIX được sử dụng với terminal thường, vốn chỉ có bàn phím và màn hình mà thôi. Điểm nổi bật đối với terminal thường là chúng không có khả năng xử lý. Cổng liên lạc ở terminal thường được nối trực tiếp hoặc gián tiếp với server. Khi gõ bàn phím ở terminal thường, mỗi ký tự gõ vào đều được chuyển về server xử lý. Trong khi đó terminal thông minh có thể xử lý tại chỗ vài công việc đơn giản. Chẳng hạn như máy thu tiền hoặc máy rút tiền tự động là những terminal thông minh. Terminal thông minh lưu trữ yêu cầu giao dịch, sau đó mới chuyển toàn bộ yêu cầu này, thay vì chuyển từng ký tự mỗi lần gõ phím như terminal thường.

Muốn kết nối terminal với server qua mạng điện thoại, chúng ta cần hai modem. Modem thứ nhất ở đầu này chuyển đổi tín hiệu số của terminal (hoặc của máy tính) thành tín hiệu tương tự (analog) phù hợp cho đường điện thoại, modem thứ hai kết nối đường điện thoại ở đầu kia với server. Muốn làm việc từ xa, qua terminal chúng ta quay số điện thoại ở đầu kia và liên lạc được với server khi modem ở đầu kia nhận lời.

Muốn tăng số lượng các cổng cho user kết nối vào, chúng ta cần cài đặt một bộ ghép nối đa cổng. Thông thường một máy PC chỉ có hai cổng nối tiếp COM1 và COM2. Nếu muốn PC của mình trở thành server cho hơn hai user, chúng ta phải dùng bộ ghép nối đa cổng. Đó là một bìa để lắp đặt vào bus trong PC, đi cùng một cái hộp có tám ổ nối hoặc nhiều hơn và với một bó cáp nối bìa với hộp. Bộ ghép nối đa cổng còn đi kèm một phần mềm để giúp PC điều khiển các cổng nối tiếp đó.

1.3. Các hệ thống xử lý phân tán

Ở mô hình xử lý phân tán, terminal được thay thế bằng trạm làm việc (workstation), vốn là một máy vi tính chạy DOS hoặc Linux/UNIX, và chúng ta có thể chạy chương trình hoặc lưu thông tin trên server hay ở trạm làm việc cũng được. Sau khi xử lý tập ở trạm làm việc xong, chúng ta lưu trữ thông tin trên server để những user khác có thể truy cập. Chúng ta có thể tùy ý in từ máy in tại chỗ hoặc máy in kết nối với server.

Bởi vì hiện nay các trạm làm việc rất mạnh và rẻ tiền, có khả năng là cơ quan của chúng ta sử dụng hệ thống xử lý phân tán thay vì hệ thống xử lý tập trung.

1.4. Các thành phần của mô hình xử lý phân tán

Mô hình xử lý phân tán sử dụng các server tập, trạm làm việc, bìa giao diện mạng, cùng với một số thiết bị khác như bộ tập trung (hub), bộ khuếch đại (repeater), bộ cầu (bridge), bộ định tuyến (router) và bộ cổng ngõ (gateway). Chức năng của server tập là phân phối các tập và các đoạn chương trình đến các trạm làm việc. Hơn 90% công việc xử lý được tiến hành tại các trạm làm việc, chỉ có từ 5 đến 10% công việc dành cho server, chủ yếu là các nhiệm vụ quản trị.

Ngoài chức năng server tập, chúng ta có thể dùng PC như là một trạm làm việc Linux. Linux được thiết kế để chạy với một cấu hình phần cứng rất khiêm tốn: trước kia vào giữa thập niên 1990 chúng ta đã có thể chạy các bản Linux đầu tiên với CPU 386 SX và 4 MB RAM! Vì các máy tính hiện nay mạnh hơn hẳn yêu cầu tối thiểu của Linux, chúng ta có thể yên tâm, tất nhiên còn phải tùy theo phiên bản và nhà sản xuất Linux: như với RedHat, bản 5.x có thể chạy với 4 MB RAM còn bản 7.x thì chỉ chạy tốt khi bộ nhớ RAM lớn hơn 64 MB. Còn phải chưa bao nhiêu khoảng trống trên ổ cứng thì tùy vào việc chúng ta muốn cài đặt bao nhiêu phần mềm. Nếu chạy Linux hoàn toàn từ CD-ROM (thí dụ như bản SuSE Linux 7.3 cho CPU Intel) thì chúng ta chỉ cần 5 MB đĩa cứng. Nếu cài đặt từ đĩa cứng thì cần tối thiểu từ 80 đến 300 MB, còn cài trọn bộ thì cần khoảng 3.5 GB, tùy theo phiên bản và nhà sản xuất.

Nói chung, các tài nguyên cần dành ưu tiên cho trạm làm việc, nơi mà phần lớn công tác xử lý được thực thi. Tùy vào loại công việc dự định thực hiện, chúng ta sẽ dần dà đưa thêm tài nguyên vào. Thí dụ những phần mềm xử lý văn bản chiếm chẳng bao nhiêu tài nguyên (ổ cứng, RAM, chất lượng màn hình) khi so với những công việc xử lý hình ảnh, chẳng hạn như các chương trình CAD

(thiết kế bằng máy tính). Với những ứng dụng kiểu CAD, chúng ta cần những ổ cứng lớn (hơn 1 GB) và đầy đủ RAM (hơn 16 MB, có thể lên đến 512 MB), cùng với màn hình và bìa video độ phân giải cao (1280 X 1024 điểm hoặc nhiều hơn). Sau đó, chúng ta có thể cần đến một ổ băng từ để sao lưu và một ổ CD-ROM để đọc các chương trình lớn.

-NIC (bìa giao diện mạng) được gắn vào một khe trên bo mẹ (mother board) và thực hiện qua ổ nối (BNC hoặc RJ-45) mới liên kết vật lý giữa máy tính với dây cáp mạng. Chúng ta có thể nối mạng bằng cáp đồng trục hoặc cáp xoắn đôi.

-Hub (bộ tập trung) là nơi kết nối các cáp mạng, chẳng hạn loại hub khuếch đại thụ động (passive repeater hub) thường có bốn ổ nối RJ-45. Hub chuyển mạch (switching hub) thường có ít nhất tám cổng RJ-45. Loại Hub này vừa khuếch đại vừa chủ động chuyển mạch các tín hiệu.

-Repeater (bộ khuếch đại) có chức năng khuếch đại hoặc tái sinh tín hiệu trên mạng để tăng khoảng cách kết nối của cáp mạng.

-Bridge (bộ cầu) được dùng để nối hai mạng có giao thức và giao diện giống nhau.

-Router (bộ định tuyến) được dùng ở các mạng lớn và phức tạp, cho phép tạo lập và lựa chọn đường đi cho các gói tin đạt mục đích về cùng một địa chỉ trên mạng. Router sẽ xác định đâu là lộ trình tối ưu tại mỗi thời điểm và gửi gói tin theo lộ trình ấy.

-Gateway (cổng ngõ) sẽ chuyển đổi các giao thức cần thiết để cho hai mạng dùng các giao thức khác nhau liên lạc được với nhau. Thí dụ muốn liên lạc giữa các mạng TCP/IP, Netware hoặc AppleTalk thì đều phải qua trung gian gateway thích hợp.

2. Các mô hình và quản trị trong môi trường mạng

Mục tiêu :

- *Nêu được vai trò của mô hình client/server*
- *Lựa chọn được phần cứng và phần mềm*
- *Trình bày được những công việc chung trong quản trị mạng*

2.1. Mô hình client/server

Kết quả của việc xử lý phân tán là mô hình client/server (khách/chủ). Các hệ Linux có thể được dùng cho mô hình này với tư cách là khách, hoặc là chủ, hoặc cả hai.

Để tìm hiểu việc thiết lập quan hệ client/server, chúng ta hãy giả định nhiều trạm làm việc Linux (client) được kết nối với một server theo topo hình tuyến. Server này sẽ dành thư mục riêng cho từng client. Mỗi thư mục chứa nhiều tệp và tệp này lại được sao lưu đều đặn. Server cũng có những thư mục chung để các client có thể chia sẻ một số tệp với nhau. Thường thì một máy in laser được nối với server để mọi người dùng chung và một ổ băng từ cũng nối

như thế để sao lưu các ổ cứng dung lượng lớn. Ngoài ra, một số client còn có máy in riêng của mình được kết nối tại chỗ.

2.2. Quản trị trong môi trường mạng

Hình thức thường gặp của một mạng Linux/UNIX bao gồm nhiều máy tính lớn nhỏ khác nhau được kết nối bằng cáp mạng hoặc đường điện thoại. Quản trị mạng là công việc của một hoặc nhiều người tại một (số) trong những máy của mạng.

Hầu như ai cũng có khả năng tìm hiểu về Linux và quản trị một mạng máy vi tính. Với tính kiên trì cộng với nhiều thực hành, ngay cả những người với hiểu biết hạn chế về tin học cũng có thể học cách quản trị một mạng Linux.

2.3. Xác định vai trò quản trị viên mạng

Gần như nơi nào có một số máy chạy UNIX/Linux được nối mạng thì nơi ấy sẽ có một người chuyên trách mạng. Quản trị viên mạng cần có kiến thức nhất định để quyết định các kết nối vào hệ thống (qua mạng cục bộ hay modem), mức độ an ninh cần thiết và việc chia sẻ các thiết bị ngoại vi. Hàng ngày, quản trị viên kiểm tra danh sách tên hệ thống, địa chỉ mạng và việc vào ra của các user.

Những tổ chức lớn với mạng lưới hàng trăm máy vi tính thường có nhiều quản trị viên ở các lĩnh vực liên quan. Việc này là cần thiết nếu các user có nhu cầu chuyên môn, thí dụ về in ấn hoặc tạo mẫu đa dạng. Việc quản lý máy in và công tác in đòi hỏi kiến thức sâu rộng về những loại máy in nhất định và làm cách nào để cho chúng tương thích với các máy tính và phần mềm tác nghiệp.

2.4. Lựa chọn phần cứng và phần mềm

Là một quản trị viên hệ thống, chúng ta phải quan tâm đến nhiều yếu tố trước khi quyết định chọn phần cứng và phần mềm cho những hệ thống nằm trong phạm vi trách nhiệm của mình.

Nếu những hệ thống ấy ở cự ly gần và tập trung trong cùng một toà nhà, thì mạng cục bộ (LAN) là giải pháp vừa ít tốn kém vừa có tốc độ cao. Lúc này chúng ta chỉ cần gắn cho mỗi chiếc máy Linux một bìa giao diện mạng Ethernet có chạy giao thức TCP/IP, bởi vì TCP/IP là giao thức chuẩn cho các bản phát hành Linux.

Để kết nối tốc độ thấp ở cự ly lớn hơn, chúng ta nên dùng modem thoại chạy giao thức PPP (Point-to-Point Protocol), hoặc giao thức SLIP (Serial Line Internet Protocol, giao thức Internet đường nối tiếp), cho các kết nối TCP/IP không đồng bộ. Chúng ta cũng có thể dùng giao thức UUCP (UNIX-toUNIX Copy Protocol) cho thư điện tử, diễn đàn News và truyền tệp FTP (mặc dù UUCP có giới hạn nhất định).

Đối với cự ly dài hơn và tốc độ cao hơn, chúng ta chọn công nghệ ISDN hoặc xDSL hoặc thuê đường truyền riêng (Leased Line) của một công ty viễn thông phù hợp.

Chúng ta không nên mua bất kỳ thiết bị mạng nào đã qua sử dụng. Một số thiết bị mạng được bán cùng với phần mềm điều khiển (driver) giúp chúng chạy

với DOS, nhưng điều đó không có nghĩa là chúng sẽ chạy trơn tru với Linux. Hệ thống Linux có thể có nhiều driver chuẩn cho mạng được gắn kèm (built-in). Bảng sau liệt kê vài bìa Ethernet được Linux hỗ trợ. Chúng ta nên kiểm tra lại tệp HOWTO về Ethernet để theo dõi những cập nhật mới nhất.

Bảng 6.1: Vài loại bìa Ethernet được Linux hỗ trợ

Nhà sản xuất	Bìa
3Com	3c503, 3c503/16, 3c509, 3c579
SMC (Western Digital)	WD8003, WD8013, SMC Elite, SMC Elite Plus, SMC Elite 16 ULTRA
Novell Ethernet	NE1000, NE2000, NE1500, NE2100
D-Link	DE-600, DE-650, DE-100, DE-200, DE-220-T
Hewlett-Packard	27245A, 27247B, 27252A, 27247A, J2405A
Digital	DE200, DE210, DE202, DE100, DEPCA (rev.E)
Allied Telesis	AT1500, AT1700
PureData	PDUC8028, PI8023

Các phần mềm ứng dụng không được đóng gói chung với những sản phẩm về mạng vẫn có thể sử dụng ở môi trường mạng. Thí dụ chúng ta cài đặt một ứng dụng trên Linux, sau đó nhiều user trên những máy khác vẫn có thể dùng ứng dụng này bằng cách chạy các lệnh từ xa được viết cho UNIX. Một thí dụ khác: chúng ta có thể chia sẻ một ứng dụng bằng cách lắp ghép từ xa hệ thống tệp chứa ứng dụng ấy, sau đó lại chạy ứng dụng ngay từ hệ thống tại chỗ.

2.5. Những công việc chung trong quản trị mạng

Quản trị mạng có nhiều công việc khác nhau. Mạng máy tính không chỉ được thiết lập một lần rồi thôi mà nó luôn luôn phát triển. Khi mua phần cứng và phần mềm, quản trị viên cần hiểu rõ các user trông chờ điều gì ở mình và họ sẽ được những tiện ích nào.

2.5.1. Thiết lập hệ thống

Nếu là người dùng mạng, dù chỉ trong một phân đoạn (segment) Ethernet, chúng ta nên thực hiện các thử nghiệm về tính liên tác (interoperability). Khi kết nối xa qua mạng điện thoại, chúng ta cũng nên thử nghiệm đường truyền.

Khi là quản trị viên, chúng ta càng cần thử nghiệm toàn bộ việc nối các cáp, thiết bị mạng và thiết bị tin học. Phần mềm mạng phải được cài đặt và sẵn sàng kết nối. Ngay cả khi hệ thống có các tiện ích Plug and Play (cắm là chạy) chúng ta cũng cần kiểm tra lại chúng.

Điều có lợi khi mua một máy chưa cài đặt sẵn hệ điều hành là chúng ta có cơ hội thiết lập hệ thống tệp sao cho phù hợp với yêu cầu riêng biệt của mình. Chúng ta biết hệ thống mạng sẽ dùng phần mềm nào, cũng như tổng số user và cường độ sử dụng máy của họ.

Lưu ý: để khỏi tốn nhiều thời gian và công sức vào việc thiết lập lại mạng, chúng ta cần sao lưu tức khắc các tệp cấu hình ngay sau khi thiết lập một hệ thống.

Khi thiết lập Linux xong và thấy hệ thống chạy tốt, chúng ta sẽ cài các phần mềm ứng dụng. Như đã biết, phần mềm chạy với Linux thường phức tạp hơn các hệ đơn dụng, do đó chúng ta nên thử những phần mềm ấy nhiều lần sau khi cài đặt để đảm bảo rằng chúng chạy trơn tru, trước hết là các tiện ích.

Đến đây, mặc dù chưa nối mạng song chúng ta vẫn cần chuẩn bị đưa danh sách user vào hệ thống. Chúng ta phải cấp các định danh đăng nhập (login ID) cho một vài user chính, cùng với mật khẩu chung ban đầu. Việc này vừa đảm bảo an ninh ban đầu, vừa tạo cơ hội cho những user quan trọng sử dụng hệ thống ngay sau khi cài đặt xong.

Sau khi cài đặt, chúng ta nên nối mạng ngay để chắc chắn rằng mình có thể liên lạc với mọi nơi. Tiếp theo chúng ta thử chuyển một số tệp lớn nhỏ khác nhau từ máy này sang máy khác. Thư điện tử phải được thông suốt. Tất cả các máy phải nhận diện được mỗi máy mới nối vào hệ thống, có nghĩa là chúng ta phải thêm tên của máy mới vào cơ sở dữ liệu host name. Nếu đang sử dụng dịch vụ tên miền tại chỗ, chúng ta phải thêm host name vào cơ sở dữ liệu DNS. Nếu không dùng DNS thì chúng ta thêm tên vào các tệp /etc/hosts trên các hệ thống khác.

2.5.2. Thao tác các thiết bị ngoại vi

In ấn là một thao tác phổ biến mà quản trị viên cần quan tâm chuẩn bị sẵn sàng. Theo dõi, bảo trì máy in là một công việc khá quan trọng và chiếm nhiều thời gian của quản trị viên. Người quản trị phải biết quy trình in ấn và thiết lập các cấu hình thích hợp với đặc điểm của từng loại thiết bị.

Còn modem là phương tiện ít tốn tiền nhất để nối mạng ở cự ly lớn. Modem làm việc với giao thức PPP hoặc UUCP có thể giúp một nhóm nhỏ quản trị viên trông coi được nhiều máy tính. Tuy nhiên cũng như máy in, modem cần được chăm sóc từ đầu để luôn luôn chạy tốt. Chúng ta nên chọn thử một vài modem có thương hiệu nổi tiếng và tìm hiểu kỹ các tính năng của chúng.

2.5.3. Giám sát hệ thống

Khi cài đặt xong UNIX/Linux, chúng ta có thể thiết lập các công cụ để giám sát hệ thống mới. Việc giám sát một hệ thống đang hoạt động là công việc thường trực của quản trị viên, song gánh nặng này sẽ ổn định sau một thời gian (dĩ nhiên nếu chúng ta không liên tục bổ sung vào mạng nhiều máy mới hoặc phần mềm mới). Thỉnh thoảng cũng có thiết bị nào đó ngưng hoạt động hoặc cần điều chỉnh. Một quản trị viên giỏi là người biết rõ vấn đề phát sinh do phần cứng hay phần mềm và cách giải quyết.

2.5.4. Nâng cấp phần mềm

Có vài gói phần mềm trong Linux cần được cập nhật thường xuyên. Quản trị viên phải lo việc này, bởi vì Linux được cấp miễn phí trên Internet và luôn

được chỉnh sửa. Việc chỉnh sửa nhằm để khắc phục những lỗi đã phát hiện, nhưng chúng ta lại mất thời gian tải nạp bản cập nhật và cài đặt nó trên các máy.

Chúng ta không nên cài đặt hết mọi phiên bản mới vào toàn bộ các máy, mà trước hết hãy thử nghiệm bản nâng cấp trên một máy. Khi nào chắc chắn rằng bản nâng cấp chạy tốt, lúc ấy hãy cập nhật nó vào những máy khác.

Quản trị viên giỏi là người biết cài đặt những phần chỉnh sửa hoặc những phiên bản mới mà không cần phải thân chinh đi đến từng máy trong mạng. Mới nghe qua điều này có vẻ khó tin, song UNIX có những công cụ giúp chúng ta thực hiện công tác này.

2.6. Huấn luyện quản trị viên

Thông thường các cơ quan đều có người biết sử dụng máy vi tính, nhưng lại hiếm có quản trị viên hệ thống chuyên nghiệp.

Công tác quản trị UNIX/Linux đòi hỏi có kiến thức vững chắc về một số chủ đề chung sau đây:

-Thiết kế và sử dụng các hệ Linux/UNIX. Quản trị viên phải có hiểu biết xuyên suốt về một số kỹ thuật như đổi hướng (redirection), ống dẫn (pipeline), xử lý hậu trường (background processing), v.v.

-Trình soạn thảo văn bản vi. Hầu như bất kỳ chiếc máy tính nào được cài đặt UNIX/Linux thì đều có sẵn trình soạn thảo vi. Nhiều người không thích vi và chuyển sang dùng các trình soạn thảo khác thân thiện hơn, tuy nhiên quản trị viên nên có kiến thức và kỹ năng sử dụng vi bởi vì đó là mẫu số chung của các hệ UNIX/Linux.

-Lập trình shell script. Phần lớn chương trình dùng để quản trị UNIX/Linux được viết theo ngôn ngữ kịch bản shell script nhưng chúng ta nên sửa đổi chúng chút ít cho phù hợp nhu cầu của mình. Nhiều công cụ được nhắc đến ở đây đòi hỏi chúng ta phải biết phối hợp và sử dụng các chương trình shell. Mỗi user thường thích sử dụng loại shell quen thuộc nhất của mình. Thí dụ bash là một bản nâng cấp rất mạnh từ bản Shell Bourne mà Linux dùng làm shell mặc định. Ngoài ra hai bản khác là Z shell và T shell cũng hay được kèm theo các bản phát hành Linux. Tuy nhiên ở bước đầu chúng ta hãy dùng Shell Bourne cho thật thuần thục. Nên biết rằng các chương trình kịch bản của những đồng tác giả Linux đều được viết trong Shell Bourne. Chúng ta cũng nên tìm hiểu ngôn ngữ quản trị hệ thống Perl. Ngôn ngữ này giúp chúng ta một bộ công cụ rất tốt để quản trị hệ thống trong môi trường lập trình.

-Liên lạc. Muốn quản trị mạng máy tính cho thật hiệu quả, điều chủ yếu là chúng ta phải có kiến thức về TCP/IP và các giao thức có liên quan. Tương tự, chúng ta sẽ phải hiểu biết về PPP nếu muốn thiết lập một kết nối Internet không đồng bộ. Tất cả các giao thức ấy có thể học ngay tại môi trường thực hành với nhiều tùy chọn sẵn có. Đương nhiên chúng ta vẫn có thể ghi danh học tại các lớp lý thuyết hoặc tự mua sách về học, song nếu như thế chúng ta phải mất thêm thời gian thử nghiệm.

-Các kiến thức cơ bản về UNIX. Trong nhà trường, các kiến thức cơ bản về UNIX thường không được dạy, thậm chí còn không được đề cập đến và chúng ta sẽ phải nắm bắt chúng trong quá trình thực tế. Thí dụ chúng ta sẽ biết rằng các tệp thi hành nhị phân thường được lưu trữ tại các thư mục bin, chẳng hạn như /usr/bin, hoặc /bin, hay là /usr/local/bin. Tương tự, những thư mục lib, chẳng hạn như /usr/lib được dùng để chứa tệp thư viện, do đó chúng ta sẽ đưa các thư viện riêng của mình vào một thư mục đại loại như /usr/local/lib. Nắm được và làm theo những kiến thức cơ bản về UNIX/Linux, chúng ta sẽ tiết kiệm được thì giờ khi dò tìm và giải quyết nhiều vấn đề phức tạp hơn.

Chúng ta nên theo những chương trình đào tạo thực hành, tức là học đến đâu làm đến đấy. Học xong một bài nào đó ở lớp, chúng ta cần áp dụng ngay điều mình vừa học vào thực tiễn.

B. CÂU HỎI VÀ BÀI TẬP

Câu 1: Trình bày những công việc của một quản trị viên hệ thống?

Câu 2: Em hãy xác định vai trò của một quản trị viên mạng?

Câu 3: Trình bày các mô hình và quản trị trong môi trường mạng?

Câu 4: Nêu những công việc chung trong quản trị mạng?

Câu 5: Thực hành các tập tin cấu hình mạng?

Hướng dẫn thực hành:

1. Tập tin /etc/hosts

Do not remove the following line, or various programs

that require network functionality will fail. 127.0.0.1 localhost.localdomain
localhost 200.201.202.1 linuxsvr.dng.vn linuxsvr

2. Tập tin /etc/sysconfig/network

NETWORKING=yes FORWARD_IPV4=false HOSTNAME=linuxsvr.edu.vn
DOMAIN=edu.vn GATEWAY=200.201.202.1

3. Tập tin /etc/sysconfig/network-scripts/ifcfg-eth0

DEVICE=eth0

BOOTPROTO=none ONBOOT=yes USERCTL=no PEERDNS=no

TYPE=Ethernet IPADDR=200.201.202.1

NETMASK=255.255.255.0

NETWORK=200.201.202.0

BROADCAST=200.201.202.255

4. Chạy chương trình X- Windows hỗ trợ cấu hình hệ thống :

redhat-config-network

5. Khởi động lại dịch vụ mạng

[root@linuxsvr root]#/etc/init.d/network restart

Shutting down interface eth0: [OK] Shutting down loopback

interface: [OK] Setting network parameters: [OK]

Bringing up loopback interface:

[OK]

[OK] Bringing up interface eth0:

6. Kiểm tra bằng lệnh :

```
[root@linuxsvr root]#hostname
```

```
linuxsvr.dng.vn
```

7. Xem thông tin về cấu hình thiết bị mạng

```
[root@linuxsvr root]#ifconfig
```

```
eth0  Link encap:Ethernet HWaddr 00:06:7B:02:71:21
```

```
inet addr:200.201.202.1 Bcast:200.201.202.255 Mask:255.255.255.0
```

```
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
```

```
RX packets:2326 errors:0 dropped:0 overruns:0 frame:0
```

```
TX packets:70927 errors:0 dropped:0 overruns:0 carrier:0 collisions:0
```

```
txqueuelen:100
```

```
RX bytes:218392 (213.2 Kb) TX bytes:6939053 (6.6 Mb) Interrupt:9 Base  
address:0x4c00
```

```
lo    Link encap:Local Loopback
```

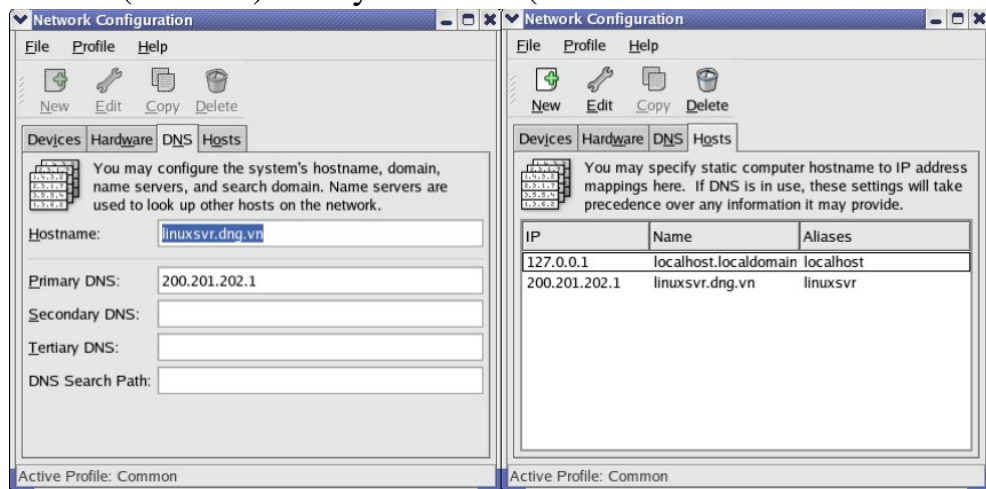
```
inet addr:127.0.0.1 Mask:255.0.0.0
```

```
UP LOOPBACK RUNNING MTU:16436 Metric:1
```

```
RX packets:933 errors:0 dropped:0 overruns:0 frame:0
```

```
TX packets:933 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:0
```

```
RX bytes:87261 (85.2 Kb) TX bytes:87261 (85.2 Kb)
```



Cấu hình dịch vụ mạng bằng tiện ích redhat-config-network

Câu 6: Thực hành cấu hình dịch vụ DNS?

Hướng dẫn thực hành:

1. Các tập tin cấu hình dịch vụ DNS

a. Tập tin /etc/host.conf

order hosts,bind

b. Tập tin /etc/resolv.conf

:search dng.vn

nameserver 200.201.202.1

c. Tập tin /etc/named.conf


```

# named.conf - configuration for bind
# Generated automatically by redhat-config-bind, alchemist et al.
# Any changes not supported by redhat-config-bind should be put
# in /etc/named.custom
controls {
inet 127.0.0.1 allow { localhost; } keys { rndckey; };
};
include "/etc/named.custom"; include "/etc/rndc.key";
zone "0.0.127.in-addr.arpa" { type master;
file "0.0.127.in-addr.arpa.zone";
};
zone "localhost" { type master;
file "localhost.zone";
};
zone "dng.vn" { type master;
file "dng.vn.zone";
};
zone "edu.vn" { type master;
file "edu.vn.zone";
};
d. Tập tin /var/named/dng.vn.zone
$TTL 86400
@ IN SOA dng. root.localhost ( 1 ; serial
28800 ; refresh 7200 ; retry
604800 ; expire 86400 ; ttl
)
IN NS 200.201.202.1
www IN A 200.201.202.1
tankhoi01 IN A 200.201.202.1
tankhoi02 IN A 200.201.202.2
e. Tập tin /var/named/edu.vn.zone
$TTL 86400
@ IN SOA edu. root.localhost ( 2 ;
serial
28800 ; refresh
7200 ; retry
604800 ; expire
86400 ; ttl
)
IN NS 200.201.202.1.
www IN A 200.201.202.1
tankhoi01 IN A 200.201.202.1 tankhoi02 IN

```

```

A      200.201.202.2
g. Tập tin /var/named/0.0.127.in-addr.arpa.zone
$TTL 86400
@      IN      SOA  localhost.  root.linuxsvr.dng.vn ( 36 ;
        serial
        28800 ; refresh
        7200 ; retry
        604800 ; expire
        86400 ; ttk
        )

```

```

@      IN      NS      localhost.
1      IN      PTR     localhost.
1      IN      PTR     www.
1      IN      PTR     tankhoi01.
2      IN      PTR     tankhoi02.
1      IN      PTR     www.
1      IN      PTR     tankhoi01.
2      IN      PTR     tankhoi02.

```

```

h. Tập tin /var/named/localhost.zone
$TTL 86400
@      IN      SOA  @ root.localhost ( 1 ; serial
28800 ; refresh
7200 ; retry 604800 ; expire 86400
        IN      NS      localhost.
        @      IN      A      127.0.0.1

```

i. **Lệnh khởi động dịch vụ DNS**

```
/etc/init.d/named restart
```

2. **Các lệnh và tiện ích hỗ trợ**

a. **Lệnh nslookup**

```
#nslookup
```

Note: nslookup is deprecated and may be removed from future releases.

Consider using the `dig' or `host' programs instead. Run nslookup with the `-sil[ent]' option to prevent this message from appearing.

```
> www.dng.vn
```

```

Server:      200.201.202.1
Address:     200.201.202.1#53
Name:        www.dng.vn Address: 200.201.202.1
> tankhoi02.edu.vn
Server:      200.201.202.1
Address:     200.201.202.1#53
Name:        tankhoi02.edu.vn Address: 200.201.202.2

```

b. Lệnh host

```

#host tankhoi01.dng.vn
tankhoi01.dng.vn has address 200.201.202.1

```

c. Lệnh dig

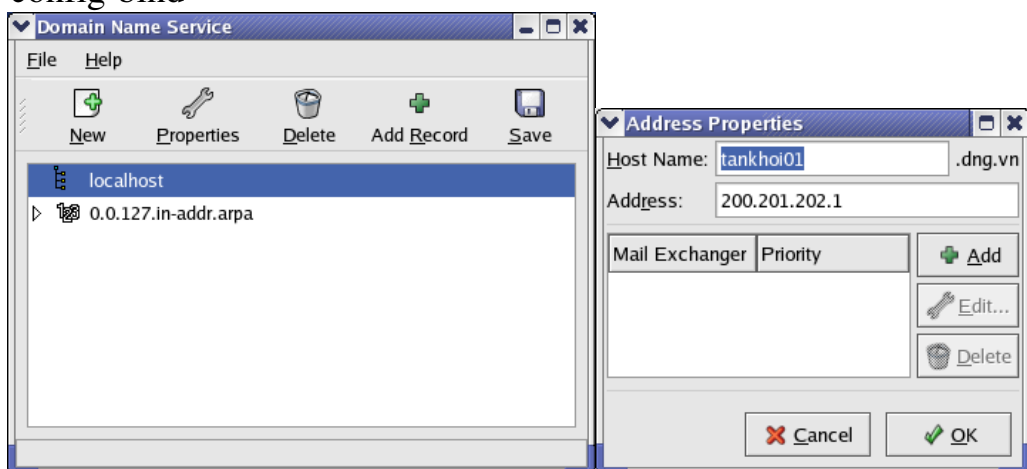
```

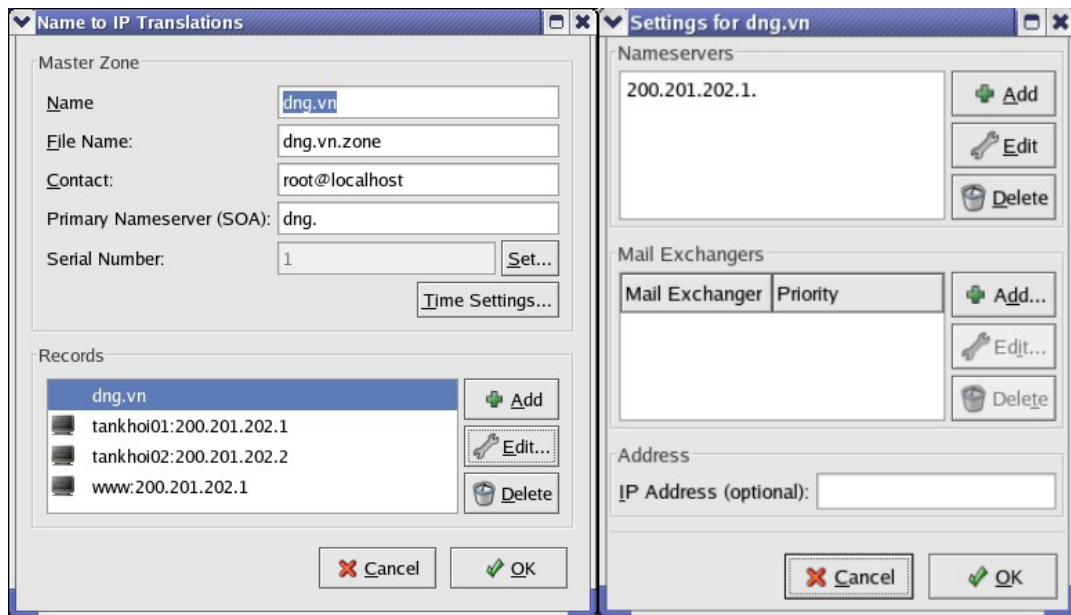
# dig dng.vn
; <<> DiG 9.2.1 <<> dng.vn
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 58922
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0
;; QUESTION SECTION:
;dng.vn.      IN      A
;; AUTHORITY SECTION:
dng.vn.      86400IN    SOA     dng. root.localhost.dng.vn. 1 28800 7200 604800
86400
;; Query time: 28 msec
;; SERVER: 200.201.202.1#53(200.201.202.1)
;; WHEN: Mon Mar 22 09:14:13 2004
;; MSG SIZE rcvd: 78

```

d. Tiện ích redhat-config-bind

```
#redhat-config-bind
```





Cấu hình dịch vụ BIND bằng tiện ích redhat-config-bind.

BÀI 7: KHỞI ĐỘNG VÀ ĐÓNG TẮT

Mã bài: MĐ37-07

❖ Giới thiệu

Hai thao tác bắt buộc và thông thường nhất đối với một quản trị viên là khởi động và đóng tắt hệ thống. Tuy nhiên muốn khởi động Linux thì những thao tác đó cần được tiến hành đúng hướng dẫn.

❖ Mục tiêu

- Trình bày được trình quản lý mỗi LILO.
- Trình bày được trình quản lý mỗi GRUB.
- Nêu được tiến trình khởi động.
- Nêu được các bước mỗi Linux bằng đĩa mềm.
- Khởi động được bằng trình mỗi.
- Đóng tắt được Linux.

❖ Nội dung chính

A. LÝ THUYẾT

1. Trình quản lý môi LILO

Mục tiêu :

- Thiết lập được cấu hình LILO
- Sử dụng được LILO

LILO là một trình môi nằm trong các bản phát hành Red Hat Linux và là trình môi mặc định của các phiên bản Red Hat trước 7.1, từ phiên bản 7.2, ta có thêm trình môi GRUB. Nếu muốn sử dụng GRUB thay vì LILO, chúng ta có thể bỏ qua mục này và chuyển sang mục sau (trình quản lý môi GRUB).

Chúng ta tùy ý cài đặt LILO vào đĩa cứng hoặc đĩa mềm. Cách dễ nhất để cài đặt LILO là dùng chương trình cài đặt của Red Hat hoặc Caldera, vì hai chương trình này giúp chúng ta tự động thực hiện một phần lớn công việc theo menu (thực đơn).

Lưu ý: Chúng ta nên cài đặt LILO bằng chương trình cài đặt của Red Hat hoặc Caldera. Cài đặt trình môi là một tiến trình nhiều rủi ro, dễ gây ra hỏng hóc dữ liệu trên ổ cứng nếu làm không đúng cách (xem “Cài đặt LILO”).

Khi cài đặt LILO xong, chúng ta có thể chọn ngay hệ điều hành mong muốn ở thời điểm môi. Nếu không, LILO sẽ đếm giá trị timeout (thời gian chờ đợi), sau đó sẽ tự động môi cho hệ điều hành mặc định khởi động.

1.1. Thiết lập cấu hình LILO

LILO đọc thông tin chứa ở tệp cấu hình /etc/lilo.conf để biết xem trong máy chúng ta có những hệ điều hành nào và các thông tin khởi động đang nằm ở đâu. LILO được lập cấu hình để môi các đoạn thông tin của tệp trên cho từng hệ điều hành.

Sau đây là thí dụ về ba đoạn của một tệp /etc/lilo.conf. Đoạn 1:

```
boot = /dev/hda
map = /boot/map
install = /boot/boot.b
prompt
timeout = 50
message = /boot/message
lba32
default = linux
```

Đoạn 2:

```
image = /boot/vmlinuz-2.4.0-0.43.6
label=linux
initrd = /boot/inditrd-2.4.0-0.43.6.img
read-only
root = /dev/hda5
```

Đoạn 3:

other = /dev/hda1

label = dos

Đoạn thứ nhất cho LILO biết:

- Trình mỗi nằm ở ổ đĩa cứng /dev/hda/
- Kiểm tra tệp map trong thư mục /boot/map
- Nạp các thông báo trong tiến trình mỗi từ tệp /boot/message
- Có thể cài đặt một tệp đặc biệt (/boot/boot.b) như là một rãnh mỗi mới
- Ổ đĩa cứng đang hỗ trợ LBA32 (dòng này thường có giá trị là linear, không nên đổi lại dòng này nếu chúng ta chưa biết rõ đĩa cứng của mình; chúng ta có thể tìm hiểu đĩa cứng có hỗ trợ LBA32 hay không bằng cách kiểm tra thông tin BIOS)
- Hệ điều hành mặc định là Linux
- Thời gian chờ trước khi nạp hệ mặc định là 5 giây (đơn vị tham số được tính bằng 1/10 của giây)

Đoạn thứ hai cho LILO biết:

- lõi Linux đang ở tệp /boot/vmlinuz-2.4.0-0.43.6
- Linux là tên hệ điều hành sẽ xuất hiện tại **menu** khởi động của LILO
- Vị trí root của hệ thống tệp Linux đang ở ổ đĩa cứng /dev/hda5

Đoạn thứ ba cho LILO biết:

- Có phân vùng của một hệ điều hành nữa đang ở ổ đĩa cứng /dev/hda1
- DOS là tên hệ điều hành thứ hai (còn dùng để chạy Windows)

1.2. Sử dụng LILO

Khi cài đặt LILO, chúng ta sẽ đặt giá trị timeout và hệ điều hành mặc định. Giá trị này cho phép chúng ta suy nghĩ trong một thời gian ngắn trước khi hệ mặc định tự khởi động, xem có nên đổi sang một hệ điều hành khác hay không. Sau thời gian timeout nếu chúng ta không can thiệp gì cả, LILO sẽ khởi động hệ điều hành mặc định.

Từ RedHat 7.x trở đi, ta có một **menu** với giao diện đồ họa để tiện cho việc chọn hệ điều hành.

Khi bật nút điện một máy tính có LILO ở đĩa cứng hay ở đĩa mềm, màn hình LILO: sẽ xuất hiện. Đến đây chúng ta có nhiều tùy chọn. Hoặc chúng ta chờ LILO khởi động hệ điều hành mặc định, hoặc chúng ta bấm <Ctrl>, <Alt> hay <Shift> cho LILO khởi động tức khắc. Chúng ta có thể nhấn <Enter> khi thấy tên hệ điều hành mà chúng ta muốn LILO khởi động ngay đang được chọn.

Sau đó việc khởi động bắt đầu được chuyển từ LILO về cho hệ điều hành quản lý. Nếu máy chúng ta không cài đặt GRUB, chúng ta có thể xem tiếp mục 9.4 để hiểu hơn về tiến trình khởi động ở mức hệ điều hành.

2. Tiến trình khởi động

Mục tiêu :

- Trình bày được các bước khởi động hệ thống Linux

Red Hat và phần lớn các bản phát hành sau này của Linux sử dụng tiến trình khởi động mang tên SysV init thay vì kiểu BSD init cũ. SysV init là chương trình đầu tiên mà kernel thực thi ngay từ lúc khởi động máy, do đó init được mang số định danh tiến trình (PID) số 1, trở thành tiến trình “mẹ” của tất cả các tiến trình khác chạy dưới Linux. PID của một tiến trình là mã số mà hệ điều hành dùng để nhận diện tiến trình ấy. Nhiều lệnh của Linux dùng PID ấy làm tham số.

Khi khởi động, Linux đi theo những bước sau:

- Kernel chạy SysV init, chương trình này nằm trong thư mục /sbin.
- SysV init chạy shell script /etc/rc.d/rc.sysinit.
- rc.sysinit lập các biến hệ thống khác nhau và thực hiện các thao tác ban đầu.

đầu.

- SysV init chạy shell script /etc/rc.d/rc.serial.
- rc.serial sử dụng một số lệnh setserial để thiết lập cấu hình của các cổng serial cho hệ thống. Chúng ta có thể xem trang **man** của lệnh setserial để biết cách sử dụng.

- SysV init chạy tất cả các script được quy ước là mặc định.

- SysV init chạy script /etc/rc.d/rc.local.

Chương trình này khởi động nhiều tiến trình khác, đồng thời chuyển thông báo cho thiết bị và cho tệp đăng nhập hệ thống /var/log/messages biết về trạng thái của từng tiến trình đã khởi động.

Lưu ý: Tệp đăng nhập hệ thống /var/log/messages là để giúp chúng ta giải quyết các vấn đề có thể gặp khi khởi động. Kernel lưu tất cả những thông báo lỗi trong tệp này, do đó chúng ta khởi mất công ghi chép lại khi chúng xuất hiện (nhưng nhanh chóng bị xoá mất).

Sau đây là một trình tự khởi động điển hình:

April 1 23:23:42 ns syslogd 1.3-3: restart.

April 1 23:23:43 ns kernel: klogd 1.3-3, log source = /proc/kmsg started.

April 1 23:23:45 ns kernel: Loaded 4189 symbols from /boot/System.map.

April 1 23:23:45 ns kernel: Symbold match kernel version 2.0.31.

April 1 23:23:45 ns kernel: Loaded 2 symbols from 3 modules.

April 1 23:23:45 ns kernel: Console: 16 point font, 400 scans

April 1 23:23:45 ns kernel: Console: colour VGA+ 0x25, 1 virtual console (max 630

April 1 23:23:45 ns kernel: pci_init: no BIOS32 detected

April 1 23:23:45 ns kernel: Calibrating delay loop.. ok - 49.97 BogoMIPS

April 1 23:23:45 ns kernel: Memory: 30816k/32768k available (736k kernel code, 384k reserved, 382k data)

April 1 23:23:45 ns kernel: This processor honours the WP bit even when in supervisor mode. Good.

April 1 23:23:45 ns kernel: Swansea University Computer Society NET3.035.

April 1 23:23:45 ns kernel: Swansea University computer society TCP/IP for NET3.034.
April 1 23:23:45 ns kernel: IP Protocols: IGMP, ICMP, UDP, TCP
April 1 23:23:45 ns kernel: VFS: Diskquotas version dquot_5.6.0 initialized
April 1 23:23:45 ns kernel:
April 1 23:23:45 ns kernel: Checking 386/387 coupling... ok,
fpu using exception 16 error reporting.
April 1 23:23:45 ns kernel: Checking 'hlt' instruction... ok.
April 1 23:23:45 ns kernel: Linux version 2.0.31
(root@porky.redhat.com) (gcc version 2.7.2.3) #1 Sun Now o 21:45:23 EST 1997
April 1 23:23:45 ns kernel: Starting kswapd v 1.4.2.2
April 1 23:23:45 ns kernel: Serial driver version 4.13 with no serial options enabled
April 1 23:23:45 ns kernel: tty00 at 0x03f8 (irq = 4) is a 16550A
April 1 23:23:45 ns kernel: tty01 at 0x02f (irq = 3) is a 16550A
April 1 23:23:45 ns kernel: Real Time Clock Driver v1.07
April 1 23:23:45 ns kernel: Ramdisk driver initialized : 16 ramdisks of 4096K size
April 1 23:23:45 ns kernel: hda: Micropolis 2217A, 1551MB w/508kB Cache,
CHS=3152/16/63
April 1 23:23:45 ns kernel: hdb: Maxtor 72700 AP, 2583MB w/128kB Cache,
CHS=20746/15/17
April 1 23:23:45 ns kernel: ide0 at 0x1f0-0x1f7, 0x3f6 on irq 14
April 1 23:23:45 ns kernel: Floppy drive(s): fd0 is 1.44M
April 1 23:23:45 ns kernel: FDC 0 is an 8272A
April 1 23:23:45 ns kernel: md driver 0.35 MAX_MD_DEV=4, MAX_REAL=8
April 1 23:23:45 ns kernel: scsi : 0 hosts.
April 1 23:23:45 ns kernel: scsi :mdetected total.
April 1 23:23:45 ns kernel: Phân vùng check:
April 1 23:23:45 ns kernel: hda: hda1
April 1 23:23:45 ns kernel: hdb: hdb1 hdb2
April 1 23:23:45 ns kernel: VFS: Mounted root (ext2 filesystem) readonly.
April 1 23:23:45 ns kernel: Adding Swap: 3300k swap-space (priority -1)
April 1 23:23:45 ns kernel: sysctl: ip forwarding off
April 1 23:23:45 ns kernel: Swansea Univerity Computer Society IPX 0.34 for NET
3.035
April 1 23:23:45 ns kernel: IPX Portions Copyright 1995 Caldera, Inc.
April 1 23:23:45 ns kernel: Appletalk 0.17 for Linux NET 3.035
April 1 23:23:45 ns kernel: eth0: 3c509 at 0x300 tag 1, 10baseT port, address 00 60
97 13 30 e1, IRQ 10.
April 1 23:23:45 ns kernel: 3c509.c:1.12 6/4/97 becker@cesdis.gsfc.nasa.gov
April 1 23:23:45 ns kernel: eth0: Setting Rx mode to 1 addresses.

April 1 23:23:45 ns named[2431]: starting. named 4.9.6-REL Thu Nov 6 23:29:57 EST 1997

^Iroot@porky.redhat.com:/usr/src/bs/BUILD/ bind-4.9.6/named

SysV init kích hoạt mọi tiến trình do hệ điều hành yêu cầu, chẳng hạn như cho phép việc thao tác trên mạng, sử dụng con chuột, cùng một số thao tác cơ bản khác như vào ra terminal. SysV init biết phải kích hoạt những tiến trình nào bằng cách đọc các tệp cấu hình trong thư mục /etc/rc.d. Sau này những tệp ấy sẽ được xử lý riêng lẻ tùy theo cấp chạy chương trình (run level) đã xác định. Cấp chạy chương trình quy định việc cung cấp các loại dịch vụ, từ chế độ chạy single-user (một người sử dụng, cấp 1) đến multiuser, multitasking và chế độ chạy tất cả các tiến trình (cấp 3). Bảng sau giới thiệu các cấp chạy của Linux. Giá trị mặc định của cấp chạy được lưu trong tệp /etc/inittab.

Bảng 7.1: Các cấp chạy của Linux

Cấp	Mô tả
0	Ngừng chạy
1	Single-user, không chạy các dịch vụ mạng NFS
2	Multiuser, không chạy các dịch vụ mạng NFS
3	Multiuser, hỗ trợ hoàn toàn các dịch vụ mạng
4	Không sử dụng (đang để dành)
5	Multiuser với các dịch vụ mạng và đăng nhập với giao diện đồ họa
6	Khởi động lại

Chương trình SysV init sử dụng cấu trúc thư mục như sau:

```
init.d
rc0.d
rc1.d
rc2.d
rc3.d
rc4.d
rc5.d
rc6.d
```

Những con số n trong tên thư mục rcn.d tương ứng với cấp chạy n trong bảng 9.1. Mỗi thư mục chứa nhiều shell script có khả năng kích hoạt hoặc dừng hẳn các dịch vụ cần thiết cho mỗi cấp chạy. Các script này cũng kích hoạt hệ thống tệp và khoá tệp vào trạng thái theo ý user. Thật ra các tệp trong các thư mục rcn.d chỉ là các liên kết đến các tệp script trong thư mục init.d. Do đó khi chúng ta muốn sửa chữa gì trong các tệp script khởi động, chúng ta không cần phải vào từng thư mục của chế độ khởi động mà chỉ cần vào thư mục init.d để sửa. (xem thêm ở dưới).

Mỗi thư mục chứa nhiều shell script, tên mỗi tệp script bắt đầu bằng ký tự S hoặc K (Start hoặc Kill: Bắt đầu hoặc Huỷ) theo sau là một con số có hai chữ số. Con số này chỉ để xếp thứ tự trình tự chứ không có ý nghĩa nào khác.

Mặc dù có thể chấp nhận nhiều tham số, song mỗi script thường chấp nhận một đối số dòng lệnh, đối số này dùng để kích hoạt hoặc dừng hẳn, init cung cấp một trong hai thứ, hoặc bắt đầu hoặc dừng hẳn tùy thuộc vào việc liệu rc có được gọi vào để thay đổi cấp chạy hay không.

Chúng ta có thể thực hành các script theo cách thủ công nếu cần phải cấu hình lại một dịch vụ nào đó. Chẳng hạn chúng ta có thể dùng sendmail với lệnh sau (chúng ta phải đăng nhập với tư cách root mới được phép ra lệnh thực thi các script init):

```
/etc/rc.d/init.d/sendmail stop
/etc/rc.d/init.d/sendmail stop
/etc/rc.d/init.d/sendmail start
```

Chúng ta lưu ý hai điều sau đây. Thứ nhất, câu lệnh được lặp lại hai lần với tham số stop là để đảm bảo hệ thống có đủ thời gian ngừng tiến trình trước khi lệnh start được gọi. Thứ hai, chúng ta nhận thấy script được thực thi từ thư mục init.d, mà không từ thư mục của cấp chạy. Ngoài ra script này cũng không có chữ nào hoặc số nào.

Khi liệt kê các tệp trong bất kỳ thư mục cấp chạy nào, chúng ta sẽ thấy rằng chúng liên kết với các tệp trong thư mục init.d.

Sau đây là một thư mục rc.3.d điển hình:

```
lrwxrwxrwx 1 root root 16 Jan 25 21:56 K08autofs->../init.d/autofs
lrwxrwxrwx 1 root root 18 Dec 14 12:17 K10pnserver->../init.d/pnserver
lrwxrwxrwx 1 root root 17 Dec 14 12:17 K20rusersd->../init.d/rusersd
lrwxrwxrwx 1 root root 15 Dec 14 12:17 K20rwhod->../init.d/rwhod
lrwxrwxrwx 1 root root 15 Dec 14 12:17 S15nfsfs->../init.d/nfsfs
lrwxrwxrwx 1 root root 16 Dec 14 12:17 S20random->../init.d/random
lrwxrwxrwx 1 root root 16 Dec 14 12:17 S30syslog->../init.d/syslog
lrwxrwxrwx 1 root root 13 Dec 14 12:17 S40atd->../init.d/atd
lrwxrwxrwx 1 root root 15 Dec 14 12:17 S40crond->../init.d/crond
lrwxrwxrwx 1 root root 14 Dec 14 12:17 S50inet->../init.d/inet
lrwxrwxrwx 1 root root 15 Dec 14 12:17 K10pnserver->../init.d/pnserver
lrwxrwxrwx 1 root root 15 Dec 14 12:17 S55named->../init.d/named
lrwxrwxrwx 1 root root 13 Dec 14 12:17 S60lpd->../init.d/lpd
lrwxrwxrwx 1 root root 13 Jan 31 20:17 S72amd->../init.d/amd
lrwxrwxrwx 1 root root 18 Dec 14 12:17 S75keytable->../init.d/keytable
lrwxrwxrwx 1 root root 18 Dec 14 12:17 S80sendmail->../init.d/sendmail
```

Bảng sau sẽ giới thiệu vài script kích hoạt quan trọng trong thư mục ấy.

Bảng 7.2: Các script của rc.3 init

Tên script	Daemon	Mô tả
------------	--------	-------

S15nfsfs	nfs	Xử lý dịch vụ tệp mạng NFS
S30syslog	syslog	Ghi lại các thông báo hệ thống vào /var/log/messages
S40atd	atd	Cho phép các user thực hiện một công việc nhất định vào thời gian nhất định qua lệnh atd
S40crond	cron	Lập thời biểu batch cho Linux qua lệnh crontab
S50inet	inetd	Super server PID 1
S55named	name server	Cung cấp dịch vụ tên miền DNS
S60lpd	lpd	Nạp daemon máy in để chuẩn bị in

SysV init kiểm tra tất cả các tệp trong thư mục cấp chạy được chỉ định và chuyển một trong hai tham số start hay stop dựa vào ký tự đầu tiên của tên tệp.

Thư mục rc.d còn chứa ba hay bốn tệp gọi là rc, rc.local, rc.sysinit và (hoặc không có) rc.serial. Shell script mang tên rc có nhiệm vụ khởi động lại hệ thống ở một cấp chạy khác, script này sẽ căn cứ vào tham số là con số tương ứng với cấp chạy mới. Sau khi tiến trình khởi động kích hoạt xong tất cả các script khác thì mới đến lượt tệp rc.local. Chúng ta có thể đặt bất kỳ chỉ lệnh kích hoạt tại chỗ nào vào tệp này. Tệp rc.local sau đây giới thiệu một thí dụ về việc kích hoạt một tiến trình tại chỗ, gọi là secure shell, tiến trình này cho phép truy cập từ xa có bảo vệ hệ thống.

```
#!/bin/sh
# This script will be executed *after*
# all the other init scripts
# You can put your own initialization stuff
# in here if you don't
# want to do the full sys v style init stuff.
if [-f/etc/redhat-release]; then
R=$(cat / etc/redhat-release)
else
R=release 3.0.3
if
arch=$(uname-m)
a=a
case _$arch in
_a*) a=an;;
_i*) a=an;;
esac
# This will overwrite /etc/inssue at every boot.
# So, make any changes you want to make to /etc/issue
# here or you will lose them when you reboot.
echo >/etc/issue
echo RedHat Linux $R>> /etc/issue
```

```
echo Kernel $(uname-r) on $a$ (uname-m)>> /etc/issue
cp-f /etc/issue/etc/issue.net
echo>> /etc/issue
##Start sshd
```

```
/usr/local/sbin/sshd
```

rc.sysinit là tệp đầu tiên mà SysV init chạy ngay khi khởi hành. Script này thực thi một số chức năng, chẳng hạn như thiết lập các biến cho cả hệ thống (thí dụ như hostname), kiểm tra hệ thống tệp và bắt đầu sửa chữa, bật mở quota của các user và nạp hệ thống tệp /proc. Script trên cũng kích hoạt một tiến trình tại chỗ mang tên sshd, vốn là một shell daemon an toàn, chuyên hỗ trợ các lệnh từ xa và telnet.

Cấp chạy mặc định được quyết định tại /etc/inittab bằng lệnh:

```
id:3:initdefault:
```

Nhận được lệnh này, khi khởi động hệ thống sẽ chạy ở cấp 3 (chế độ multiuser và multitasking hỗ trợ hoàn toàn các dịch vụ mạng).

Sau đây là một tệp /etc/inittab điển hình:

```
# inittab This file describes how the INIT process
# hold set up the system in a certain run-level
#
# Author: Mique1 van Smoorenburg. <miquels@drinkel.n1.mugnet.org>
# Modified for RHS Linux by Marc Ewing and Donnie Barnes
#
# Default run level. The run levels 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)
id:3:initdefault:
# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit
10:0:wait:/etc/rc.d/rc 0
11:1:wait:/etc/rc.d/rc 1
12:2:wait:/etc/rc.d/rc 2
13:3:wait:/etc/rc.d/rc 3
14:4:wait:/etc/rc.d/rc 4
15:5:wait:/etc/rc.d/rc 5
16:6:wait:/etc/rc.d/rc 6
# Things to run in every run level.
```

```

ud::once:/sbin/update
# Trap CTRL-ALT-DELETE - Allow shutdown system with
# key combination 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 power 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 run levels
1:12345: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 run level 5
x:5:respawn:/usr/bin/x11/xdm-nodaemon

```

Chúng ta đừng vội chạy mặc định ở cấp 0 hoặc cấp 6 nếu không hệ thống sẽ không làm gì cả vì nó sẽ dừng ngay khi khởi động lại (restart) hay mới lại (reboot) liên tục, đôi khi còn làm hỏng tệp. Nếu vì lý do nào đó mà tệp inittab bị hỏng, hay không thể đăng nhập được (quên mật khẩu hay tệp passwd bị hỏng), chúng ta có thể khởi động lại ở chế độ single-user để sửa chữa. Để làm như thế, tại dấu nhắc khởi động LILO, chúng ta gõ tham số “linux single” như sau:

```
LILO boot: linux single
```

LILO là chương trình môi Linux, được bàn kỹ trong giáo trình này.

Còn nếu chúng ta đã cài GRUB, chúng ta cần thực hiện lần lượt các bước sau:

- + Tại màn hình khởi động GRUB, chọn Red Hat Linux và bấm phím <e> để sửa.
- + Chọn dòng kernel và bấm <e>
- + Tại dấu nhắc, gõ “single” và <Enter>
- + Khi đã trở lại màn hình GRUB với thông tin về kernel, bấm để khởi động cấp chạy 1.

3. Đóng tắt Linux

Mục tiêu :

- Nêu được các lưu ý khi đóng tắt linux
- Đóng tắt được hệ thống Linux bằng các lệnh cơ bản

Với Linux, chúng ta nên cẩn thận khi đóng tắt, không được tắt điện ngay. Linux lưu các thông tin I/O của hệ thống tệp trong vùng đệm bộ nhớ, do đó việc chúng ta tắt điện đột ngột có thể sẽ làm hệ thống tệp hỏng hóc.

Cách tốt nhất để đóng tắt Linux là sử dụng lệnh shutdown, với cú pháp như sau:

`/sbin/shutdown [các cờ] thời gian [thông báo]`

“thông báo” là thông báo cho toàn thể các user hiện đang làm việc, và “thời gian” là thời gian mà việc đóng tắt xảy ra. Đối số thời gian có những dạng như sau:

- Dạng thời gian tuyệt đối theo kiểu gg:pp, với gg là số giờ (một hoặc hai chữ số) và pp là số phút của thời gian ấy. Giá trị pp phải có hai chữ số.
- Dạng +m, với m là số phút phải chờ trước khi đóng tắt. Chúng ta có thể thay +0 bằng từ now khi muốn tắt ngay. Thí dụ:

`[root@web root]#/sbin/shutdown -h now`

Cờ	Ý nghĩa
-t sec	Chờ một thời gian đã được xác định bằng số giây giữa thời điểm phát lời cảnh báo và thời điểm đóng tắt tất cả mọi tiến trình. Thời gian trì hoãn này đủ để các chương trình khác kết thúc tiến trình đóng tắt riêng biệt
-k	Không thực sự đóng tắt hệ thống mà chỉ cảnh báo các user khác
-r	Khởi động lại (reboot) sau khi đóng tắt
-h	Dừng hẳn hệ thống (halt) sau khi đóng tắt
-n	Đừng đồng bộ hoá các đĩa trước khi khởi động lại hoặc dừng hẳn. Chúng ta cẩn thận khi dùng flag này, vì có nguy cơ làm hỏng dữ liệu
-f	Khởi động “nhanh”. Flag này sẽ tạo ra tệp /etc/fastboot. Boot script có tên rc sẽ tìm tệp này và nếu có, sẽ không thực hiện fsck
-c	Hủy một tiến trình đóng tắt đang thi hành. Flag này vô hiệu hoá đối số thời gian

Bảng 7.3: Các cờ (flag) của lệnh shutdown

Lệnh shutdown thông báo các user biết hệ thống sắp đóng tắt, đóng cổng đăng nhập hệ thống, sau đó gửi một tín hiệu SIGTERM đến tất cả các tiến trình để thực hiện việc tắt đúng cách. Tiếp theo shutdown dựa vào sự chọn lựa của chúng ta ở dòng lệnh shutdown để khởi động lại hoặc dừng hẳn.

Lưu ý: Chúng ta muốn dừng hẳn hoặc khởi động lại bằng cách gõ trực tiếp lệnh halt hoặc reboot cũng được. Tuy nhiên nếu làm như vậy thì hệ thống sẽ đóng tắt tức khắc mà các user khác không thể biết trước. Chúng ta chỉ nên dùng

hai lệnh vừa kể khi chỉ có một mình sử dụng hệ thống. Muốn biết trên mạng còn user nào không, chúng ta bấm <w> hoặc dùng lệnh who.

B. CÂU HỎI VÀ BÀI TẬP

Câu 1: Trình bày trình quản lý môi LILO?

Câu 2: Trình bày trình quản lý môi GRUB?

Câu 3: Nêu các tiến trình khởi động?

BÀI 8: QUẢN LÝ TÀI KHOẢN

Mã bài: MD37-08

❖ Giới thiệu

Muốn làm quản trị viên Linux, chúng ta phải tìm hiểu các công cụ và kỹ thuật khác nhau để quản lý tài khoản của các user. Tác vụ này bao gồm việc đăng ký các user mới để họ có thể đăng nhập hệ thống; thiết lập các quyền ưu tiên cho họ; tạo ra các thư mục “nhà” (home directory) và gán chúng cho họ; sắp xếp user nào vào nhóm nào và ngược lại sẽ xoá tên của một vài user trong hệ thống khi cần thiết.

❖ Mục tiêu

- Làm việc được với các user.
- Làm việc được với nhóm user.
- Quản lý được home directory.
- Quản trị được các user qua giao diện web.
- Rèn luyện khả năng tổ chức, quản trị.

❖ Nội dung chính

A. LÝ THUYẾT

1. Làm việc với các user và nhóm user

Mục tiêu :

- Thêm được một user vào hệ thống
- Sử dụng được lệnh *adduser*
- Thiết lập được mật khẩu cho user
- Thêm và xoá được một nhóm của hệ thống

1.1. Làm việc với các user

Chúng ta cần cho mỗi user một tên đăng nhập đặc trưng và riêng biệt. Tên này giúp chúng ta nhận dạng từng user và tránh được tình trạng người này xoá dữ liệu của người kia.

Mỗi user phải có mật khẩu riêng. User chỉ không cần mật khẩu khi có một mình trong hệ thống và hoàn toàn không kết nối với ai khác qua modem hoặc qua mạng.

Khi có lý do chính đáng để bỏ ai đó ra ngoài hệ thống, quản trị viên phải xoá tên đăng nhập của người ấy cùng với các tệp mà các user khác không còn nhu cầu sử dụng nữa.

1.1.1. Thêm vào một user

Khi thêm một user vào hệ thống, tệp mật khẩu có thêm một mục với dạng như sau:

```
logname:enrpt_passwd:userID:groupID:userInfor:login_directory:login_shell
```

Trong cú pháp trên, các trường được phân cách bằng dấu hai chấm.

Bảng 8.1: Các trường trong một mục ghi của tệp /etc/passwd

Trường	Mô tả
Logname	Tên dùng khi đăng nhập
enrpt_passwd	Mật khẩu để nhận dạng user; đây là biện pháp đầu tiên để ngăn chặn việc vi phạm an ninh. Trường này thường được mã hoá
userID	Số định danh mà hệ thống dùng để phân biệt từng user
groupID	Số định danh hoặc tên đặc trưng dùng để nhận dạng nhóm sơ cấp gán cho user. Nếu thuộc về nhiều nhóm, một user có thể chuyển qua lại các nhóm khác khi được quản trị viên cho phép
userInfo	Thông tin về user, chẳng hạn như tên họ hoặc chức vụ
login_directory	Home directory của user, nơi user có mặt khi đăng nhập
login_shell	Shell được user dùng sau khi đăng nhập (thí dụ /bin/bash nếu user dùng shell bash)

Thí dụ:

```
[root@web etc]# cat /etc/passwd
```

```
nva:45b9d200f5f4683e569dcaa3ed6f6fb:500:231:Nva:/home/nva:/bin/bash
```

```
pla:aeswb00feø90e9cd47a01d86514a976:511:787:Pla:/home/pla:/bin/bash
```

```
bhh:4d844e2e58e14a0b377b276065c86036:515:787:Bhh:/home/bhh:/bin/bash
```

```
root:8aaa77148543346ea015cf18c57b0ce4d:0:0:root:/root:/bin/bash
```

Lệnh **adduser** hoặc **useradd** sẽ thêm user vào hệ thống. Chúng ta gõ lệnh này với tên của user mà chúng ta muốn thêm vào. Mục sau sẽ nói chi tiết hơn về chủ đề này.

Trong thực tế kể từ phiên bản 6.x, RedHat Linux đã sử dụng công cụ shadow để bảo mật hơn nữa. Khi đó tệp passwd có dạng rút gọn, thí dụ như sau:

```
gdm:x:42:42::/home/gdm:/bin/bash
squid:x:23:23::/var/spool/squid:dev/null
nva:x:500:231:Ngo Van An:/home/nva:/bin/bash
pla:x:511:787:Phan Lan Anh (PI):/home/pla:/bin/bash
bhh:x:515:787: Bui Huy Hung (PI):/home/bhh:/bin/bash
```

Trong đó x đóng vai trò đại diện cho mật khẩu thật đang được chứa trong tệp /etc/shadow, thí dụ:

```
nva:$1$32rbaeg0d$4UtJgG/7rMqQJypA7pf0p0:10857:0:9999:7:-1:-1:134537356
pla: $1$3j86/RIF$qVUOxa.ZuPXnGLWuAC6i/:11360:0:9999:7:-1:-1:134549706
bhh:$1$EdjW7kPY$Hyn/xTNttk8fsWsZyvQo.0:10857:0:9999:7:-1:-1:134538444
```

1.1.2. Sử dụng lệnh adduser

Trong các phiên bản cũ (RedHat Linux 5.x), khi thêm vào một user, chúng ta chỉ việc gõ lệnh **adduser** kèm với tên của user ấy, thí dụ:

```
[root@digital alberto] # adduser lananh
Looking for first available UID...502
Looking for first available GID...502
Adding login:lananh...done.
Creating home directory: /home/lananh...done.
Creating mailbox: /var/spool/mail/lananh...done.
Don't forget to set the password.
```

Lệnh **adduser** chép các tệp có tên bắt đầu bằng dấu chấm (.) từ thư mục /etc/skel sang home directory của user. Thư mục /etc/skel phải có các tệp khuôn mẫu (template) để cho tất cả các user dùng chung. Những template ấy phải có các tệp cấu hình “riêng” như .profile, .cshrc và .login để cấu hình shell; .mailrc để thiết lập thư điện tử; .emacs để cho các user có thể sử dụng **emacs** làm trình soạn thảo v.v.

Lệnh **adduser** là một shell script nằm ở thư mục /usr/sbin, do đó chúng ta có thể tùy nghi chỉnh sửa script này chạy theo ý mình khi cần thực hiện thêm vài động tác nữa lúc thêm trương khoản mới vào. Loại chỉnh sửa thường gặp nhất là bắt **adduser** cung cấp nguyên tên user thay vì ép một tên user mặc định vào tệp mật khẩu. Nếu không thích thay đổi trong script để script hỏi tên user, chúng ta phải thay đổi thủ công bằng lệnh chfn như sau:

```
# chfn bhh
changing finger information for bhh.
Name [RH Linux User]: tvl
Office []:
Office Phone []:
Home Phone []:
Finger information changed.
```

Chúng ta không thể dùng lệnh **adduser** để thiết lập mật khẩu cho trường khoản mà phải dùng lệnh **passwd**.

Trong các phiên bản từ 6.x trở đi, RedHat Linux đã chuyển script này thành một lệnh với các tham số như sau:

```
adduser [-u uid [-o]] [-g group] [-G group...]
[-d home] [-s shell] [-c comment] [-m [-k template]]
[-f inactive] [-e expire] [-p passwd] [-n] [-r] name
adduser -D [-g group] [-b base] [-s shell]
[-f inactive] [-e expire]
```

Trong đó:

Trường	Ký hiệu (ghi chú)
logname	Name (tên đăng nhập)
userID	uid (nên để tự động)
groupID	group (có thể dùng tên của nhóm hay dùng gid)
userInfo	comment (cần đóng ngoặc đơn hoặc kép cho thông tin sẽ điền)
login_directory	Home (thư mục “nhà” mặc định dùng khi đăng nhập)
login_shell	Shell (shell mặc định dùng khi đăng nhập)

Thí dụ:

```
[root@mail /boot]# adduser -u 401 -g mail -c “thu ky” office
```

```
[root@mail /boot]# adduser -g mail -c ‘trien khai’ develop
```

1.1.3. Thiết lập mật khẩu cho user

Chúng ta dùng lệnh **passwd** để thiết lập mật khẩu ban đầu cho user. Sau đó từng user sẽ tự thay mật khẩu theo ý mình khi đã đăng nhập vào hệ thống. Sau đây là các bước căn bản để sử dụng lệnh **passwd**.

+ Gõ lệnh và tên đăng nhập (thí dụ như **passwd pla**) rồi bấm <Enter>

+ Tại dấu nhắc **New password:**, chúng ta gõ mật khẩu vào.

+ Khi máy nhắc phải gõ mật khẩu lần nữa, chúng ta chỉ việc gõ lại mật khẩu mới như sau:

New password (again): mật_khẩu_mới

Rồi mật khẩu đó được mã hoá và cất vào tệp **/etc/passwd**.

Mật khẩu phải đáp ứng hai điều kiện: dài ít nhất sáu ký tự (tám ký tự thì an toàn hơn), có cả chữ thường và chữ hoa cùng với các dấu phân cách và chữ số.

Khi đăng ký thêm nhiều user, chúng ta thường thích cho họ mật khẩu ngắn gọn và dễ nhớ. Việc quy định số ký tự tối thiểu là ở chúng ta, theo mặc định RedHat Linux 7.x yêu cầu chúng ta phải nhập ít nhất 6 ký tự. Chúng ta có thể quy định lại thông qua trình **linuxconf** nhưng không nên giảm bớt số ký tự này và không nên chọn mật khẩu dễ nhớ cho người dùng, bởi vì một mật khẩu tốt là tuyến phòng thủ đầu tiên chống lại tin tặc. Chúng ta cần giải thích điều này cho các user biết và nên đều đặn thay đổi mật khẩu.

- Có những nơi mật khẩu được yêu cầu thay đổi hàng tuần. Tuy nhiên điều đó dễ đưa đến tình trạng quên và nhầm lẫn mật khẩu. Chúng ta cần nhắc nhở việc ghi mật khẩu ra giấy cũng sơ hở như việc chọn mật khẩu để nhớ cho người dùng.

- Để tránh tình trạng quên mật khẩu, có thể cho ghi lại mật khẩu đó, bỏ vào phong bì dán niêm phong và cất trong tủ sắt.

Mỗi khi một user được gán mật khẩu, mục ghi của tệp sẽ có dạng thí dụ như sau:

```
pla: Anh.89&^0gW:123:21:Phan Lan Anh:/users/pla:/bin/bash
```

Trường thứ hai là mật khẩu ở dạng mã hoá, không phải là những ký tự lung tung.

Ghi chú: Thỉnh thoảng các user lại quên mật khẩu của mình. Chúng ta không thể nhắc cho từng user nhớ mật khẩu riêng. Tuy nhiên chúng ta có thể xoá mật khẩu bị quên bằng cách dùng lệnh `passwd` thiết lập một mật khẩu mới và thông báo nó cho user biết để tự đặt lại mật khẩu. Với tư cách quản trị viên, chúng ta nên thiết lập quy trình xử lý trường hợp vừa kể và cho các user biết quy trình ấy.

1.1.4. Gỡ bỏ một user

Chúng ta có thể gỡ bỏ một user theo nhiều cấp độ khác nhau. Việc gỡ bỏ một user ra khỏi hệ thống không phải là động tác “một đi không trở lại”. Sau đây là một vài tuỳ chọn:

Chỉ gỡ khả năng đăng nhập. Điều này có ích khi một user nào đó phải đi xa một thời gian và sau đó sẽ trở lại vào hệ thống. Thư mục, tệp và thông tin về nhóm của user đó

được giữ nguyên. Quản trị viên chỉ phải chỉnh sửa tệp mật khẩu và gỡ dấu sao (*) vào trường thứ hai của mục ghi user như sau:

```
pla:*:123:21:Phan Lan Anh:/users/pla:/bin/bash
```

Cách này không còn áp dụng trên các phiên bản mới với công cụ `shadow`. Gỡ bỏ user khỏi tệp mật khẩu nhưng vẫn để dữ liệu của user trên hệ thống. Hình thức này có ích khi các user khác muốn sử dụng những tệp dữ liệu ấy, hoặc có ai đó nhận nhiệm vụ thay cho user cũ. Quản trị viên xoá mục ghi của user cũ khỏi tệp mật khẩu bằng trình soạn thảo hoặc bằng lệnh:

```
userdel tên_đăng_nhập.
```

Sau đó thay đổi quyền sở hữu và vị trí các tệp của user đó bằng lệnh `chown` và `mv`.

Gỡ bỏ user ra khỏi tệp mật khẩu và gỡ bỏ tất cả tệp thuộc sở hữu của user ấy. Đây là hình thức cao nhất và đầy đủ nhất để xoá bỏ một user. Quản trị viên xoá mục ghi của user ở tệp mật khẩu và huỷ luôn tất cả các tệp của user ấy trong toàn hệ thống theo lệnh **find** như sau:

```
find user_home_directory exec rm{ };
```

Ta cũng có thể dùng lệnh `userdel` với tham số `-r` như sau:

```
userdel -r tên_đăng_nhập
```

Ghi chú: Nếu hệ thống của chúng ta sử dụng các tệp cấu hình khác, chẳng hạn như tệp bí danh e-mail, thì vì an toàn chúng ta cũng phải xoá tên user ở những tệp ấy.

1.2. Làm việc với nhóm

Mỗi user là thành viên của một nhóm. Tuỳ theo tính chất của mỗi nhóm, quản trị viên sẽ chỉ định nhóm ấy có khả năng gì hoặc ưu tiên nào. Thí dụ có nhóm chuyên về việc phân tích doanh số của công ty, quản trị viên sẽ cho quyền nhóm này truy cập phạm vi tệp rộng lớn hơn so với một nhóm khác chỉ chuyên về tìm tòi sản phẩm mới.

Tệp mật khẩu chứa thông tin của một user. Trong khi đó thông tin của cả nhóm được chứa tại tệp /etc/group. Sau đây là thí dụ một mục ghi:
office::21:tv1, pla, nva

Ở thí dụ này, tên nhóm là office, số định danh nhóm là 21 và các thành viên của nhóm là tv1, pla và nva. Các thư mục và tệp có thuộc tính quyền hạn gắn liền với chủ sở hữu, nhóm và các yếu tố khác. Một user có thể là thành viên của nhiều nhóm khác nhau và quản trị viên có thể thay đổi việc tham gia các nhóm của user.

1.2.1. Thêm vào một nhóm

Chúng ta có thể tạo ra một nhóm mới bằng cách chỉnh sửa trực tiếp trong tệp /etc/group và đưa thông tin của nhóm ấy vào làm một mục ghi.

Trong tệp /etc/group, mỗi nhóm đều có số định danh riêng. Linux chỉ quan tâm đến con số này chứ không quan tâm đến tên nhóm. Do đó nếu chúng ta cấp cho hai nhóm cùng một con số định danh, Linux sẽ coi đây chỉ là một nhóm.

1.2.2. Xoá bỏ một nhóm

Muốn xoá bỏ một nhóm, chúng ta xoá đi mục ghi tương ứng trong tệp /etc/group. Tiếp theo chúng ta phải chuyển tất cả các tệp với GID tương ứng sang cho một nhóm khác. Chúng ta dùng lệnh **find** như sau:

```
find /-gid nhóm_A find user_home_directory -exec chgrp nhóm_mới {} \;
```

Chúng ta cũng có thể dùng các lệnh về nhóm như: groupdel, groupmod và groupadd để xử lý các thao tác trên.

2. Quản lý home directory

Mục tiêu :

- Trình bày được cách quản lý home directory

Nếu chúng ta dự kiến hệ thống của mình sẽ có nhiều user, hãy suy nghĩ trước về cách sắp xếp các home directory cho hợp lý. Cần cố gắng đặt tất cả các home directory của cùng một hệ thống vào một thư mục cấp cao nhất. Như thế, nếu sau này chúng ta có sắp xếp chúng lại như thế nào thì chúng cũng nằm chung với nhau.

Thí dụ chúng ta có thể chỉ định rằng thư mục /home là thư mục cấp cao nhất dành cho các thư mục của user. Dưới thư mục /home này, chúng ta tập hợp các user theo từng khối. Các trương khoản cho user của khối office (văn phòng) sẽ nằm ở /home/office, các trương khoản cho user của khối develop (triển khai)

sẽ nằm ở /home/develop, v.v. Từ đó các home directory của những user trên hệ thống máy chúng ta sẽ nằm trong những thư mục ấy, hoặc chúng ta sẽ tạo ra thêm thư mục mới khi có yêu cầu. Vì các thư mục cho phép user sẽ chiếm nhiều chỗ trên ổ cứng, chúng ta phải suy nghĩ cách xếp đặt các nhóm logic trên nhiều hệ thống tệp vật lý khác nhau. Khi cần thêm khoảng trống trên ổ cứng, chúng ta chỉ việc tạo ra một hạng mục mới cho các home directory mới và lắp hạng mục ấy vào một hệ thống tệp như là một điểm ghép (**mount point**) trong thư mục /home.

3. Quản trị qua giao diện web

Mục tiêu :

- *Quản trị được các user và nhóm qua giao diện web*

Từ bản phát hành Red Hat 5.1 bắt đầu có một công cụ quản trị hệ thống mang tên Linuxconf. Công cụ này giúp chúng ta xử lý nhiều tác vụ quản trị, chẳng hạn như làm việc với user và với nhóm. Ngoài hai giao diện thường gặp là dòng lệnh và X Windows, Linuxconf còn hỗ trợ tác vụ quản trị qua giao diện web.

Chúng ta cũng có thể dùng một trình duyệt sẵn có trong giao diện đồ họa GNOME, gọi là Nautilus. Khi khởi động GNOME, công cụ Nautilus hoạt động ngay và mở cửa sổ “START HERE” trong đó chúng ta có thể thấy các biểu tượng “System settings”, “Server Configuration” (tương tự trong “Control Panel” của Windows) để quản lý các cấu hình của hệ thống.

B. CÂU HỎI VÀ BÀI TẬP

Câu 1: Nêu cách thêm user và thiết lập password cho user?

Câu 2: Nêu cách thêm một nhóm và xóa bỏ một nhóm?

Câu 3: Trình bày cách quản lý home directory và quản trị qua giao diện web?

Câu 4: Thực hành quản lý tài khoản của hệ thống?

Hướng dẫn thực hành:

1. Tài khoản người dùng:

Mỗi người sử dụng trên hệ thống được mô tả qua các thông tin sau:

- username : tên người sử dụng
- password : mật khẩu (nếu có)
- uid : số nhận dạng (user identify number)
- gid : số của nhóm (group identify number)
- comment : chú thích
- Thư mục chủ của tài khoản (home directory)
- Shell đăng nhập (chương trình chạy lúc bắt đầu phiên làm việc) Các thông tin trên được chứa trong tệp tin /etc/passwd

2. Tài khoản nhóm người dùng:

Một nhóm người sử dụng được mô tả bằng các thông tin sau:

- groupname: tên của nhóm
- gid : số của nhóm (gid: group identify number)
- danh sách các tài khoản thuộc nhóm

Các thông tin trên được chứa trong tập tin /etc/group

3. Thực hành tạo tài khoản hệ thống

Tạo nhóm cntt2004

```
#groupadd cntt2004
```

Xem tập tin /etc/group

```
#cat /etc/group
```

Tạo một account user01 mới thuộc nhóm cntt2004

```
#useradd -g cntt2004 -c "Tai khoan user01" user01
```

```
#passwd user01
```

Xem tập tin /etc/passwd, /etc/shadow

```
#cat /etc/passwd
```

```
#cat /etc/shadow
```

Thử đăng nhập vào hệ thống với tài khoản là user01

Tạo một account user02

```
#useradd user02
```

```
#passwd user02
```

Đưa user02 vào nhóm cntt2004

```
#usermod -g cntt2004 user02
```

Thử đăng nhập vào hệ thống với tài khoản là user02

Xóa user02

```
#userdel user02
```

```
#cat /etc/passwd
```

Câu 5: Thực hành phân quyền người dùng trên hệ thống tập tin?

Hướng dẫn thực hành:

1. Các quyền truy xuất trên tập tin

Khi tập tin được tạo lập, các thông tin sau đây đồng thời được ghi lại:

- uid của người tạo tập tin
- gid của người tạo tập tin
- Các quyền thâm nhập tập tin khác . . .
- Tập tin được bảo vệ bởi một tập hợp các bit định nghĩa quyền thâm nhập

r w x	r w x	r w x
suid	soid	
owner	group	other

Trong đó:

r Quyền đọc nội dung tập tin, thư mục

w Quyền tạo và xoá nội dung tập tin, tạo và xoá tập tin trong thư mục x
Quyền thực thi tập tin. Quyền truy xuất qua lại trên thư mục.

- Các quyền với thư mục chỉ có hiệu lực ở một mức nhất định, thư mục con có thể được bảo vệ trong khi thư mục cha thì không.

- Lệnh `ls -lF` liệt kê danh sách các tập tin và các thuộc tính của chúng trong một danh mục, qua đó ta có thể xem các thông tin như loại tập tin, quyền truy nhập, người sở hữu và kích thước của tập tin. . .

2. Lệnh `chmod`

Lệnh `chmod` cho phép thay đổi quyền trên tập tin của người dùng. Chỉ những người sở hữu tập tin này mới có thể thay đổi được mức đặc quyền đối với tập tin này. Có thể thực hiện lệnh theo hai cách:

+ Dùng các ký hiệu tượng trưng:

Cú pháp : `chmod {a,u,g,o}{+,-,=}{r,w,x} <filename>`

Trong đó : u (user), g (group), o (other), a (all)

Các toán tử : + thêm quyền. - bớt quyền. = gán giá trị khác

+ Dùng thông số tuyệt đối:

Cú pháp : `chmod <mode> <filename>`

trong đó mode là một số cơ số 8 (octal)

r w x r - x r - - 1 1 1 1 0 1 1 0 0

7 5 4

`$chmod 754 filename`

`$chmod g-w,o+r baitho.doc`

`$chmod a+r baocao.txt`

`$chmod +r baocao.txt`

`$chmod og-x baocao.txt` không cho thực thi

`$chmod u+rwx baocao.txt` cho phép người sở hữu có thể đọc, viết và thực thi.

`$chmod o-rwx baocao.txt` không cho truy nhập tập tin.

`$chmod 777 *` Đặt các quyền cho tất cả các đối tượng sử dụng .
trên toàn bộ tập tin trong thư mục hiện hành

3. Thay đổi người hoặc nhóm sở hữu tập tin

- Lệnh `chown` cho phép thay đổi người sở hữu, nhóm sở hữu trên tập tin.

- Lệnh `chgrp` cho phép thay đổi nhóm sở hữu trên tập tin.

4. Thực hành thay đổi quyền sử dụng cho các đối tượng trên tập tin

a. Tạo một tập tin mới `/home/baocao.txt`

b. Đổi chủ sở hữu của tập tin `/home/baocao.txt` là `user01`

`#chown user01 /home/baocao.txt`

c. Phân quyền `rwxr--r--` cho các đối tượng trên tập tin `/home/baocao.txt`.

`#chmod 744 /home/baocao.txt`

d. Đăng nhập vào hệ thống với tài khoản `user01`. Thử thay đổi nội dung tập tin `/home/baocao.txt`.

e. Đăng nhập vào hệ thống với tài khoản khác. Thử thay đổi nội dung tập tin `/home/baocao.txt`. Nhận xét ?

5. Thực hành phân quyền sử dụng cho các đối tượng

a. Tạo nhóm người sử dụng có tên `cntt2004`.

b. Bổ sung các user01, user02 vào nhóm cntt2004.

```
#usermod -g cntt2004 user01
```

```
#usermod -g cntt2004 user02
```

c. Tạo thư mục /home/common

```
#mkdir /home/common
```

d. Đổi nhóm sở hữu của thư mục /home/common là nhóm cntt2004.

```
#chown :cntt2004 /home/common
```

hoặc

```
#chgrp cntt2004 /home/common
```

e. Phân quyền rwx cho đối tượng nhóm cntt2004 trên thư mục /home/common

```
#chmod g+rwx /home/common
```

```
#ls -lF /home
```

f. Đăng nhập vào hệ thống với tài khoản user01. Tạo thư mục mới trong /home/common.

g. Đăng nhập vào hệ thống với một tài khoản khác không thuộc nhóm cntt2004.

Thử tạo thư mục mới trong /home/common. Nhận xét ?.

BÀI 9: SAO LƯU DỮ LIỆU

Mã bài: MD37-09

❖ Giới thiệu

Có nhiều nguyên nhân làm hỏng hóc hoặc mất mát dữ liệu như tệp chẳng may bị xoá, trục trặc phần cứng, thông tin liên quan nằm ở những tệp không còn truy cập được. Một quản trị viên giỏi phải đảm bảo sao cho các user vẫn tham khảo được những tệp “đã mất” như thế. Muốn vậy, chúng ta phải sao lưu dữ liệu kịp thời.

Tương lai của cơ quan chúng ta - và tương lai của chúng ta tại cơ quan - có thể tùy thuộc vào việc các user truy cập được những tệp đã sao lưu ấy. Tại những thời điểm nóng bỏng như thế, bản thân chúng ta cũng như các người khác sẽ có cơ hội nhận thức giá trị của thời gian và công sức để sao chép dữ liệu một cách đều đặn, chặt chẽ và theo một thời điểm đầy đủ. Việc sao lưu tệp không mấy hấp dẫn, nhưng quản trị viên không thể không am tường tiến trình sao lưu.

❖ Mục tiêu

- Nêu được vấn đề về sao lưu.
- Trình bày được các thủ thuật sao lưu.
- Hoạch định được thời biểu sao lưu.
- Thực hiện được công việc sao lưu và phục hồi tệp.

❖ Nội dung chính

A. LÝ THUYẾT

1. Vấn đề về sao lưu

Mục tiêu :

- *Nêu được một số câu hỏi lưu ý khi sao lưu*
 - *Biết được công việc của quản trị viên cần phải sao lưu như thế nào*
- Khi sao lưu một hệ thống, chúng ta lưu ý một số câu hỏi như sau:
- Sao lưu toàn bộ hay tăng dần? Sao lưu toàn bộ dĩ nhiên rất an tâm nhưng chúng ta sẽ tốn nhiều thời gian và dung lượng ổ cứng để lưu tất cả các tệp. Trong khi đó việc sao lưu tăng dần chỉ sao lưu những tệp nào đã có thay đổi kể từ lần sao lưu gần nhất mà thôi.
 - Cần sao lưu những hệ thống tệp nào? Những hệ thống tệp nào hiện hành thì cần được sao lưu đều đặn, những gì còn lại không nhất thiết phải làm thường xuyên. Là quản trị viên, chúng ta phải đảm bảo luôn sẵn có bản sao lưu của tất cả các hệ thống tệp.
 - Lưu tệp trên những phương tiện nào? Tùy thuộc vào những thiết bị sẵn có của hệ thống, chúng ta có thể sao lưu dữ liệu trên các loại đĩa cứng, mềm, CD, quang-từ (MOD), DVD, hoặc các loại băng từ v.v. Mỗi loại đều khác nhau về giá cả, tốc độ và dung lượng, do đó chúng ta nên chọn loại nào cho vừa túi tiền, song không nên quên rằng loại rẻ nhất thường lại làm cho chúng ta tốn nhiều thì giờ nhất.

- Việc sao lưu sẽ ảnh hưởng ra sao đối với các user? Sao lưu luôn làm hệ thống công kênh thêm. Liệu đây có phải là gánh nặng không chính đáng cho các user?

Hơn nữa nếu có tệp nào thay đổi trong khi đang sao lưu thì bản thân tệp ấy sẽ không được sao lưu. Điều này thật phiền phức, nhất là khi chúng ta đang sao lưu một cơ sở dữ liệu quan trọng. Vậy chỉ nên sao lưu khi không còn ai khác sử dụng hệ thống?

- Liệu một vài câu lệnh tương đối đơn giản và được nhiều người sử dụng như **tar** và **cpio** có đáp ứng đủ yêu cầu của chúng ta?

- Làm sao có thông tin về các tệp đã sao lưu? Chúng ta phải ghi số hoặc tệp nhật ký và dán nhãn trên mỗi vật mang dữ liệu sao lưu để dễ tìm khi cần dùng đến. Một vài thủ tục và câu lệnh sẽ giúp chúng ta chuẩn bị mục lục hoặc danh sách những gì được sao lưu.

Quản trị viên cần sao lưu theo một quy trình tự động hoặc càng ít can thiệp thủ công càng tốt. Hơn nữa nên sao lưu khi nào hệ thống không còn ai khác sử dụng để bảo đảm an toàn.

Hai điều trên phải được cân bằng với sự tiện lợi và chi phí. Liệu một quản trị viên có phải thức cho đến tận nửa đêm cuối tuần để thực hiện sao lưu toàn bộ?

Liệu có nên bỏ ra nhiều tiền mua băng từ DAT nhằm tự động hoá việc sao lưu toàn bộ hệ thống vào lúc ba giờ sáng mà không cần can thiệp thủ công? Chúng ta phải quan tâm những vấn đề như thế và nhớ rằng thông tin sao lưu được quản lý tốt sẽ giúp chúng ta tiết kiệm được tiền bạc và công sức.

2. Các thủ thuật sao lưu

Mục tiêu :

- *Trình bày được các thủ thuật sao lưu hệ thống*

Mục đích của sao lưu là để khi cần sẽ phục hồi được các tệp cá nhân hoặc tệp hệ thống thật dễ dàng và nhanh chóng. Cho dù chúng ta làm điều gì khi sao lưu thì cũng phải ghi nhớ mục đích ấy.

Chúng ta nên lập một kế hoạch sao lưu. Ghi ra những tệp cần sao lưu, thời biểu xử lý chúng và sau này sẽ phục hồi chúng như thế nào. Thông báo cho các user biết thời biểu cùng với yêu cầu phục hồi tệp. Kế hoạch đã lập ra như thế nào, chúng ta cứ làm theo như thế.

Sao lưu xong phải kiểm tra. Chúng ta thử đọc bằng mục lục trên vật mang dữ liệu sao lưu, hoặc thử phục hồi một tệp nào đó. Nên nhớ rằng bản thân vật mang dữ liệu sao lưu – tức băng hoặc đĩa – đều có thể bị lỗi.

Chúng ta phải sao lưu như thế nào để có thể phục hồi tệp ở một máy khác hoặc ở một hệ thống tệp khác trên cùng một máy. Chúng ta nên sử dụng các tiện ích sao lưu và tiện ích tồn trữ (archive) có khả năng tạo ra tệp tồn trữ mà các hệ Linux hoặc hệ khác đều đọc được.

Bảo đảm tìm thấy các tệp cần thiết bằng cách dán nhãn trên tất cả các vật mang dữ liệu sao lưu. Nếu có nhiều băng hoặc đĩa, phải ghi số thứ tự và ngày tháng sao lưu.

Hãy tránh các tai hoạ nhưng vẫn dự kiến việc chúng sẽ xảy ra và sao lưu các tệp như thế nào để chúng ta có thể phục hồi toàn bộ hệ thống với khoảng thời gian hợp lý. Nên cất các bản sao lưu ở vài nơi xa để phòng trường hợp hoả hoạn thiêu trụi hoàn toàn máy. Nhiều tổ chức đã thuê nơi an toàn để chứa các băng đĩa quan trọng.

Chúng ta cất băng đĩa chỗ nào thì cũng nên cất kèm danh sách toàn bộ cấu hình phần cứng hệ thống.

Thỉnh thoảng chúng ta nên đánh giá lại các thủ tục sao lưu, xem chúng có thực sự thoả mãn yêu cầu hiện hành hay không.

Nhiều công cụ có thể giúp chúng ta tự động hoá tiến trình sao lưu. Chúng ta thử xem qua các tệp Linux tại sunsite.unc.edu để có thêm thông tin. Ngoài ra Linux cũng chấp nhận phần mở rộng FTAPE. FTAPE giúp chúng ta sao lưu dữ liệu lên băng từ QIC-80 thông qua giao diện đĩa mềm. Xem trên Internet tệp HOWTO về FTAPE để biết thêm chi tiết.

3. Hoạch định thời biểu sao lưu

Mục tiêu :

- *Nêu được ý nghĩa của việc hoạch định thời biểu sao lưu*
- *Hoạch định được thời biểu của việc sao lưu dữ liệu*

Như đã nói ở trên, việc hoạch định thời biểu sao lưu là rất quan trọng, cần phù hợp với yêu cầu của hệ thống và quản trị viên cần tuân theo kế hoạch ấy.

Điều lý tưởng là khi có yêu cầu phục hồi tệp thì phải thực hiện được ngay. Thật ra cũng không nhất thiết cần cầu toàn như thế, song trên nguyên tắc quản trị viên phải nắm khả năng phục hồi tệp mỗi ngày. Để đạt mục đích này, chúng ta chọn giải pháp chung giữa hình thức sao lưu toàn bộ và sao lưu tăng dần. Như chúng ta đã biết, một bản sao lưu toàn bộ chứa tất cả các tệp của hệ thống, trong khi sao lưu tăng dần chỉ chứa những tệp nào đã bị thay đổi kể từ lúc sao lưu gần nhất.

Việc sao lưu tăng dần cũng có cấp độ khác nhau: tăng dần kể từ lần sao lưu toàn bộ gần nhất, hoặc tăng dần kể từ lần sao lưu tăng dần gần nhất. Sau đây chúng ta thử đặt cấp độ cho các hình thức sao lưu vừa kể:

- Cấp 0: sao lưu toàn bộ
- Cấp 1: sao lưu tăng dần so với lần sao lưu toàn bộ gần nhất
- Cấp 2: sao lưu tăng dần so với lần sao lưu tăng dần gần nhất.

Sau đây là 2 thí dụ về thời biểu sao lưu:

A. Một ngày sao lưu toàn bộ, các ngày khác sao lưu tăng dần.

Ngày 1: Cấp 0, sao lưu toàn bộ

Ngày 2: Cấp 1, sao lưu tăng dần

Ngày 3: Cấp 1, sao lưu tăng dần

Ngày 4: Cấp 1, sao lưu tăng dần

Ngày 5: Cấp 1, sao lưu tăng dần

Nếu chúng ta tạo ra và lưu trữ chỉ mục của mỗi lần sao lưu, sau này chúng ta chỉ cần bản sao lưu của một ngày để phục hồi tệp cá nhân và bản sao lưu của hai ngày (bản của ngày 1 và bản của một ngày khác) để phục hồi toàn bộ hệ thống.

B. Sao lưu toàn bộ mỗi tháng, sao lưu tăng dần mỗi tuần và tăng dần mỗi ngày.

+ *Thứ ba tuần đầu tiên: Cấp 0, sao lưu toàn bộ.*

+ *Những ngày Thứ ba sau: Cấp 1, sao lưu tăng dần.*

+ *Mỗi ngày khác: Cấp 2, sao lưu tăng dần.*

Theo cách này, nếu cần phục hồi một tệp chưa bị thay đổi gì suốt tháng qua thì chúng ta phải dùng đến bản sao lưu toàn bộ. Chúng ta phải dùng đến bản sao lưu Cấp 1 nếu nội dung tệp đã thay đổi ở tuần trước nhưng tuần này chưa thay đổi gì. Chúng ta sẽ phải dùng bản sao lưu Cấp 2 nếu tệp vừa kể đã bị thay đổi ngay trong tuần này.

Thời biểu này có vẻ phức tạp hơn thí dụ trước, song tác vụ sao lưu hàng ngày sẽ chiếm ít thời gian hơn.

Ngoài ra chúng ta nên lưu trữ các tệp sao lưu suốt một khoảng thời gian nào đó, phòng trường hợp cần phục hồi một phiên bản cũ. Thí dụ cứ mỗi tuần chúng ta lại thực hiện một bản sao lưu toàn bộ rồi lưu cho bốn tuần sau đó. Nếu cần sao và tồn trữ lâu hơn nữa thì cứ mỗi nửa tháng chúng ta lại sao lưu toàn bộ cho một quý.

4. Thực hiện sao lưu và phục hồi tệp

Mục tiêu:

- Sử dụng được tiện ích *tar*

- Sử dụng được lệnh *cpio* để chép các tệp tồn trữ

Linux có nhiều tiện ích giúp chúng ta trong tác vụ này. Một số tiện ích thật đơn giản và một số khác phức tạp hơn. Càng đơn giản thì tiện ích càng bị giới hạn. Xin giới thiệu với chúng ta hai chương trình sau:

* **tar** sẵn có trên mọi hệ thống Linux hoặc UNIX.

* **cpio** sẵn có trên mọi hệ thống UNIX, dùng để sao chép tệp, **cpio** dễ dùng và mạnh hơn **tar**.

4.1. Tiện ích *tar*

Từ đầu, tiện ích **tar** của UNIX đã được thiết kế để tạo ra bản sao tồn trữ trên băng từ. Chúng ta có thể dùng **tar** để chép sang bất kỳ thiết bị nào, **tar** có nhiều thuận lợi như: dễ sử dụng, ổn định, có thể đọc tệp tồn trữ trên mọi hệ thống Linux hoặc UNIX.

Điểm bất tiện của một vài phiên bản **tar** là đòi hỏi phải lưu toàn bộ các tệp trên cùng một đĩa hoặc một băng. Do đó nếu có hỏng hóc một phần vật mang dữ liệu, chẳng hạn như đĩa bị bad sector hoặc băng bị bad block, thì coi như

chúng ta mất trắng toàn bộ bản sao lưu. Tiếp theo, **tar** không thể sao lưu các tệp đặc biệt như tệp thiết bị. Và bản thân **tar** chỉ có thể thực hiện sao lưu toàn bộ mà thôi. Nếu cần sao lưu tăng dần, chúng ta phải lập trình qua shell.

Bảng 9.1 Các tùy chọn thường dùng với lệnh tar

Tùy chọn	Mô tả
c	Tạo ra tệp tồn trữ.
x	Khai thác hoặc phục hồi tệp từ tệp tồn trữ trên thiết bị mặc định, hoặc trên thiết bị được xác định bằng tùy chọn f.
f tên	Tạo ra hoặc đọc tệp tồn trữ tên, với tên là tên của tệp hoặc tên của thiết bị được xác định ở /dev, chẳng hạn như /dev/rmt0.
Z	Nén hoặc bung tệp tar .
z	Nén hoặc bung tệp tar bằng gzip .
M	Tạo ra bảng sao lưu tar nhiều tệp.
t	Tạo ra chỉ mục tất cả các tệp lưu trong bản tồn trữ và liệt kê với stdout.
v	Chọn chế độ chi tiết.

Mời chúng ta xem vài thí dụ về cách sử dụng **tar** để sao lưu và phục hồi tệp. Lệnh sau đây sẽ chép thư mục /home vào đĩa mềm /dev/fd0:

```
tar -cf /dev/fd0 /home
```

Trong trường hợp này tùy chọn f cho biết tệp tồn trữ được chứa trên đĩa mềm /dev/fd0.

Lệnh sau đây cũng sao lưu thư mục /home:

```
tar -cvfzM /dev/fd0 /home | tee homeindex
```

Tùy chọn v chỉ định chế độ chi tiết, z cho biết phải nén tệp tồn trữ và M bắt **tar** tạo ra bản sao lưu có nhiều tệp. Mỗi khi đĩa mềm đã đầy, **tar** nhắc chúng ta đưa đĩa khác vào. Chỉ mục homeindex chứa danh sách các tệp được sao lưu để chúng ta tham khảo.

Lệnh **find** có ích trong việc tìm kiếm các tệp đã có thay đổi trong khoảng thời gian nào đó, với mục đích là sao lưu chúng theo thời biểu nhất định. Thí dụ sau đây dùng lệnh **find** để tạo danh sách các tệp đã thay đổi vào ngày cuối cùng:

```
find /home -mtime -1 -type f -print > bkuplst; tar cvfzM /dev/fd0 cat bkuplst | tee homeindex
```

Muốn dùng danh sách làm nhập liệu cho lệnh **tar**, chúng ta hãy đặt lệnh **cat bkuplst** trong dấu nháy ngược (còn gọi là dấu huyền), thí dụ: ``cat bkuplst``. Lúc này shell sẽ hiểu là phải thực hiện lệnh trong một shell thứ cấp và sau đó hiển thị kết quả tại vị trí của lệnh nào được đặt trong dấu nháy ngược.

Lệnh sau đây phục hồi tệp /home/dave/notes.txt từ thiết bị /dev/fd0:

```
tar xvf /dev/fd0/ home/dave/notes.txt
```

Lưu ý: Chúng ta có thể tự động hoá các lệnh vừa kể bằng cách đưa chúng vào tệp **crontab** ở root. Thí dụ chúng ta đưa mục ghi này vào tệp **crontab** để sao lưu thư mục /home mỗi ngày vào lúc 1 giờ 30 sáng:

```
30 01 * * * tar cvfz / def/fd0 /home >homeindex
```

Khi cần thực hiện các sao lưu phức tạp hơn, chúng ta tạo ra các shell script điều khiển. Các shell script như thế có thể chạy qua trung gian **cron**.

Ngoài ra chúng ta cũng có thể dùng lệnh **tar** để tạo các tệp tồn trữ bên trong hệ thống tệp của Linux thay vì ghi lên thiết bị sao lưu. Như thế, chúng ta có thể sao lưu một nhóm tệp cùng với cấu trúc thư mục của chúng chỉ vào một tệp duy nhất. Chúng ta chỉ cần ghi tên tệp vào chỗ đối số của tùy chọn **f** thay vì ghi tên thiết bị sao lưu. Sau đây là thí dụ về việc tạo tệp tồn trữ chứa các tệp của thư mục và thư mục cấp dưới bằng lệnh **tar**:

```
tar cvf home /sao_luu.tar /home/dave
```

Kết quả là chúng ta có tệp /home/sao_luu.tar chứa thông tin sao lưu của thư mục /home/dave cùng với tất cả các tệp và thư mục cấp dưới của /home/dave.

Ghi chú: Bản thân lệnh **tar** không thực hiện việc nén tệp. Muốn nén tệp **tar**, chúng ta xác định tùy chọn **z** với lệnh **tar**, hoặc chúng ta dùng chương trình nén, chẳng hạn như **gzip** đối với tệp **tar** sau cùng.

Khi sử dụng **tar** để tạo tệp tồn trữ, chúng ta nên khai báo mục ghi cao cấp nhất trong tệp **tar** là một thư mục. Như thế mỗi khi chúng ta khai thác, tất cả các tệp chứa trong tệp **tar** đó đều nằm trong một thư mục trung tâm của thư mục hiện hành. Nếu không, chúng ta sẽ rối lên với hàng trăm tệp được bung ra trong thư mục hiện hành.

Giả sử bên dưới thư mục hiện hành của chúng ta là thư mục mang tên data, trong đó có hàng trăm tệp. Có hai cách cơ bản để tạo tệp **tar** cho thư mục này. Cách thứ nhất, chúng ta chuyển sang thư mục data và tạo tệp **tar** từ đấy:

```
$ pwd
/home/dave
$ cd data
$ pwd
/home/dave/data
$ tar cvf.. /data.tar*
```

Kết quả là một tệp **tar** được tạo ra trong /home/dave chỉ chứa nội dung của thư mục data nhưng không chứa mục ghi nào cho thư mục. Khi ra lệnh khai thác tệp **tar** này, Linux sẽ không tạo ra một thư mục để chứa các tệp, ngược lại hàng trăm tệp sẽ được bung vào thư mục hiện hành.

Cách thứ hai để tạo tệp **tar** là khởi đầu từ thư mục mẹ của data và xác định đối tượng để sao lưu chính là tên thư mục, thí dụ:

```
$ pwd
/home/dave
```

\$ tar cvf data.tar data

Thí dụ trên cũng tạo ra một tệp tồn trữ của thư mục data, nhưng mục ghi thư mục là điều đầu tiên được tạo ra trong tệp tồn trữ ấy. Do đó sau này khi khai thác tệp **tar**, cái đầu tiên được tạo ra chính là thư mục data và tất cả các tệp bên trong thư mục data sẽ được xếp vào trong thư mục con data.

Ghi chú: Nếu muốn tạo ra tệp **tar** chứa tất cả các tệp trong thư mục, chúng ta nên xác định một vị trí riêng biệt cho tệp **tar**, không nằm trong thư mục hiện hành, lệnh **tar** sẽ không bị nhầm lẫn khi cố gắng đưa tệp **tar** sẵn có trước đó vào tệp **tar** đang được tạo ra lần thứ hai.

4.2. Sử dụng cpio

Lệnh tổng quát để chép các tệp tồn trữ là **cpio**. Chúng ta có thể tạo ra các bản sao lưu bằng cách sử dụng **cpio** với tùy chọn **-0**, hoặc để phục hồi tệp bằng tùy chọn **-i**. Lệnh **cpio** sẽ lấy nhập liệu từ đầu vào tiêu chuẩn (Stdin) và gửi xuất liệu đến đầu ra tiêu chuẩn (Stdout).

cpio có những lợi điểm như:

- Có khả năng sao lưu bất kỳ nhóm tệp nào.
- Có khả năng sao lưu những tệp đặc biệt.
- Lưu trữ thông tin hiệu quả hơn **tar**.
- Khi phục hồi dữ liệu, **cpio** bỏ qua các bad sector và bad block.
- Bản sao lưu của **cpio** có thể phục hồi được trên hầu hết các hệ thống

Linux hoặc UNIX.

Có người cho rằng cú pháp của **cpio** khó hơn cú pháp của **tar**. Ngoài ra khi thực hiện sao lưu tăng dần, chúng ta phải thực hiện lập trình qua shell.

Bảng 9.2 Các tùy chọn thường dùng với **cpio**

Tùy chọn	Mô tả
-0	Chép và tạo ra một tệp tồn trữ ở đầu ra tiêu chuẩn
-B	Hạn chế xuất liệu và nhập liệu ở mức 5.120 byte cho từng mẫu tin. Có ích khi lưu trữ trên băng từ.
-I	Khai thác và sao chép các tệp từ stdin. Được dùng đặc biệt khi Stdin lại là bản sao từ đầu ra của một lệnh cpio khác.
-t	Tạo bảng mục lục của nhập liệu.

Sau đây là vài thí dụ sử dụng **cpio** để sao lưu và phục hồi tệp: Lệnh **ls** sẽ sao chép các tệp trong thư mục /home sang thiết bị /dev/fd0, còn lệnh **find** sẽ cho chép toàn bộ cây thư mục của /home (kể cả thư mục con):

```
ls /home | cpio -0> /dev/fd0
```

```
find /home/. | cpio -ov> dev/fd0
```

Lệnh sau đây sẽ khai thác các tệp từ thiết bị /dev/fd0, đồng thời tạo ra bảng chỉ mục trong tệp bkup.indx:

```
cpio -it</dev/fd10>bkup.indx
```

Thí dụ sau đây sử dụng lệnh **find** để tạo ra danh sách những tệp nào của thư mục /home đã thay đổi nội dung vào ngày gần nhất:

```
cpio -i /home/dave/notes.txt < /dev/fd0
```

Ghi chú: Chúng ta phải ghi tên đầy đủ của tệp khi phục hồi tệp ấy bằng **cpio**

Lưu ý: Muốn tự động hoá tất cả các lệnh vừa kể, chúng ta đặt chúng vào tệp **crontab** của root. Thí dụ, chúng ta đặt mục ghi sau đây vào tệp **cron** ở root để máy tự động sao lưu thư mục /home vào lúc 1 giờ 30 mỗi ngày:

```
30 01 *** find /home/. | cpio -0 > /dev.f0
```

Nếu muốn thực hiện những sao lưu phức tạp hơn, chúng ta tạo ra các shell script điều khiển và chạy chúng qua **cron**.

Các bảng 11.1 và 11.2 đã liệt kê các tùy chọn thường dùng cho **tar** và **cpio**. Muốn biết đầy đủ chi tiết về các tùy chọn, mời chúng ta tham khảo bằng lệnh **man**.

B. CÂU HỎI VÀ BÀI TẬP

Câu 1: Trình bày các vấn đề về sao lưu?

Câu 2: Trình bày các thủ thuật sao lưu?

Câu 3: Nêu cách hoạch định thời biểu sao lưu?

Câu 4: Nêu cách sử dụng lệnh cpio?

BÀI 10: QUẢN LÝ TỆP VÀ THƯ MỤC

Mã bài: MD37-10

❖ Giới thiệu

Hầu hết các lệnh Linux (và do đó cả các shell script) đều tập trung vào thao tác trên các tệp và thư mục. Những thao tác này được các shell Linux diễn dịch dễ dàng nhờ một cú pháp thích hợp. Nói chung có thể chia các lệnh ấy thành hai hạng:

- Những lệnh thao tác bản thân tệp như là đối tượng
- Những lệnh chỉ thao tác nội dung tệp

Chương này cung cấp cho quản trị viên và độc giả quan tâm những lệnh coi bản thân tệp như là đối tượng, nghĩa là những lệnh có chức năng di dời, đổi tên, xoá bỏ, định vị và thay đổi các thuộc tính của tệp và thư mục. Ngoài ra ở đây cũng lướt nhanh qua những lệnh thao tác nội dung tệp.

❖ Mục tiêu

- Liệt kê được các tệp
- Tổ chức được các tệp
- Sao chép được các tệp
- Di dời và đặt tên lại tệp
- Xoá tệp hoặc thư mục
- Xem nội dung của tệp
- Duyệt tìm tệp
- Thay đổi nhãn ngày giờ
- Nén và nối tệp
- Rèn luyện khả năng tổ chức, quản trị.

❖ Nội dung chính

A. LÝ THUYẾT

1. Các thao tác cơ bản với tệp

Mục tiêu :

- Trình bày được các thao tác cơ bản với tệp
- Thực hiện được thành thạo các thao tác cơ bản với tệp

1.1. Liệt kê tệp

Lệnh cơ bản để liệt kê tệp là **ls**, hoạt động tùy theo cách ra lệnh. Nếu dùng trong ống dẫn (pipe), mỗi tệp sẽ hiển thị trên một dòng như mặc định của vài phiên bản UNIX, chẳng hạn như SCO. Những phiên bản khác liệt kê tệp thành nhiều cột.

Hầu hết người sử dụng đều thích hình thức cột. Phiên bản nào mặc định **ls** là lệnh liệt kê thành từng dòng thường có riêng một lệnh khác, chẳng hạn như **lc**, để liệt kê thành từng cột.

Hành động của lệnh **ls** thay đổi tùy theo các cờ (flag) đi kèm, dạng như -abcd. Nhìn chung các phiên bản của lệnh **ls** có hai dạng: dạng xuất phát từ UNIX System V và dạng xuất phát từ UNIX Berkeley. Bởi vì các hệ Berkeley dần dần hội tụ với System V cho nên chương này tập trung phân tích các cờ của System V.

Muốn biết mình đang dùng phiên bản **ls** nào, chúng ta xem tài liệu kèm theo hệ hoặc gõ lệnh **man ls**.

Các cờ được lệnh **ls** sử dụng có thể viết liền hay viết rời, kết quả như nhau. Thí dụ **ls -l -F** và **ls -lF** đều cho cùng một kết quả.

Bảng 10.1: Các cờ dùng với lệnh **ls**

Cờ	Mô tả
-a	Liệt kê mọi tệp vì ls mặc định không liệt kê những tệp bắt đầu bằng dấu chấm (thường đó là các tệp cấu hình có rất nhiều trong home directory nên sẽ làm rối mắt nếu ls lại mặc định là liệt kê mọi tệp)

-A	Như -a, song không liệt kê 2 bí danh “.” và “..”. Vì tên những tệp này bắt đầu bằng dấu chấm nên -a liệt kê cả chúng
-b	Buộc các ký tự phi đồ họa phải xuất hiện ở dạng bát phân \ddd, cờ -b tiện dùng hơn -q vì nó giúp chúng ta hình dung các ký tự ấy ra như thế nào
-c	Sử dụng thời gian của lần truy cập, chỉnh sửa gần nhất để phân loại. Linux dán 3 loại nhãn ngày giờ vào tệp: lúc tạo ra tệp, lần truy cập gần nhất và lần chỉnh sửa gần nhất. Các tệp thường liệt kê theo thứ tự abc, nhưng chữ in hoa được xếp trước chữ thường
-d tên_tệp	Nếu đối số là một thư mục, cờ này chỉ liệt kê tên (không liệt kê nội dung); thường dùng với -l để xem trạng thái thư mục. Thường nội dung thư mục chỉ được liệt kê khi nào được khẳng định là phải liệt kê, hoặc dùng chung với wildcard (ký tự đại diện). Do đó lệnh ls chỉ liệt kê tên thư mục, trong khi ls * lại liệt kê tệp, thư mục và nội dung của bất kỳ thư mục nào có mặt trong thư mục hiện hành
-F	Đánh dấu thư mục bằng dấu chéo /, đánh dấu tệp khả thi bằng dấu sao, các kết nối tượng trưng bằng dấu a vòng @, các FIFO bằng dấu và các rãnh cắm (socket) bằng dấu -
-i	In các inode tại cột đầu tiên của bản báo cáo. Nếu chúng ta liệt kê những tệp kết nối, nên lưu ý là chúng đều có cùng số inode.
-l	Liệt kê mục ghi thư mục ở dạng dài, bao gồm cả quyền hạn, số kết nối, chủ sở hữu, kích thước tính bằng byte và ngày giờ chỉnh sửa gần nhất. Nếu đây là tệp đặc biệt, trường kích thước sẽ chứa số hiệu thiết bị trường và thứ. Nếu thời gian chỉnh sửa cách nay hơn sáu tháng, sẽ hiển thị tháng, ngày và năm; ngược lại sẽ chỉ hiện ngày giờ. Nếu tệp là kết nối tượng trưng, tên đường dẫn của tệp đích sẽ được hiển thị sau dấu mũi tên. Chúng ta có thể phối hợp -l với các tùy chọn khác, thí dụ như -n
-n	Thay vì tên, sẽ hiển thị số ID của user và nhóm kết hợp với từng tệp và thư mục. Thông thường ls chỉ liệt kê tên. -n rất tiện lợi khi thiết lập mạng, chẳng hạn như TCP/IP, vì chúng ta cần biết số ID để cấp quyền hạn qua nhiều hệ thống khác nhau đang nối mạng
-q	Hiển thị các ký tự phi đồ họa trong tên tệp thành dấu hỏi. Đối với ls, đây là hành động mặc định khi xuất ra terminal. Nếu vô tình tệp được tạo ra bằng những ký tự không in được, cờ -q sẽ hiển thị tệp ấy
-r	Đảo ngược thứ tự sắp xếp abc hoặc thời gian
-s	Hiển thị kích thước tệp, gồm cả những block gián tiếp ánh xạ

	tệp. Nếu có biến môi trường POSIX_CORRECT, mỗi block sẽ là 512 byte
-t	Sắp xếp theo thứ tự thời điểm chỉnh sửa (gần nhất xếp trước) thay vì theo tên. Nếu muốn xem những tệp lâu đời trước, chúng ta gõ -rt.
-u	Sử dụng thời điểm truy cập gần nhất, thay vì thời điểm thay đổi gần nhất để sắp xếp (với -t) hoặc để in (với -l).
-x	Buộc Linux xuất mục ghi ra thành dạng nhiều cột dàn hàng ngang, thay vì ở cuối trang.

Nếu có bản Linux Slackware, chúng ta sẽ nhận thấy rằng ls cũng hiện các màu khác nhau cho từng loại tệp. Màu được xác định ở tệp cấu hình DIR_COLORS trong thư mục /etc (từ phiên bản 6.1 trở đi, RedHat cũng sử dụng màu khi liệt kê tệp).

Cấu hình mặc định bao gồm:

- Tệp khả thi: màu xanh lá cây (green)
- Thư mục: màu xanh biển (blue)
- Kết nối tượng trưng: màu lục lam (cyan).

Muốn căn chỉnh màu, chúng ta chép tệp DIR_COLORS về home directory rồi đổi tên thành .dir_colors.

Bảng sau liệt kê các định nghĩa màu khả dụng. Mời chúng ta tham khảo thêm các trang **man** về tệp DIR_COLORS.

Bảng 10.2: Các giá trị DIR_COLORS thể hiện màu sắc.

Giá trị	Mô tả
0	Phục hồi các màu mặc định
1	Cho vùng có màu sáng hơn.
4	Cho vùng văn bản gạch dưới.
5	Cho vùng văn bản nhấp nháy.
30	Cho vùng tiền cảnh màu đen.
31	Cho vùng tiền cảnh màu đỏ.
32	Cho vùng tiền cảnh màu xanh lá cây.
33	Cho vùng tiền cảnh màu vàng (hoặc nâu).
34	Cho vùng tiền cảnh màu xanh biển.
35	Cho vùng tiền cảnh màu đỏ tía.
36	Cho vùng tiền cảnh màu lục lam.
37	Cho vùng tiền cảnh màu trắng hoặc xám.
40	Cho vùng nền màu đen.
41	Cho vùng nền màu đỏ.
42	Cho vùng nền màu xanh lá cây
43	Cho vùng nền màu vàng hoặc nâu.
44	Cho vùng nền màu xanh biển.
45	Cho vùng nền màu đỏ tía.

46	Cho vùng nền màu lục lam.
47	Cho vùng nền màu trắng hoặc xám.

1.2. Tổ chức tệp

Linux không có quy định bất biến cho việc tổ chức tệp. Tệp của Linux không có phần mở rộng (thí dụ như phần đuôi .EXE cho những tệp khả thi của DOS). Chúng ta có thể và chúng ta nên tổ chức cách đặt tên tệp cho riêng mình.

Xin nhắc chúng ta rằng ở thuở ban đầu của Linux mỗi tên tệp đều ghi đầy đủ đường dẫn với các thư mục của nó.

Tuy nhiên, phải nhìn nhận việc ngày càng có nhiều ứng dụng Linux đến từ thế giới của DOS và các ứng dụng này mang theo những quy ước của mình đến cho Linux. Giới cung cấp phần mềm thường khuyên chúng ta nên đặt tên tệp có phần mở rộng và thích hợp với ứng dụng của tệp.

Nếu muốn tạo ra những lệnh riêng của mình, chúng ta nên tổ chức thư mục dựa theo cách của Linux, với những thư mục /bin, /lib và /etc. Hãy tạo cấu trúc thư mục cấp dưới trong thư mục /home của chúng ta. Để cho mọi việc được đơn giản, chúng ta nên đặt các lệnh thi hành trong /bin, các lệnh phụ trong /lib và các tệp cấu hình khởi động trong /etc. Và đây chỉ là một gợi ý chứ không phải là bắt buộc.

Chúng ta tạo ra thư mục bằng lệnh **mkdir** với cú pháp đơn giản như sau:

mkdir tên_thư_mục

Với tên_thư_mục là tên đặt cho thư mục mới. Trên nguyên tắc, để tạo ra một thư mục cấp dưới, chúng ta phải có quyền hạn ghi trong thư mục hiện hành. Tuy nhiên ở đây chúng ta tạo ra thư mục cấp dưới trong chính /home của chúng ta, do đó sẽ không gặp rắc rối gì cả.

Giả sử chúng ta tạo ra ba chương trình mang tên prog1, prog2 và prog3, tất cả đều đặt trong \$HOME/bin. Xin nhớ \$HOME chính là home directory của chúng ta.

Nếu muốn những chương trình riêng của mình được chạy như thể chúng là thành phần tiêu chuẩn của bộ lệnh Linux, chúng ta phải ghi \$HOME/bin vào biến môi trường PATH. Để thực hiện việc này, chúng ta dùng lệnh sau đây trong shell của Bourne và Korn.

```
PATH = $PATH :$HOME/bin; export PATH
```

Với shell C, chúng ta sẽ gõ như sau:

```
setenv PATH $PATH $HOME/bin
```

Ghi chú: \$HOME đại diện cho đường dẫn đầy đủ đến home directory của chúng ta. Nếu đó là /home/ams, thì \$HOME/bin sẽ được viết là /home/ams/bin.

Nếu các chương trình của chúng ta gọi đến chương trình phụ, chúng ta nên tạo thư mục cấp dưới trong thư mục \$HOME/lib. Chúng ta có thể tạo ra thư mục cấp dưới cho từng chương trình phụ. Thí dụ lệnh riêng pgm1 có thể gọi \$HOME/lib/pgm1a.

Tương tự, nếu lệnh prog1 yêu cầu phải có bảng cấu hình khởi động, chúng ta có thể đặt tên bảng đó là \$HOME/etc/pgm1.rc. Dữ liệu của chúng ta có thể đặt trong thư mục \$HOME/data/pgm1.

1.3. Sao chép tệp

Lệnh sao chép tệp có dạng:

cp tệp_nguồn tệp_đích

Chúng ta phải có quyền hạn đọc từ tệp_nguồn mà chúng ta định sao chép và quyền hạn ghi vào thư mục đích (và tệp mà chúng ta định ghi đè lên phải có thực).

Chúng ta nên lưu ý một số điểm như sau:

-Nếu tệp_đích là tên của một tệp có sẵn, thì chúng ta sẽ ghi đè lên tệp có sẵn ấy.

-Nếu gõ tên thư mục đích sau lệnh **cp**, tệp sẽ được chép vào thư mục ấy và vẫn giữ nguyên tên cũ. Thí dụ chúng ta gõ lệnh:

cp tệp_nguồn thư_mục_đích thì tệp sẽ được chép vào thư_mục_đích dưới dạng thư_mục_đích/tên_tệp_nguồn.

Chúng ta có thể chép một danh sách các tệp1, tệp2, tệp3 vào thư_mục_đích bằng lệnh:

cp tệp1, tệp2, tệp3 ... thư_mục_đích

Nếu thư_mục_đích không phải là một thư mục, máy sẽ báo lỗi. Ngoài thư_mục_đích, nếu bất kỳ thành phần nào của danh sách tệp lại là một thư mục, thì máy cũng báo lỗi.

Hãy cẩn thận khi dùng wildcard (ký tự thay thế) với lệnh **cp**, bởi vì có khả năng rủi ro là chúng ta sẽ chép nhiều thứ hơn là dự định.

Ghi chú: Những người sử dụng Linux thường để trong máy nhiều tệp dạng DOS, đồng thời làm cho Linux truy cập được hệ thống tệp DOS. Hầu hết các lệnh Linux nhận biết khi nào một phân vùng DOS là đích hoặc nguồn, cho nên trong tiến trình chép tệp, DOS có thể xử lý việc diễn dịch tệp. Thông thường khi mở ra và lưu lại một tệp văn bản của UNIX/Linux trong DOS bằng WordPad, WordPad luôn lưu thêm ký tự về đầu dòng (CR) cạnh ký tự xuống dòng (LF), trong khi các hệ Linux và UNIX chỉ dùng một ký tự LF để báo hiệu về đầu dòng mới.

1.4. Di dời và đặt tên lại tệp

Lệnh **mv** của Linux giúp chúng ta di dời và đặt lại tên cho cả tệp lẫn thư mục. Cú pháp và quy định của **mv** cũng giống như của **cp**, nghĩa là chúng ta toàn quyền di dời bao nhiêu tệp cũng được, miễn là cuối lệnh phải có tên thư mục và chúng ta phải có quyền ghi đối với thư mục đó. Chỉ khác là **cp** không thể di dời và đặt lại tên thư mục.

Khi chúng ta ra lệnh di dời hoặc đặt lại tên tệp, thực ra chỉ có mục ghi trong tệp thư mục là thay đổi. Ngoại trừ trường hợp vị trí mới nằm trên ổ đĩa

hoặc phân vùng vật lý khác, nếu không tệp và nội dung thư mục không di dời gì cả.

Nếu chúng ta gõ lệnh **rm** (xoá) hoặc **cp** mà không cho thêm tuỳ chọn về thư mục, máy sẽ không thi hành lệnh và báo lỗi rằng chúng ta đang xử lý một thư mục. Muốn xoá hoặc chép thư mục, chúng ta phải dùng cờ -r với **rm** và **cp**. Tuy nhiên lệnh **mv** lại di dời thư mục không gặp vấn đề gì.

1.5. Xoá tệp hoặc thư mục

Như chúng ta đã biết, muốn xoá bỏ tệp hoặc thư mục, chúng ta dùng lệnh **rm**. Muốn xoá tệp không thuộc quyền sở hữu của mình, chúng ta phải có quyền hạn ghi trên thư mục chứa tệp cần xoá.

Nếu đang làm chủ tệp, tất nhiên là chúng ta có thể tuỳ ý xoá bỏ, nhưng với điều kiện là quyền hạn của thư mục chứa tệp đó cho phép chúng ta ghi.

Lệnh **rm** * sẽ xoá toàn nhóm tệp mà chúng ta có quyền ghi trong thư mục hiện hành, nhưng thư mục cấp dưới không bị ảnh hưởng. Muốn xoá thư mục cấp dưới, chúng ta phải dùng tuỳ chọn -r (recursive, đệ quy).

Ở một vài phiên bản, **rm** sẽ tạm dừng để hỏi chúng ta có thực sự muốn xoá tệp mà chúng ta có quyền sở hữu nhưng lại không có quyền hạn ghi. Một vài phiên bản khác lại thắc mắc khi chúng ta gõ lệnh **rm** kèm với wildcard. Đương nhiên chúng ta có thể viết riêng cho mình một macro hoặc một shell script để tạo cơ hội suy nghĩ trước khi khẳng định xoá tệp.

Nếu phiên bản **rm** đang sử dụng dẫn đo khi chúng ta ra lệnh xoá những tệp mà mình sở hữu nhưng lại không có quyền hạn ghi, chúng ta có thể phòng ngừa việc lỡ tay xoá mọi thứ trong thư mục bằng lệnh sau đây:

```
# touch "0 0"
```

Lệnh trên tạo ra tệp mang tên "0 0". Trong chuỗi ASCII, ký tự số "0" được sắp xếp trước mọi ký tự chữ, do đó khi gõ lệnh **rm** * thì **rm** sẽ thử xoá tệp "0 0" đầu tiên và dừng lại hỏi. Nếu quả thật không muốn xoá mọi thứ trong thư mục, chúng ta còn kịp bấm hoặc <Ctrl-c> để huỷ (kill) tiến trình **rm**. Muốn thử, chúng ta hãy xoá tệp "0 0" xem. Đừng gõ **rm** * nữa nếu phiên bản chúng ta đang dùng lại không dừng lại hỏi.

Một cách hay hơn nữa để phòng trường hợp lỡ tay xoá tất là chúng ta dùng cờ -i với **rm**. Ở đây -i viết thay chữ interactive (tương tác). Nếu gõ **rm** -i tên_tệp, máy sẽ yêu cầu chúng ta khẳng định. Và chúng ta phải trả lời "yes" trước khi tệp được xoá. Nếu chúng ta gõ **rm** -i *, máy sẽ buộc chúng ta trả lời cho từng tệp trong thư mục.

Khi phải thường xuyên sử dụng **rm** -i, chúng ta có thể cài lệnh này vào trong một shell script hoặc tạo ra một chức năng shell. Nếu thảo ra shell script, chúng ta nên nhớ là shell sẽ duyệt qua các lệnh trong thư mục liệt kê ở biến PATH theo đúng thứ tự liệt kê. Nếu thư mục \$HOME/bin của chúng ta ở dưới chót, shell script mang tên **rm** sẽ không bao giờ được tìm thấy. Chúng ta có thể

đặt thư mục \$HOME/bin ở đầu danh sách của biến PATH, hoặc tạo ra một lệnh mới, chẳng hạn như **del**. Nếu tạo ra shell script mang tên "**del**", chúng ta phải đánh dấu "khả thi" bằng lệnh **chmod** để shell có thể nhận ra nó. Khi tạo ra lệnh **del**, chúng ta chỉ phải ghi một lệnh duy nhất là **rm -i \$***.

Bởi vì khi gõ lệnh **del ***, shell sẽ diễn dịch ra là **rm -i ***.

Còn một cách khác để xoá tệp là dùng alias (bí danh). Chúng ta có thể hiểu alias như là một lệnh shell nội bộ (giống các lệnh doskey như của DOS kể từ phiên bản 5.0).

Trong khi đang dùng shell C, nếu muốn thêm alias chúng ta phải chỉnh sửa lại tệp mang tên .cshrc. Chúng ta có thể sử dụng bất kỳ trình soạn thảo văn bản nào, chẳng hạn như vi (xem "Sử dụng trình soạn thảo vi") để chỉnh sửa tệp này. Với shell C, chúng ta thêm những dòng sau đây vào đầu tệp .cshrc:

```
rm()
{
/bin/rm -i $*
}
```

Muốn thêm **alias** vào shell Korn, chúng ta thêm những dòng sau vào tệp \$HOME/.kshrc

```
alias rm 'rm -i $*'
```

Khi định xoá một thư mục bằng lệnh **rm**, máy sẽ báo đấy là thư mục và không được xoá. Nếu muốn xoá các thư mục rỗng, chúng ta dùng lệnh **rmdir** như với DOS.

Linux còn một cách khác để xoá thư mục cùng với nội dung, nhưng cách này rất nguy hiểm: lệnh **rm -r** sẽ xoá bất kỳ thư mục và tệp nào được phát hiện.

Thí dụ chúng ta có thư mục ./foo chứa tệp và thư mục cấp dưới, lệnh **rm -r foo** sẽ xoá sạch nội dung thư mục ./foo, kể cả các thư mục cấp dưới.

Với lệnh **rm -i -r**, máy sẽ yêu cầu chúng ta khẳng định trước khi thực hiện thao tác xoá. Trong trường hợp thư mục chứa rỗng, **rm** sẽ không tự ý xoá, giống như trường hợp chúng ta ra lệnh **rm** mà không kèm tùy chọn nào.

Ghi chú: Chúng ta không nhất thiết phải đặt từng cờ cho mỗi lệnh Linux. Nếu cờ nào không cần đối số kèm theo, chúng ta có thể phối hợp chúng với nhau. Thí dụ, **rm -i -r** có thể được viết là:

```
rm -ir
```

1.6. Xem nội dung của tệp

Hầu hết các lệnh Linux đều đưa kết quả ra thiết bị xuất chuẩn, tức màn hình. Những lệnh này thường hiển thị ngay lên màn hình trong hoặc sau khi thao tác tệp hoặc nhập liệu trực tiếp từ bàn phím. Muốn tệp được hiển thị theo ý mình, chúng ta hãy chọn một lệnh Linux thích hợp, trong đó có ba lệnh tiêu chuẩn sẽ được bàn kỹ hơn ở dưới đây, đó là: **cat**, **more** và **less**.

1.6.1. Các thiết bị xuất nhập chuẩn

Linux, cũng như mọi hệ thống UNIX, khi khởi động sẽ mở ra bốn (tệp) thiết bị xuất nhập chuẩn đó là: nhập tiêu chuẩn (stdin), xuất tiêu chuẩn (stdout), lỗi tiêu chuẩn (stderr) và cổng phụ (AUX). Những thiết bị này được coi như mặc định, tức là không cần nêu tên trong dòng lệnh.

Đầu vào/ra	Tiếng Anh	Tên	Thiết bị mặc định
Nhập tiêu chuẩn	standard input	stdin	Bàn phím
Xuất tiêu chuẩn	standard output	stdout	Màn hình
Lỗi tiêu chuẩn	standard error	stderr	Màn hình
Cổng phụ	auxiliary	AUX	Cổng COM

Ghi chú: Nếu không muốn dùng những giá trị mặc định, chúng ta có thể dùng các dấu > và < với tên (tệp) để chỉnh hướng các đầu vào/ra theo ý mình.

1.6.2. Xem tệp bằng lệnh cat

cat là một lệnh dùng để xem những tệp văn bản ngắn dạng ký tự ASCII. **cat** tên_tệp mở một tệp, sau đó hiển thị hết nội dung lên màn hình. Nếu đối số là nhiều tệp thì **cat** móc nối chúng lại với nhau trước khi đưa lên màn hình, tệp này tiếp nối tệp kia, thí dụ: **cat** tệp1 tệp2 tệp3

Khi dùng **cat** để hiển thị một tệp dài hơn 1 trang, chúng ta sẽ thấy nội dung của tệp khó đọc vì các dòng trôi nhanh như chạy. Muốn đọc, chúng ta phải luân phiên bấm <Ctrl-s> và <Ctrl-q> để máy biết khi nào thì cho trôi hay dừng lại. Nếu không, chúng ta có thể xem từng trang màn hình bằng các lệnh **more** hoặc **less**.

1.6.3. Xem tệp bằng lệnh more

Cả lệnh **more** và **less** đều hiển thị mỗi lần một trang màn hình. Căn cứ vào biến môi trường TERM và cơ sở dữ liệu về các loại terminal có trên máy, lệnh **more** và **less** sẽ xác định mỗi trang đó gồm bao nhiêu dòng.

Lệnh **more** ra đời trước **less**, xuất xứ từ phiên bản Berkeley UNIX. Lệnh này ngày càng chứng tỏ hiệu quả của mình nên đã trở thành một tiêu chuẩn, cũng như trường hợp của trình soạn thảo vi.

Cũng như **cat**, hình thức đơn giản nhất của lệnh **more** là:
more tên_tệp

Máy sẽ hiển thị trang màn hình đầu tiên của tệp. Muốn xem tiếp trang sau, chúng ta bấm thanh phím Spacebar, còn nếu bấm phím <Enter> thì màn hình chỉ dịch lên một dòng.

Nếu đang xem một chuỗi tệp (bằng lệnh **more** tệp1 tệp2 ...) và muốn dừng lại để chỉnh sửa tệp nào đó, chúng ta bấm phím <e> hoặc <v>.

Bấm phím <e> với **more** nghĩa là chúng ta gọi một trình soạn thảo đã được xác định trong biến môi trường shell EDIT liên quan đến tệp hiện hành.

Nếu bấm <v>, máy sẽ gọi trình soạn thảo nào được xác định trong biến môi trường VISUAL. Nếu chúng ta chưa định nghĩa các biến này trong môi

trường, **more** sẽ mặc định dùng trình soạn thảo **ed** khi chúng ta bấm <e> và trình soạn thảo **vi** khi bấm <v>.

Với lệnh **more**, nếu muốn xem lại trang màn hình đã lật qua, chúng ta dùng phím Ngoài ra lệnh **less** cũng có thể giúp chúng ta việc này.

1.6.4. Xem tệp bằng lệnh **less**

Điểm bất tiện của lệnh **less** là chúng ta không thể sử dụng phần mềm soạn thảo với tệp đang được hiển thị. Tuy nhiên để bù lại, lệnh **less** giúp chúng ta di chuyển tới lui khi xem tệp.

Công dụng của lệnh **less** gần giống lệnh **more**. Để xem từng trang của tệp, chúng ta gõ:

Máy sẽ hiển thị đầy một trang màn hình. Cũng như với lệnh **more**, chúng ta bấm phím Spacebar để xem trang tiếp theo.

Muốn xem màn hình vừa trôi qua, chúng ta bấm . Muốn di chuyển đến vị trí nào đó trong tệp (vị trí này được biểu thị bằng số phần trăm), chúng ta bấm <p> và khi dấu nhắc <:> hiện ra, chúng ta gõ số phần trăm vào.

1.6.5. Duyệt tìm xuyên tệp và thoát khỏi shell

Hai lệnh **less** và **more** giúp chúng ta duyệt suốt tệp đang hiển thị để tìm ra một chuỗi ký tự. Lệnh **less** còn cho phép chúng ta tìm ngược về phía đầu tệp bằng cú pháp sau:

less / chuỗi_ký_tự

Sau khi tìm thấy chuỗi_ký_tự, lệnh **less** và **more** sẽ hiển thị chuỗi ấy ở đầu một trang mới mở. Với **less**, khi chúng ta bấm <n> máy sẽ lặp lại thao tác tìm vừa rồi.

Khi dùng hai lệnh **less** và **more**, nếu bấm <!> chúng ta sẽ thoát khỏi shell. Thực ra khi thoát khỏi bằng lệnh <!>, chúng ta đang ở trong một shell thứ cấp. Từ đây chúng ta phải thoát khỏi giống như chúng ta đăng xuất một phiên làm việc bình thường.

Tuỳ vào shell đang sử dụng, chúng ta có thể bấm <Ctrl-d> hoặc gõ **exit** để trở về màn hình **more** hoặc **less** mà chúng ta vừa thoát khỏi.

Nếu bấm <Ctrl-d> mà máy yêu cầu chúng ta dùng lệnh **logout** thay vì <Ctrl-d>, chúng ta hãy làm theo máy.

1.6.6. Xem tệp bằng những cách khác

Có thể hiển thị nội dung tệp bằng nhiều cách khác. Thí dụ, nếu muốn xem nội dung của một tệp nhị phân, chúng ta dùng lệnh **od** (từ chữ **o**ctal **d**ump). Lệnh **od** hiển thị tệp dưới dạng mặc định là bát phân (hệ đếm cơ số 8). Dùng kèm với những cờ khác, **od** có thể hiển thị tệp nhị phân dưới dạng thập phân, ASCII, hoặc thập lục phân (cơ số 16).

Sở dĩ phải hiển thị tệp theo những gián tiếp như dạng bát phân, thập phân, hoặc thập lục phân là vì hiển thị dữ liệu ở dạng nhị phân vẫn còn là một vấn đề nan giải. Nếu dữ liệu được thể hiện ở dạng ASCII thì rất dễ đọc, vì các ký tự

ASCII là dạng mà mọi người nhìn thấy khi xem hầu hết các tệp văn bản. Song nếu tệp là một chương trình nhị phân thì dữ liệu không thể được trình bày ở dạng ASCII. Trong trường hợp này chúng ta phải hiển thị bằng hình thức chuyển đổi hệ đếm số.

UNIX ra đời trên những máy minicomputer sử dụng các ký tự 12 bit. Chúng ta ngày nay thường lấy byte (8 bit) làm một chuẩn độ dài bộ nhớ. Mặc dù có thể hiển thị theo hệ thập phân quen thuộc, song vấn đề là hiển thị cái gì - một byte, một chữ, hay 32 bit? Thao tác hiển thị một số lượng bit nào đó, đòi hỏi phải chuyển đổi hệ nhị phân căn cứ trên số lượng bit ấy. Với hệ 12 bit chúng ta có thể hiển thị toàn bộ 12 bit với bốn con số (hiển thị bằng 2^3 , dưới dạng bát phân). Các hệ UNIX trước đây đều chạy trên những máy tính 12 bit như thế, do đó phần lớn các phép ghi của UNIX (và do đó của Linux) đều ở dạng bát phân. Bất cứ byte nào cũng có thể hiển thị bằng một mã bát phân với ba chữ số có dạng như sau (thí dụ hiển thị giá trị thập phân của 8):

\010

Vì thế giới ngày nay đã chuyển sang dạng byte 8 bit nên bát phân không còn hiệu quả khi trình bày dữ liệu. Hệ thập lục phân (cơ sở 16 hoặc 2^4) là cách tốt hơn.

Một byte 8 bit có thể trình bày bằng hai chữ số thập lục phân. Thí dụ byte có giá trị thập phân là 10 sẽ được hiển thị thành 0A theo cơ sở 16.

Lệnh **od** giúp chúng ta chọn lựa cách hiển thị dữ liệu. Hình thức tổng quát như sau:

od [tùy_chọn] [tệp]

hoặc:

od - traditional [tệp] [[+]offset[[+]nhãn]]

Bảng 10.3: Các cờ dùng với lệnh **od**

Cờ ngắn	Cờ dài	Mô tả
-A	--address-radix=cơ sở	Xác định cách in những khoảng trống offset trong tệp
-N	--read-bytes=byte	Giới hạn phân kết xuất bằng lượng byte nhập liệu cho mỗi tệp
-j	--skip-bytes=byte	Bỏ qua lượng byte nhập liệu đầu tiên ở mỗi tệp
-s	--strings[=byte]	Xuất chuỗi với ít nhất là bao nhiêu byte ký tự đồ họa
-t	--format=loại	Chọn một hoặc nhiều dạng xuất liệu
-v	--output-duplicates	Không cho phép dùng * để đánh dấu việc bỏ dòng
-w	--width[=byte]	Xuất bao nhiêu byte ở mỗi dòng xuất liệu
	--traditional	Chấp nhận đối số ở dạng pre-POSIX

	--help	Hiển thị phần trợ giúp này và thoát khỏi
	--version	Xuất thông tin phiên bản và thoát khỏi

Chúng ta có thể dùng chung dạng thức pre-POSIX ở bảng sau đây với các lệnh ở bảng trên và nhận được kết quả tổng hợp.

Bảng 10.4: Các đặc tả dạng pre-POSIX dùng với lệnh **od**.

Cờ ngắn	Tương đương POSIX	Mô tả
-a	-t a	Chọn các ký tự có tên
-b	-t c	Chọn các byte bát phân
-c	-t c	Chọn các ký tự ASCII hoặc backlash escape
-d	-t u2	Chọn các thập phân ngắn không đánh dấu
-f	-t fF	Chọn các float
-h	-t x2	Chọn các thập lục phân ngắn
-I	-t d2	Chọn thập phân ngắn
-l	-t d4	Chọn thập phân dài
-o	-t o2	Chọn bát phân ngắn
-x	-t x2	Chọn thập lục phân ngắn

Trong bảng trên, kích cỡ là một số và cũng có thể là C cho sizeof(char), S cho sizeof(short), I cho sizeof(int), hoặc L cho sizeof(long). Nếu là loại f, kích cỡ có thể là F cho sizeof(float), D cho sizeof(double), hoặc L cho sizeof(Long double).

Ghi chú: Trong ngôn ngữ C, sizeof là một chức năng phân hồi số byte trong cấu trúc dữ liệu được dùng làm tham số. Thí dụ chúng ta sẽ dùng chức năng gọi sau đây để xác định số byte trong một số nguyên trên hệ thống máy, bởi vì số byte trong một số nguyên tùy thuộc vào hệ thống:

sizeof(int); cỡ số trong bảng 17.3 đại diện cho số hệ thống và sẽ là d cho thập phân deca, O cho bát phân (octa), X cho thập lục (hexa), hoặc n cho none (không có). Byte là thập lục phân với tiền tố 0x hoặc 0X; byte được nhân với 512 nếu có hậu tố b (bit), nhân với 1024 nếu có k (kilo) và với 1.048.476 nếu có m (mega). Nếu không có con số nào, -s sẽ mặc định 3, trong trường hợp tương tự, -w sẽ mặc định 32. Theo mặc định, **od** dùng -A0 -td2 -w16.

1.7. Duyệt tìm tệp

Nếu dùng lệnh **ls** tìm vẫn không ra một tệp, chúng ta thử dùng lệnh **find**. **find** là công cụ cực kỳ mạnh và cũng chính vì thế mà **find** lại trở thành một trong những lệnh khó dùng nhất. **find** có ba phần, mỗi phần lại có thể gồm nhiều phần nhỏ:

- Tìm ở đâu
- Tìm cái gì
- Tìm được rồi thì thực hiện thao tác nào.

Nếu biết tên tệp nhưng không biết vị trí tệp ấy, chúng ta gõ lệnh như sau:

find /-name tên_tệp -print

Ghi chú: Hãy cẩn thận khi duyệt tìm từ thư mục gốc. Trên những hệ thống lớn, nếu tìm từ thư mục gốc, máy sẽ lục lọi mọi thư mục bên dưới rồi đến phiên các thư mục cấp dưới nữa, chưa kể những ổ đĩa khác tại chỗ và ổ đĩa từ xa, do đó sẽ tốn rất nhiều thì giờ.

Do đó có lẽ chúng ta nên giới hạn việc duyệt tìm ở một hai thư mục thôi. Thí dụ, khi biết có tệp ở thư mục /usr hoặc /usr2, chúng ta gõ lệnh:

```
find /usr /usr2 -name tên_tệp -print
```

Chúng ta có thể dùng nhiều tùy chọn với **find**. Bảng sau đây chỉ liệt kê một phần nhỏ. Muốn tham khảo mọi tùy chọn, chúng ta gõ lệnh:

```
man find
```

Bảng 10.6: Một mẫu các cờ dùng với lệnh **find**.

Lệnh	Mô tả
- name tệp	Biến tệp có thể là tên của một tệp hoặc tên tệp kèm ký tự wildcard. Nếu có kèm wildcard, những tệp nào khớp với wildcard sẽ được chọn để xử lý.
-links n	Bất kỳ tệp nào có số kết nối bằng hoặc lớn hơn n sẽ được chọn. Chúng ta thay thế n bằng con số cần thiết.
-size n [c]	Bất kỳ tệp nào có số block 512 byte bằng hoặc lớn hơn n sẽ được chọn. Nếu có c gắn theo n, máy sẽ chọn những tệp có số ký tự bằng hoặc lớn hơn n.
-atime n	Tệp nào được truy cập trong vòng n ngày sẽ được chọn. Chúng ta lưu ý là thao tác duyệt tìm tệp bằng lệnh find sẽ thay đổi nhãn ngày tháng của tệp.
-exec cmd	Sau khi chọn xong danh sách tệp, chúng ta có thể dùng danh sách này làm đối số để chạy một lệnh Linux. Có hai quy định đơn giản với -exec: tên của tệp được đại diện bởi {} và lệnh phải tận cùng bằng dấu chấm phẩy thoát, nghĩa là \;. Giả sử khi đã đăng nhập với quyền hạn root, chúng ta tạo ra một thư mục user. Kết quả là mọi tệp đều thuộc về root, nhưng lẽ ra chúng phải thuộc quyền sở hữu của user đó. Chúng ta phải ra lệnh sau đây để đổi chủ sở hữu của mọi tệp trong /usr/pla và mọi thư mục cấp dưới từ root sang pla: find /home/pla -exec chown pla {} \;
-print	In tên và vị trí các tệp được chọn.

Lệnh **find** giúp chúng ta thực hiện nhiều thử nghiệm logic đối với tệp. Thí dụ chúng ta muốn tìm một loạt những tên tệp nhưng không thể dùng wildcard để đại diện, vì không khớp hoàn toàn. Trường hợp này chúng ta dùng tùy chọn or (-o) như sau:

```
find /home (-name tệp1 -o -name tệp2) -print
```

1.8. Thay đổi nhãn ngày giờ

Mỗi tệp Linux duy trì ba nhãn ngày giờ:

- Ngày giờ tạo ra tệp
- Ngày giờ lần thay đổi gần nhất và
- Ngày giờ lần truy cập gần nhất

Chúng ta có thể thay đổi nhãn thời gian tạo ra tệp, trừ khi chúng ta thực hiện thao tác sao chép và đặt lại tên tệp. Mỗi khi tệp được mở hoặc được đọc, nhãn thời gian truy cập sẽ thay đổi. Và như đã nói ở đoạn trên, dùng lệnh **find** duyệt tìm tệp cũng khiến nhãn ngày tháng truy cập thay đổi.

Nhãn thời gian của tệp có ích khi chúng ta cần sao lưu có chọn lọc những tệp nào đã thay đổi ở thời điểm nhất định. Ở đây chúng ta cũng dùng lệnh **find**.

Nếu muốn thay đổi nhãn thời gian trong khi vẫn giữ nguyên tệp, chúng ta dùng lệnh **touch** (sửa lại). Theo mặc định, **touch** cập nhật nhãn thời gian truy cập và chỉnh sửa theo đồng hồ hệ thống hiện hành. Cũng theo mặc định, nếu chúng ta định thao tác một tệp hiện vẫn chưa có thực, **touch** sẽ tạo ra tệp ấy.

Chúng ta có thể dùng **touch** để "đánh lừa" lệnh nào có chức năng kiểm tra ngày tháng. Thí dụ nếu hệ thống máy chạy sao lưu một số tệp đã thay đổi vào thời gian nào đó, chúng ta có thể **touch** một tệp để tệp ấy được hay không được tính đến.

Lệnh **touch** có ba cờ sau đây:

-a	Chỉ cập nhật nhãn thời gian truy cập tệp.
-m	Chỉ cập nhật nhãn thời gian chỉnh sửa tệp.
-c	Không cho touch tạo ra một tệp mới.

Cú pháp mặc định của **touch** là:

touch -am danh_sách_tệp

2. Nén và nói tệp

Mục tiêu :

- Trình bày được cách nén và nói tệp

Nếu sức chứa của hệ thống có hạn, hoặc nếu có những tệp ASCII quá lớn mà ít khi dùng đến, chúng ta có thể nén chúng lại để tiết kiệm khoảng trống trên đĩa. Tiện ích tiêu chuẩn của Linux để nén tệp là **gzip**. Lệnh này có thể nén tệp ASCII còn chừng 20% kích thước ban đầu.

gzip tên_tệp

Hầu hết các hệ UNIX đều có lệnh **compress**, thường dùng với tar để nén từng nhóm tệp dùng cho tồn trữ. Tệp nén bằng lệnh **compress** có đuôi .z, thí dụ archive1.tar.z. Bản phát hành Red Hat còn có các chương trình **zip** và **unzip** để nén và tồn trữ một loạt các tệp.

Sau khi nén thành công một tệp bằng lệnh **gzip** tên_tệp, tệp nén sẽ mang tên dạng tên_tệp.gz, sau đó tệp nguyên thủy bị huỷ. Muốn phục hồi tệp nén về trạng thái cũ, chúng ta dùng lệnh sau:

gunzip tên_tệp

Ghi chú: Khi bung một tệp, chúng ta không cần phải gõ đủ cái đuôi .gz phía sau tên tệp. Lệnh **gunzip** sẽ tự động hiểu.

Trong trường hợp muốn giữ tệp ở thể nén, những lại muốn đặt ống dẫn dữ liệu sang một lệnh khác, chúng ta dùng lệnh **zcat**. Lệnh này hoạt động như lệnh **cat**, song đòi hỏi phía đầu vào phải là tệp nén. **zcat** bung tệp rồi hiển thị lên thiết bị kết xuất tiêu chuẩn.

Thí dụ. chúng ta đã nén một loạt nhiều tên và địa chỉ trong tệp mang tên **namelist** và tệp nén xong mang tên **namelist.gz**. Giờ đây nếu muốn dùng nội dung tệp nén để làm nhập liệu cho một chương trình, chúng ta dùng lệnh **zcat** để bắt đầu một ống dẫn:

```
zcat namelist | chươngtrình1 | chươngtrình2
```

zcat cũng có cùng những hạn chế như **cat**: không thể trở lui về sau trong một tệp. Linux có chương trình **zless** hoạt động như **less**, chỉ khác là **zless** làm việc với tệp nén. Những lệnh nào làm việc được với **less** sẽ làm việc được với **zless**.

Những chúng ta nào đã quen sử dụng **pkzip** của **pkware** giờ có thể dùng **zip** và **unzip** trong bản phát hành **Red Hat**, vì chúng hoạt động tương tự nhau. Mời chúng ta đọc các trang **man** của **zip** và **unzip** để biết thêm chi tiết.

3. Hệ thống thư mục trong Linux

Mục tiêu :

- Trình bày được thư mục UNIX cổ điển
- Nêu được Các thư mục trong Linux

3.1. Thư mục UNIX cổ điển

Trước UNIX System V Release 4 (thí dụ UNIX System V Release 3.2 và các bản trước đó), hầu hết các bản UNIX đều có dạng tổ chức thư mục như sau:

```
/
/etc
/lib
/tmp
/bin
/usr
    /spool
    /bin
    /include
    /tmp
    /adm
    /lib
```

Thư mục `/etc` chứa hầu hết các dữ liệu liên quan đến hệ thống và cần thiết để khởi động máy. Nó chứa các tệp như `passwd` và `inittab`, rất cần thiết để hệ thống hoạt động tốt.

Thư mục `/lib` chứa thư viện các hàm (chức năng) dành cho trình biên dịch C. Ngay cả khi hệ thống của chúng ta không có trình biên dịch này, thư mục `/lib` cũng quan trọng bởi vì nó chứa mọi hàm C dùng chung mà nhiều chương trình ứng dụng sẽ truy cập. Thư viện các hàm C dùng chung (cũng gọi là thư viện chia sẻ) chỉ được nạp vào bộ nhớ khi nào có chương trình gọi nó được thực hiện. Như thế các chương trình khả thi sẽ nhỏ gọn, nếu không chúng lại phải cộng thêm những đoạn mã chức năng giống nhau làm tốn cả đĩa cứng lẫn bộ nhớ.

Thư mục `/tmp` dùng để lưu trữ tạm thời. Thông thường, chương trình nào dùng đến `/tmp` sẽ tự động xoá sạch các tệp mà nó tạm lưu trong thư mục này sau khi chạy xong. Và bởi vì thỉnh thoảng hệ thống lại tự động dọn `/tmp`, chúng ta đừng lưu những gì quan trọng trong thư mục này.

Thư mục `/bin` và `/sbin` chứa mọi tệp khả thi cần thiết để khởi động hệ thống và những lệnh thường dùng nhất của Linux. Tuy nhiên, chúng ta nên nhớ là tệp khả thi không nhất thiết phải là tệp nhị phân. Một số tệp nhỏ trong `/bin` thực ra chỉ là shell script.

Thư mục `/usr` chứa tất cả mọi thứ còn lại. Biến `PATH` của chúng ta có chuỗi `/bin:/usr/bin` bởi vì trong thư mục `/usr/bin` chứa mọi lệnh Linux nào không có mặt trong thư mục `/bin`. Việc này có lý do lịch sử của nó. Linux cần ít nhất ba thư mục `/etc`, `/tmp` và `/bin` để tự thi hành. Vào thuở Linux còn sơ khai, đĩa cứng không có dung lượng lớn và chỉ đủ sức chứa ba thư mục ấy mà thôi, tất cả những gì còn lại đều phải đặt trên đĩa khác, đợi sau khi Linux chạy được rồi mới nạp vào. Trước đây khi Linux hãy còn là một hệ điều hành tương đối nhỏ, việc đặt thêm các thư mục con vào thư mục `/usr` không mấy khó khăn, do đó Linux có thể chạy với hai đĩa: đĩa root và đĩa `/usr`.

Thư mục `/usr/adm` chứa mọi thông tin thống kê và chẩn đoán cần thiết cho quản trị viên hệ thống. Hiện nay đó là thư mục `/var`. Nếu cả hai chương trình thống kê và chẩn đoán đều không được bật lên, thư mục này sẽ trống.

Thư mục `/include`, hiện nay là `/usr/include`, chứa mọi đoạn mã nguồn cần thiết cho những biểu thức `#include` trong các chương trình C. Chúng ta phải có ít nhất quyền hạn đọc đối với thư mục này bởi vì trong đó có tất cả đoạn mã và cấu trúc hình thành hệ thống của chúng ta. Chúng ta không nên thay đổi các tệp trong thư mục này.

Thư mục `/usr/spool`, nay là `/var/spool`, chứa thông tin tạm thời, cần thiết cho trình in ấn **lp**, cho daemon **cron** và cho trình liên lạc **UUCP**. Những tệp nào được "tuôn" (spooled) cho máy in đều nằm tạm trong thư mục `spool` cho đến khi chúng được in xong. Mọi chương trình chờ để được **cron** điều khiển, kể cả các tệp **crontab**, các tệp **batch** và at đang trong thời gian chờ đợi đều tạm trú ở đây.

Thư mục `/usr/lib` chứa tất cả những gì còn lại thuộc về hệ thống Linux chuẩn. Nói chung, thư mục này chứa mớ tệp hỗn độn nằm ở hậu trường Linux. Thư mục `/usr/bin` chứa tệp cấu hình cho máy in và terminal, chứa các trình được gọi bởi trình khác của `/bin` và `/usr/bin`, trình thư tín, **cron** và trình liên lạc UUCP.

Thư mục `/usr`, nay là thư mục `/home` chứa mọi thư mục con được cấp cho user. Quy ước chung là nếu ID đăng nhập là "phan_an" thì home directory sẽ là `/usr/phan_an`.

Cách sắp xếp thư mục như chúng ta vừa xem qua được hình thành xưa kia khi đĩa cứng vừa nhỏ vừa đắt tiền. Giờ đây khi đĩa cứng có dung lượng lớn và rẻ hơn, chúng ta có thêm một cách tổ chức Linux tốt hơn sẽ được bàn đến ở mục sau.

3.2. Các thư mục trong Linux

Cấu trúc cổ điển của UNIX gặp khó khăn khi phải sao lưu các tệp dữ liệu với thư mục `/usr` bị phân mảnh. Nhìn chung một hệ thống đòi hỏi phải có ba cấp sao lưu: bản thân hệ thống cơ bản; dữ liệu của user; và bất kỳ thay đổi nào trong những bảng định nghĩa hệ thống cơ bản.

Chúng ta chỉ nên sao lưu hệ thống cơ bản một lần sau khi có thay đổi trong những bảng điều khiển và những thay đổi này đã được sao lưu trước đó rồi. Dữ liệu của user luôn thay đổi do đó nên sao lưu thường xuyên.

Dưới đây là cấu trúc điển hình của Linux, tuy nhiên cấu trúc trong máy chúng ta có thể khác đôi chút, tùy theo chúng ta đã cài đặt những gói phần mềm nào. Các thư mục `/bin`, `/etc` và `/tmp` có cùng chức năng như ở cấu trúc cổ điển. Các bảng định nghĩa hệ thống được chuyển sang thư mục `/var`, nhờ đó mỗi khi trong thao tác của hệ thống có gì thay đổi, chúng ta chỉ cần sao lưu thư mục này mà thôi. Nét mới là mọi chương trình hệ thống đều chuyển sang thư mục `/sbin`. Tất cả những chương trình chuẩn Linux đều nằm trong `/usr/bin` và thư mục này được kết nối với `/bin`. Để dễ dàng tương thích, mọi thư mục cổ điển đều được duy trì bằng những kết nối tượng trưng. Thư mục `/usr`, giờ đây không còn chứa dữ liệu user, được tổ chức lại để chứa mớ tệp hỗn độn trước kia nằm trong thư mục `/usr/lib`.

/

/etc

 /passwd (user database)

 /rc.d (system initialization scripts)

/sbin

/bin

/tmp

/var

/lib

/home

/<tên user của chúng ta ở đây> (trường khoản của user)

/install
 /usr
 /bin
 /sbin
 /src
 /local
 /bin
 /sbin
 /lib
 /proc

B. CÂU HỎI VÀ BÀI TẬP

Câu 1: Trình bày các thao tác cơ bản với tệp?

Câu 2: Nêu cách thực hiện nén và nối tệp trong Linux?

Câu 3: Trình bày hệ thống thư mục trong Linux?

Câu 4: Thực hành tổ chức hệ thống tập tin trên Linux?

Hướng dẫn thực hành:

/etc: Cấu hình hệ thống cục bộ theo máy

/usr/bin: Chứa hầu hết các lệnh người dùng.

/dev: Các tập tin thiết bị.

/usr/man: Chứa các tài liệu trực tuyến.

/usr/include: Chứa các tập tin include chuẩn của C.

/var/log: Các tập tin lưu giữ thông tin làm việc hiện hành của người dùng.

/home: Chứa các thư mục con của các user.

/usr/lib: Chứa các tập tin thư viện của các chương trình người dùng.

Khi truy cập vào hệ thống, thư mục làm việc của người dùng được xem như là thư mục chủ. Ví dụ : Thư mục chủ của user01 sẽ là /home/user01

Nếu đường dẫn bắt đầu bằng dấu “/”, hệ thống xem đó như là một tên đường dẫn đầy đủ bắt đầu từ thư mục gốc.

Câu 5: Thực hành các lệnh thao tác trên hệ thống tập tin:

1. Tạo mới thư mục
2. Thay đổi thư mục hiện hành
3. Xem thư mục làm việc hiện hành
4. Xem thông tin về tập tin và thư mục
5. Di chuyển một hay nhiều tập tin
6. Sao chép tập tin
7. Tạo liên kết với tập tin
8. Xoá thư mục rỗng
9. Hiển thị nội dung của các tập tin
10. Xuất nội dung thông báo
11. Nén và giải nén tập tin

Hướng dẫn thực hành:**1. Tạo mới thư mục****Cú pháp :** `mkdir <dir1> <dir2> ... <dirN>`

<dir1> . . . <dirN> là tên các thư mục cần tạo.

`[user01@linux user01]$ mkdir baitap``[user01@linux user01]$ mkdir document [user01@linux user01]$ mkdir baitap\ltc``[user01@linux user01]$ ls``[user01@linux user01]$ mkdir baitap/ltc [user01@linux user01]$ mkdir baitap/perl`**2. Thay đổi thư mục hiện hành****Cú pháp :** `cd <directory>`

<directory> là thư mục muốn chuyển đến.

.: yêu cầu chuyển đến thư mục hiện hành.

.. : chuyển đến thư mục cha. `[user01@linux user01]$ cd baitap [user01@linux user01]$ cd /home``[user01@linux user01]$ cd`**3. Xem thư mục làm việc hiện hành****Cú pháp :** `pwd``[user12@linux user12]$ pwd``/home/user12 [user12@linux user12]$`**4. Xem thông tin về tập tin và thư mục****Cú pháp :** `ls <file1> <file2> ... <fileN> <Tham số>`

<file1> . . . <fileN> là danh sách tên tập tin hay thư mục.

<Tham số> :

-F : dùng để hiển thị một vài thông tin về kiểu của tập tin**-l** : (long) liệt kê kích thước tập tin, người tạo ra, các quyền người sử dụng.`[user12@linux user12]$ ls -lF`

total 75

drwxrwxr-x	2	user12	user12	1024	Apr 7 09:41	baitap/
drwxrwxr-x	2	user12	user12	1024	Apr 7 09:41	doc/
-rwxrwxr-x	1	user12	user12	71	Mar 31 10:39	hello*
-rw-rw-r--	1	user12	user12	126	Apr 7 09:26	baitho.txt
-rw-rw-r--	1	user12	user12	70	Apr 7 08:26	hello.c

`[user12@linux user12]$`

ls -lF

ls *a* : hiển thị tất cả tập tin hay thư mục con có kí tự a
ls F*E : hiển thị danh sách bắt đầu bằng F và kết thúc bằng E

5. Di chuyển một hay nhiều tập tin

Cú pháp : `mv <file1> <file2> ... <fileN> <destination>`
 <file1> ... <fileN> là danh sách tên tập tin cần di chuyển

<destination> là tập tin hay thư mục đích.

Lệnh mv có thể dùng để đổi tên tập tin.

- Chuyển nhiều tập tin

```
$ mv * directory
```

- Di chuyển thư mục [user01@linux user01]\$ mkdir ctrinh [user01@linux user01]\$ ls -lF

```
[user01@linux user01]$ mv ctrinh baitap
```

Di chuyển thư mục /home/user01/ctrinh vào thư mục /home/user01/baitap

6. Sao chép tập tin

Cú pháp : `cp <source> <destination>`

```
[user01@linux user01]$ cd baitap [user01@linux baitap]$ vi tho.txt [user01@linux baitap]$ mv tho.txt baitho.doc [user01@linux baitap]$ ls
```

```
baitho.doc ctrinh hello.c ltc perl [user01@linux baitap]$ cp baitho.doc ~/document
```

- Sao chép tất cả các tập tin vào một danh mục

```
$ cp * directory
```

7. Tạo liên kết với tập tin

Tạo liên kết với tập tin là tạo thêm cho tập tin tên mới và đường dẫn tương ứng.

Cú pháp : `ln <source> <destination>`

ls -l : xem số liên kết của tập tin.

Muốn xóa một tập tin ta phải xóa tất cả các liên kết của nó.

```
[user01@linux user01]$ pwd [user01@linux user01]$ ls -l [user01@linux user01]$ ls -l baitap
```

```
[user01@linux user01]$ ln baitap/file1 file.link [user01@linux user01]$ ls -l baitap [user01@linux user01]$ ls -l file.link
```

8. Xóa thư mục rỗng

Cú pháp : `rmdir <dir1> <dir2> ... <dirN>`

<dir1> ... <dirN> là tên những thư mục cần xóa.

```
rmdir /home/baitapxóa thư mục /home/baitap
```

9. Hiển thị nội dung của các tập tin

Cú pháp : `more <file1> <file2> ... <fileN>`

`<file1> <file2> ... <fileN>` là những tập tin cần hiển thị.

```
[user12@linux user12]$more baitho.txt // hiển thị tập tin baitho.txt
```

```
[user12@linux user12]$more mbox // Xem tất cả thư lưu trong hộp thư
```

10. Xuất nội dung thông báo

Cú pháp : `echo <arg1> <arg2> ... <argN>`

Trong đó `<arg1> <arg2> ... <argN>` là các đối số dòng lệnh.

```
[user12@linux user12]$ echo "Chao cac ban" Chao cac ban sinh vien"
```

```
[user12@linux user12]$echo * → Hiển thị nội dung thư mục
```

11. Nén và giải nén tập tin

Cú pháp : `gzip <filename>`

Nén một tập tin. Tên tập tin nén giống như tên ban đầu, kèm theo đuôi `.gz`

```
[user12@linux user12]$ gzip vanban.txt -> vanban.txt.gz
```

Cú pháp : `gunzip <filename> gzip -d <filename>`

Lệnh dùng để giải nén tập tin.

```
[user12@linux user12]$gunzip vanban.txt.gz
```

CÂU HỎI ÔN TẬP TRẮC NGHIỆM

Câu 1: Phần quan trọng nhất của hệ điều hành Linux là:

- File System
- Services
- Kernel
- Shell

Câu 2: Số phiên bản kernel của Linux có gì đặc biệt

- Số chẵn là phiên bản Ổn định
- Số lẻ là phiên bản thử nghiệm
- Không quan trọng
- Câu a và b đúng

Câu 3: Khi cài RedHat Linux ở chế độ nào thì các partion DOS bị xóa hết

- Workstation
- Server
- Workstation and Server
- Không cái nào đúng

Câu 4: Kiến trúc Kernel Linux là :

- Static
- Microkernel
- Distributed
- Monolithic

Câu 5: Tác giả của phiên bản hệ điều hành Linux đầu tiên là ?

- a. Bill Gates
- b. Linus Tolvarids
- c. Alan Turing
- d. Pascal

Câu 6: Để liệt kê các file có trong thư mục hiện hành ta dùng lệnh:

- a. lệnh ls
- b. lệnh df
- c. lệnh du
- d. lệnh cp

Câu 7: Để liệt kê đầy đủ thông tin của các file có trong thư mục hiện hành theo chúng ta dùng lệnh ls với tham số

- a. -a
- b. -l
- c. -x
- d. -n

Câu 8. Để liệt kê các file ẩn trong thư mục hiện hành theo chúng ta dùng lệnh ls với tham số

- a. -b
- b. -a
- c. -l
- d. -n

Câu 9: Để chuyển sang một thư mục khác ta dùng lệnh

- a. Lệnh cdir
- b. Lệnh cd
- c. Lệnh mkdir
- d. Lệnh dir

Câu 10: Một user có username là sinhvien và home directory của anh ta là /home/sinhvien. Để trở về home dir của anh ta 1 cách nhanh nhất, anh ta phải dùng lệnh

- a. cd
- b. cd /home/sinhvien
- c. cd / ; cd home ; cd sinhvien
- d. cd /home/sinhvien

Câu 11. Để chép một file /tmp/hello.txt vào thư mục /tmp/hello/ ta phải làm lệnh nào sau đây:

- a. copy /tmp/hello.txt /tmp/hello/
- b. cp tmp/hello.txt /tmp/hello
- c. cp /tmp/hello /tmp/hello
- d. cp /tmp/hello.txt /tmp/hello

Câu 12: Để xem nội dung một tập tin văn bản trong Linux ta có thể dùng lệnh nào sau đây

- a. cat
- b. less
- c. more
- d. cả 3 lệnh trên

Câu 13: Để đọc nội dung một đĩa CD trong Linux ta phải làm thế nào ?

- a. Phải mount trước
- b. eject cdrom
- c. cd /mnt/cdrom
- d. Không đọc được CD rom

Câu 14: Làm thế nào để đọc một đĩa mềm trong Linux

- a. mount /dev/fd0
- b. mount /dev/cdrom
- c. Không cần làm gì cả, chỉ việc đọc từ ổ a:
- d. cd /mnt/floppy

Câu 15: Tập tin nào chứa các mount point mặc định khi hệ thống boot lên:

- a. /etc/mstab
- b. /etc/mount.conf
- c. /etc/fstab
- d. /etc/modules.conf

Câu 16: Tập tin nào chứa thông tin các file system đang được mount

- a. /etc/mntab
- b. /etc/mount.conf
- c. /etc/fstab
- d. /etc/modules.conf

Câu 17: Tập tin sau đây có thuộc tính như thế nào :

-rwx--x--x hello.sh

- a. 077
- b. 644
- c. 755
- d. 711

Câu 18: Chủ sở hữu được quyền đọc ghi, nhóm được đọc, thực thi, other chỉ đọc thi ta làm lệnh nào dưới đây.

- a. chmod 665
- b. chmod 654
- c. chmod 653
- d. chmod 754

Câu 19: Tập tin có dấu chấm "." Phía trước có đặc tính gì đặc biệt:

Ví dụ: .hello.txt

- a. Tập tin ẩn
- b. Thực thi
- c. Không thấy được với lệnh ls
- d. Tập tin hệ thống

Câu 20. Lệnh nào cho phép ta tạo một account user mới trên hệ thống

- a. Lệnh adduser
- b. Lệnh useradd
- c. lệnh passwd
- d. Câu a và b đúng

Câu 21. Sau khi đánh lệnh useradd sinhvien. Hỏi user có account là sinhvien có thể login vào hệ thống được chưa ?

- a. Được
- b. Không

Câu 22. Tập tin /etc/passwd chứa thông tin gì của users hệ thống

- a. Chứa profile của người dùng
- b. Chứa uid,gid, home directory, shell
- c. Chứa password của người dùng
- d. Chứa tập shadow của người dùng

Câu 23. Tập tin /etc/shadow chứa thông tin gì của users hệ thống

- a. Chứa profile của người dùng
- b. Chứa uid,gid, home directory, shell
- c. Chứa password của người dùng
- d. Chứa login name

Câu 24. Trong hệ thống Linux user nào có quyền cao nhất

- a. User administrator
- b. User root
- c. User admin
- d. User có UID=0

Câu 25. Hệ thống Linux có mấy Run Level chính

- a. Có 7 Run Level
- b. Có 6 Run Level
- c. Có 5 Run Level
- d. Có 4 Run Level

Câu 26. Run level nào là shutdown và halt hệ thống

- a. Level 6
- b. Level 0
- c. Level 3
- d. Level 5

Câu 27. Run level nào là reboot hệ thống

- a. Level 6
- b. Level 0

- c. Level 3
- d. Level 5

Câu 28. Run level nào làm hệ thống chạy full mode with X window

- a. Level 6
- b. Level 0
- c. Level 3
- d. Level 5

Câu 29. Ở run level nào hệ thống không đòi hỏi ta phải nhập username password để

login

- a. Level 1
- b. Level 3
- c. Level 5
- d. Không có level nào

Câu 30. Chương trình soạn thảo văn bản nào là phổ biến nhất trong hđh Linux:

- a. vi
- b. Word
- c. Emacs
- d. Pico

Câu 31. Chương trình soạn thảo nào sau đây không sử dụng giao diện đồ họa

- a. emacs
- b. latex
- c. vi
- d. kword

Câu 32. Emacs là một chương trình

- a. Soạn thảo văn bản
- b. Công cụ lập trình
- c. Email client
- d. Tất cả các tính năng trên

Câu 33. Trong vi, để vào mode edit insert ta phải dùng lệnh nào :

- a. Dùng phím F4
- b. ESC-:i
- c. ESC-:q
- d. ESC-:x

Câu 34. Trong vi, để thoát không cần hỏi lại ta phải dùng lệnh nào :

- a. ESC-:q!
- b. ESC-wq
- c. ESC-w!
- d. ESC-!

Câu 35. Để xem các tiến trình hiện có trong hệ thống Linux ta dùng lệnh nào

- a. Lệnh ls
- b. Lệnh ps
- c. Lệnh cs
- d. Lệnh ds

A: b

Câu 36. Để xem chi tiết các tiến trình đang chạy trong hệ thống ta dùng lệnh ps với các tham số nào sau đây

- a. -ef
- b. -ax
- c. -axf
- d. Cả 3 câu đều đúng

Câu 37. Mỗi tiến trình chạy trong hệ thống Linux được đặc trưng bởi :

- a. PID
- b. PUID
- c. PGUID
- d. GUID

Câu 38. Tham số PPIUD dùng để chỉ:

- a. Properly process ID
- b. Parent process ID
- c. Papa Process ID
- d. Không cái nào đúng

Câu 39. Để dừng một tiến trình ta dùng lệnh nào :

- a. stop
- b. kill
- c. shutdown
- d. halt

A: b

Câu 40. Điều gì xảy ra với một tiến trình khi ta kill tiến trình cha của nó?

- a. Không có gì ảnh hưởng
- b. Tiến trình con sẽ chết theo
- c. Chương trình sẽ đọc lại file cấu hình
- d. Sẽ có một tiến trình con mới sinh ra

Câu 41. Để kill hết các tiến trình có tên là vi ta dùng lệnh nào?

- a. kill -9 vi
- b. kill -HUP vi
- c. killall -HUP vi
- d. killall -9 vi

Câu 42. Để thiết lập địa chỉ IP cho một máy Linux ta sử dụng lệnh nào trong các lệnh sau đây?

- a. ipconfig
- b. ifconfig
- c. netstat
- d. route

Câu 43. Để xem trạng thái các port đang mở của một máy Linux ta sử dụng lệnh nào trong các lệnh sau đây?

- a. ipconfig
- b. ifconfig
- c. netstat
- d. route

Câu 44. Để xem các thông tin về bảng routing trong hệ thống Linux ta sử dụng lệnh nào trong các lệnh sau đây?

- a. ifconfig
- b. netstat -nr
- c. route -n
- d. b và c đúng

Câu 45. Để đặt địa chỉ IP cho card mạng eth0 dùng lệnh ifconfig, ta phải thực hiện lệnh nào sau đây

- a. ifconfig eth0 172.16.10.11/ 255.255.255.0
- b. ifconfig eth0 172.16.10.11 netmask 255.255.255.0
- c. ifconfig eth0 172.16.10.11 mask 255.255.255.0
- d. ifconfig eth0 172.16.10.11 mask 255.255.255.0

Câu 46. Để tạm thời stop một card mạng ta dùng lệnh nào?

- a. ifconfig eth0 up
- b. ifconfig eth0 stop
- c. ifconfig eth0 start
- d. ifconfig eth0 down

Câu 47. Giả sử ta muốn thêm vào bảng routing một con đường mới: qua mạng

192.168.10.0/24 thì phải qua gateway 172.16.10.140 ta làm cách nào:

- a. route add -net 192.168.10.0 netmask 255.255.255.0 gw 172.16.10.140
- b. route add -net 192.168.10.0 mask 255.255.255.0 gw 172.16.10.140 netmask 255.255.255.0
- c. route add -net 192.168.10.0/25 gw 172.16.10.140
- d. route add -net 192.168.10.0 netmask 255.255.255.0 gw 172.16.10.0

A: a

Câu 48. Khai báo default gw 172.16.8.2 cho 1 máy Linux sử dụng làm lệnh nào:

- a. route add -net default gw 172.16.8.2
- b. route add -net 0.0.0.0 netmask 0.0.0.0 gw 172.16.8.2
- c. route add -net 0.0.0.0 netmask 255.255.255.255 gw 172.16.8.2

d. Câu a và b đúng

Câu 49. Để xem tải của hệ thống Linux ta dùng lệnh nào?

- a. Lệnh top
- b. Lệnh free
- c. Lệnh performance
- d. Không thể biết

Câu 50. Tập tin nào trong Linux định nghĩa các port cho các dịch vụ chạy trong nó?

- a. /etc/service
- b. /etc/services
- c. /etc/ports.conf
- d. /etc/httpd.conf

ĐÁP ÁN:

Câu 1: c

Câu 2: d

Câu 3: b

Câu 4: d

Câu 5: b

Câu 6: a

Câu 7: b

Câu 8: b

Câu 9: b

Câu 10: b

Câu 11: d

Câu 12: d

Câu 13: a

Câu 14: c

Câu 15: c

Câu 16: b

Câu 17: d

Câu 18: b

Câu 19: a

Câu 20: d

Câu 21: b

Câu 22: b

Câu 23: c

Câu 24: b

Câu 25: a

Câu 26: b

Câu 27: a

Câu 28: d

Câu 29: a

Câu 30: a

Câu 31: c

Câu 32: d

Câu 33: b

Câu 34: a

Câu 35: b

Câu 36: b

Câu 37: a

Câu 38: d

Câu 39: b

Câu 40: b

Câu 41: b

Câu 42: b

Câu 43: c

Câu 44: d

Câu 45: b

Câu 46: b

Câu 47: a

Câu 48: d

Câu 49: a

Câu 50: b

PHỤ LỤC

Nội dung tệp dosemu.conf và global.conf điển hình

DOSEMU.CONF

```
#####
```

```
# This file is /etc/dosemu.conf. included by <DOSEMU-LIB-DIR>/global.conf
```

```
#
```

```
# Linux DOSEMU configuration for Parser versions - 3 (dosemu-0.97.0.1)
```

```
#
```

```

# ./doc/README.txt (chapter 2.) contains a description of the syntax
# and the usage of dosemu.conf.
#
# Access rights are defined in
#
#   /etc/dosemu.users
#
#####
#
# Notes for editing this section:
#   In   $_xxx = (n) n is a numerical or boolean value
#           =   =
#   In   $-zzz = "s" s is a string
#
# Please edit only between the brackets and quotes and keep the rest intact.
#   ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
$_debug - "-a" # same format as -D commandline option
           # (but without the -D in front)
$_features= "" # list of temporary hacks, see release notes in
           # the file ChangeLog. e.g "0:1 2:0", which means
           # to set feature - 0 to 1 and feature-2 to 0.
$_mapping= "" # preferred mapping driver, one of:
           # auto, mapself, mapfile, mapshm
$_timint = (on) # emulate INT08 type timer interrupts
$_mthco = (on) # or off
$_cpu = "80386" # CPU showing to DOS, valid values: "80[345]86"
           # or "emulated" for non-native CPU (386 in this case)
$rdtsc = (on) # if possible use Pentium cycle counter
$_cpuspeed = (0)# 0 = calibrated by dosemu, else given (e.g.166.666)
$_pci = (off)
$_xms = (1024) # in Kbyte
$_ems = (2048) # in Kbyte
$_ems frame = (0xe000)
-   7
$_dpmi = (off) # in Kbyte
$_dosmem = (640) # in Kbyte, < 640
$_hardware_ram = "" # list of segment values/ranges such as
           # "0xc8000 range 0xcc000.0xcffff"
$-secure = "ngd" # secure for: n (normal users), g (guest), d (dexe)
           # empty string: depending on 'restricted'
           # "0": always insecure (not recommended)

```

```

$_odd_hosts = -4 black list such as "lucifer.hell.com billy.the.cat"
$_diskless_hosts="" # black list such as 'hacker1 newbie gateway1"
$_emusys =# empty or 3 char., config.sys - config.XXX
$_emubat = # empty or 3 char., autoexec.bat- autoexec.XXX
$_emuini = # empty or 3 char.. system.ini - system.XXX
$_hogthreshold = (1) # 0 == all CPU power to DOSEMU
$_irqpassing = "" # list of IRQ number (2-15) to pass to DOS such as
    # "3 8 10"
$_speaker = "" # or "native" or "emulated"
$_term_char_set "" # Global code page and character set selection.
    # automatic, else: ibm. latin, latin1, latin2
$_ - term - color (on) # terminal with color support
$_ - term_updfreq = (4) # time between refreshs (units: 20 == 1 second)
$_escchar = (30)# 30 == Ctrl-^, special -sequence prefix
$_rawkeyboard = (0) # bypass normal keyboard input, maybe dangerous
$_layout = "auto" # one of: finnish(-latin1). de(-latin1). be. it. us
    # uk. dk(-Iatin1), keyb-no, no-latin1. dvorak, po
    # sg(-latin1), fr(-latin1), sf(-latin1), es(-latin1)
    # sw, hu(-Iatin2), hu-cwi. keyb-user
    # hr-cp852, hr-latin2. cz-qwerty. cz-qwertz.
    # Or 'auto' (which tries to generate the table from
    # the current Linux console settings)
$_keybint = (on)# emulate PCish keyboard interrupt
$_X_updfreq = (5) # time between refreshs (units: 20 == 1 second)
$_X_title = "DOS in a BOX"# Title in the top bar of the window
$_X_icon_name = "xdos" # Text for icon, when minimized
$_X_keycode = (auto) # on == translate keyboard via dosemu keytables
    # or 'off' or 'auto'
$_X_blinkrate = (8) # blink rate for the cursor
$_X_font = "" # basename from /usr/X11R6/lib/X11/fonts/misc/*
    # (without extension) e.g. "vga"
$_X_mitshm = (on) # Use shared memory extensions
$_X_sharecmap = (off) # share the colormap with other applications
$_X_fixed-aspect = (on)# Set fixed aspect for resize the graphics window
$_X_aspect_43 = (on) # Always use an aspect ratio of 4:3 for graphics
$_X_lin_filt = (off) # Use linear filtering for >15 bpp interpolation
$_X_bilin_filt = (off) # Use bi-linear filtering for >15 bpp interpolation
$_X_model3fact = (2) # initial size factor for video mode 0x13 (320x200)
$_X_winsize = ""# "x,y" of initial windows size (defaults to float)
$_X_gamma = (1.0) # gamma correction
    $_X_vgaemu_memsize = (1024)

```

```

# size (in Kbytes) of the frame buffer for emulated vga
$_X_lfb = (on) # use linear frame buffer in VESA modes
$_X_pm_interface = (on)# use protected mode interface for VESA modes
$_X_mgrab - key "Home" # KeySym name to activate mouse grab. empty == off
$_X_vesamode "" # "xres,yres ... xres,yres"
    # List of vesamodes to add. The list has to contain
    # SPACE separated "xres.yres" pairs
$_video = vga"    # one of: plainvga. vga, ega. mda, mga. cga
$-console (0)    # use 'console' video
$_graphics (0)# use the cards BIOS to set graphics
$_videoportaccess = (1)# allow videoportaccess when 'graphics' enabled
$_vbios_seg = (Oxc000) # set the address of your VBIOS (e.g. 0xe000)
$_vbios_size = (OxIO000)# set the size of your BIOS (e.g. MON)
$_vmemsize (1024) # size of regen buffer
$_chipset = ""    # one of: plainvga, trident, et4000, diamond, avance
    # cirrus, matrox. wdvga, paradise. ati, s3, sis
$_dualmon = (0)  # if you have one vga_plus_one hgc (2 monitors)
$_vbootfloppy = "" # if you want to boot from a virtual floppy:
    # file name of the floppy image under DOSEMU_LIB_DIR
    # e.g. "floppyimage" disables $_hdimage
    # "floppyimage +hd" does - not _disable $_hdimage
$_floppy_a = "threeinch"# or "fiveinch" or "atapi" or empty, if not existing
    # optionally the device may be appended such as
    # "threeinch:/dev/fd0"
$_floppy_b = ""    # dito for B:
$_hdimage = "freedos" # list of hdimages under DOSEMU_LIB_DIR
    # assigned in this order such as
    # "hdimage_c hdimage_d hdimage_e"
    # If the name begins with '/dev/', then partion
    # access is done instead of virtual hdimage such as
    # "/dev/hdal" or "/dev/hdal:ro" for readonly
    # Currently mounted devices and swap are refused.
    # Hdimages and devices may be mixed such as
    # "hdimage_c /dev/hdal /dev/hda3:ro"
    # Note: 'wholedisk' is - not_ supported.
$_hdimage_r = $_hdimage# hdimages for 'restricted access (if different)
$_aspi = " " # list of generic SCSI devices to make available
# for the builtin aspi driver (format of an entry
# is 'device:type:mappedtarget' such as
# "sg2:WORM sg3:Sequential_Access:6 sg4:CD-ROM" or
# "sg2:4 sg3:1:6 sg4:5" (which are equal)

```

```

$_com1 = " "      # e.g. "/dev/mouse" or  "/dev/ttySO"
$_com2 = " "      # e.g. "dev/modem" or  "/dev/ttyS1"
$_com3 = " "      # dito "/dev/ttyS2"
$_com4 = " "      # dito "/dev/ttyS3"
$_ttylocks = " "      # Lock directory (e.g. "/var/lock")
# default (" ") is /usr/spool/uucp
$-mouse = " "      # one of: microsoft, mousesystems, logitech, mmseries
                # mouseman, hitachi, busmouse, ps2, jmps2
$_mouse_dev = " " # one of: com1, com2, com3, com4 or /dev/mouse
$_mouse_flags = " " # list of none or one or more of:
                # "emulate3buttons clearltr"
$_mouse_baud = (0) # baudrate, 0 == don't set
$_printer = "lp"# list of /etc/printcap) printer names to appear as
                # LPT1, LPT2, LPT3 (not all are needed, empty for none)
$_printer_timeout = (20)# idle time in seconds before spooling out
$_ports = " " # list of portnumbers such as "Ox1ce Ox1cf Ox238"
                # or "Ox1ce range Ox280,Ox29f 310"
                # or "range Ox1a0,(Ox1a0+15)"
$_ipxsupport = (off) # or on
$_novell_hack = (off)
$_vnet = (off) # 'on' for packet-multi (used by dosnet)
$_sound = (off) # sound support on/off
$_sb_base = (Ox220)
$_sb_irq = (5)
$_sb_dma = (1)
$_sb_dsp = "/dev/dsp"
$_sb_mixer = "/dev/mixer"
$_mpu_base = "Ox330"
GLOBAL.CONF
#####
# This file is global.conf, it is -always_ executed as part of the initial
# configuration.
#
# Linux dosemu configuration for parser versions - 3 (dosemu-0.97.0.1)
#
# ./doc/README-tech.txt (chapter 2.) contains a description of the syntax
# and the usage. However, you normal won't edit this file
#
# NOTES:
#
# 1. The file dosemu.conf (and optionally -/.dosemurc) contains variable

```

```

# settings, that are included by global.conf for doing the
# most reasonable configuration.
# The file dosemu.conf (and optionally ~/.dosemurc) is what gets
# updated by the menu driven 'setup-dosemu' tool.
#
# 2. We strongly recommend you to edit ONLY dosemu.conf.
# If you change global.conf, you are at your own and could break
# 'setup-dosemu'. You really need to know a lot of DOSEMU
# internals before you fiddle with editing global.conf.
# However, some very special cases can only be handled in global.conf.
#
# 3. The file global.conf (this one) must either
#
# - be in DOSEMU LIB DIR (the default, changeable via dosemu.users
# or --Flibdir)
# - or given via -F option (from root login, else only on
# non-suid-root DOSEMU)
# - or be in $HOME/,dosemu/lib.
#
# For the latter there must be given the following conditions:
# - in dosemu.users give the keyword 'private_setup' for this user
# - DOSEMU -must- be non-suid-root. except for user root or if you
# - give the keyword 'unrestricted' in addition to 'private-setup'.
#
# 4. The only compiled-in path is /etc/dosemu.users (or if not found
# /etc/dosemu/dosemu.users). however, this can be overridden by
# --Fusers, if not running suid-root. All other pathes are configurable
# and the dosemu binaries can reside everywhere in the system.
#
# This file (global.conf) may also serve as an example file for
# dosrc ( old style user configuration file )
# option -I ( configuration via commandline, see man/dos.)
#
# Access rights for suid-root running DOSEMU sessions are defined in
#
# /etc/dosemu.users
#
#####
ifdef u_forceold
undef parser_version_3
define version_3_style_used

```



```

endif
ifndef parser_version_3
    # normally won't come here, because older DOSEMUs don't use this file
    # ... but if using -F option
    include "/etc/dosemu.conf"
else
    # we are on version 3 parser
    # for security reason we restrict the PATH and restore it later
    # to its original value
    $_DOSEMU_ORIG_PATH = $PATH
    $PATH = "/usr/local/bin:/usr/bin:/bin"
    ## we set some vital variable
    if (!strlen($DOSEMU_LIB_DIR))
        # be sure, because older DOSEMU won't set this
        $DOSEMU_LIB_DIR = "/var/lib/dosemu";
        $DOSEMU_CONF_DIR = "/etc";
        $DOSEMU_HDIMAGE_DIR = $DOSEMU_LIB_DIR;
    else
        if (!strlen($DOSEMU_HDIMAGE_DIR))
            $DOSEMU_HDIMAGE_DIR = $DOSEMU_LIB_DIR;
        endif
        # make sure we have absolute paths
        $DOSEMU_LIB_DIR = shell("cd ", $DOSEMU_LIB_DIR, " && pwd -P");
        $DOSEMU_LIB_DIR = strdel($DOSEMU_LIB_DIR,
            strlen($DOSEMU_LIB_DIR)-1, 1);
        $DOSEMU_HDIMAGE_DIR = shell("cd ", $DOSEMU_HDIMAGE_DIR, " && pwd
            -P");
        $DOSEMU_HDIMAGE_DIR = strdel($DOSEMU_HDIMAGE_DIR,
            strlen($DOSEMU_HDIMAGE_DIR)-1, 1);
    endif
    $CONFIG_VERSION = (( 0 << 24) | (97 << 16) | (2 << 8) | 0)
    #           ^      ^^      ^      ^
    if ( $DOSEMU_VERSION_CODE < $CONFIG_VERSION
        abort "
        ***sorry. your ", $DOSEMU_LIB_DIR, "/global.conf doesn't match this dosemu
        version"
    endif
    $DOSEMU_VERSION = ($DOSEMU_VERSION_CODE >> 24), '.',
    (($DOSEMU_VERSION_CODE >> 16) & 255), '.',
    (($DOSEMU_VERSION_CODE >> 8) & 255), '.',
    ($DOSEMU_VERSION_CODE & 255);

```

```

$LIST_DELIM = "," #delimiters for lists <tab>, blank, comma
# we include the global settings from /etc/dosemu.conf shell("test -f ",
$DOSEMU_CONF_DIR, "/dosemu.conf")
if ( $DOSEMU_SHELL_RETURN)
abort "
***error: ", $DOSEMU_CONF_DIR, "/dosemu.conf not accessible ... giving up.
"
endif
    undef version_3_style_used # unset it to have a valid check below
    $xxx = 'include "', $DOSEMU_CONF_DIR, '/dosemu.conf"';
    $$xxx;
    ifndef version_3_style_used
        abort "
***Your ", $DOSEMU_CONF_DIR, "/dosemu.conf is obviously an old style or a
too simple one.
        Please read Quickstart and README.txt on how to upgrade.
"
endif
## we check if we have an ${HOME}/.dosemurc (given by $DOSEMU_RC),
## if yes, we include it here 'under user scope', which means it has
## all c_* classes undefined and can only change environment variables in
## its own isolated name space (prefixed with 'dosemu_').
## We later take over only such variables, that we allow.
$_dosemurc = $DOSEMU_RC # we get that passed from 'parse-config()'
shell("test -f ", $_dosemurc)
if ( (!$DOSEMU_SHELL_RETURN) && ($DOSEMU_VERSION_CODE >=
((97<<16) | (4<<8) | 3)))
    # we have a .dosemurc and can include
    $_dosemurc = 'include " ', $_dosemurc, ' " ' ;
    enter_user_scope;
    $$$_dosemurc;
    leave_user_scope;
define skip_dosrc:
## now we take over the allowed variables, only those which exist
## with 'dosemu_' prefixed will overwrite those without the prefix.
checkuservar $_debug,
    $_features, $_mapping,
    $_timint, $_mathco, $_cpu, $_pci, $_xms, $_ems, $_ems_frame,
    $_emusys, $_emubat. $_emuini
checkuservar
    $_term_char_set, $_term_color. $_term_updfreq, $_escchar, $_layout,

```

```

    $_keybint
checkuservar
    $_X_updfreq, $_X_title, $_X_icon_name, $_X_keycode, $_X_blinkrate,
    $_X_font, $_X_mitshm, $_X_sharecmap, $_X_fixed_aspect, $_X_aspect_43,
    $_X_lin_filt, $_X_bilin_filt, $_X_mode13fact. $_X_winsize, $_X_gamma,
    $_X_vgaemu_memsize,  $_X_lfb,  $_X_pm_interface,  $_X_mgrab_key,
$_X_vesamode
ifdef guest
define restricted
    else
        checkuservar
        $_speaker,
        $_com1,  $_com2.  $_com3.  $_com4.  $_mouse,  $_mouse_dev.
$_mouse_flags,
$_mouse_baud,
        $_printer, $_printer_timeout, $_ports,
        $_ipxsupport. $_novell_hack, $_vnet,
        $_sound, $_sb_base, $_sb_irq, $_sb_dma, $_sb_dsp, $_sb_mixer.
$_mpu_base
    endif
ifndef restricted
checkuservar $_hogthreshold, $_dpmi, $_vbootfloppy, $_aspi, $_ttylocks
$xxx = $ hdimage # remember the contents for later check
checkuservar $_hdimage
if ($xxx ne $_hdimage)
    # the user changed it in .dosemurc, have to check
    if (strlen($_hdimage) && ($DOSEMU_UID != $DOSEMU_EUID))
        # we are running suid-root as user, don't allow partition access
        $yyy = $_hdimage
        foreach $xxx ($LIST_DELIM, $yyy)
            if (!strncmp($xxx, "/dev/", 4))
                abort "
***sorry, your are not allowed to use ", $xxx, "in dosemurc
“
endif
done
endif
endif
endif
endif
endif
## end of .dosemurc inclusion

```

```

if ( $DOSEMU_VERSION_CODE >= ( (99<<16) | (13<<8) | 2) )
    if (strlen($_mapping))
        mappingdriver $_mapping;
    endif
endif
if (strlen($_debug))
    debug $_debug ;
else
    debug {off}
endif
if ($_cpuspeed) cpuspeed $_cpuspeed endif
if ($_rdtsc) rdtsc on else rdtsc off endif
pci $_pci
dosbanner on
timint $_timint
mathco $_mathco
if (strlen($_cpu))
    $_cpu = "cpu ", $_cpu;
    $$_cpu
else
    cpu 80386
endif
xms $_xms
if ($_ems) ems {ems_size $_ems ems_frame $_ems_frame} else ems off endif
dosmem $_dosmem
if ($_emusys ne " ")    emusys $_emusys endif
if ($_emubat ne " ")    emubat $_emubat endif
if ($_emuini ne " ")    emuini $_emuini endif
$DISABLE_MOUSE = (0)
$DISABLE_SERIAL = (0)
## terminal stuff, we check a lot to insure proper operation
$BEING_ON = " "
$USING_X = (0)
$REMOTE_HOST = " "
## Note GNU sh-utils>=2.0 changed the default behaviour such
# that it doesn't display the peer for remote connections anymore.
# We need -1 option to get this. As GNU sh-utils 1.16 doesn't
# have this option, we detect this case via shell return value.
$xxx = shell("who -1 am i 2>/dev/null")
if ($DOSEMU_SHELL_RETURN)
    # we have GNU sh-utils < 2.0

```

```

        $xxx = shell("who am i")
    endif
    if ( ($DOSEMU_STDIN_IS_CONSOLE eq "1")
        || ( strlen($xxx) && (strchr($xxx, "(") < 0 )))
        $BEING_ON = "console"
    else
        if (strstr($xxx, ":") < 0)
            $BEING_ON = "remote"
            $REMOTE_HOST = strsplit($xxx. strchr($xxx, "(")+1,99)
            $REMOTE_HOST = strdel($REMOTE_HOST,
strchr($REMOTE_HOST, "("),99)
        else
            $BEING_ON = " "
        endif
        if ( (strchr($DOSEMU_OPTIONS, "X") >=0) && ($DISPLAY ne " ") )
            $USING_X = (1)
        endif
        if (strstr("xterm dtterm", $TERM) >=0)
            $BEING_ON = $BEING_ON, "_xterm"
        else
            if (strstr("linux console", $TERM) < 0)
                # remote side must be some type of real terminal
            $BEING_ON = $BEING_ON, "_terminal"
            else
                # remote side is a Linux console
            $BEING_ON = $BEING_ON. "_console"
            endif
        endif
    endif
    warn "dosemu running on ", $BEING_ON;
    $xxx = $_term_char_set
    if ($xxx eq "")
        $xxx = "latin"
        if (strstr($BEING_ON, "_console") >=0)
            $xxx = "ibm"
        endif
    endif
    if ( ($BEING_ON ne "console") && (!$USING_X))
        $DISABLE_MOUSE = (1)
        if ($_term_color) $_term_color = "on" else $_term_color "off" endif
        terminal { charset $$xxx updatefreq $_term_updfreq escchar $_escchar

```

```

        color $$_term_color
    endif
    if ( $DOSEMU_VERSION_CODE >= ( (97<<16) | (10<<8) | 5) )
        charset $$xxx
    endif
    ## host checking. Naah, we won't let all in ;- )
    ## first: logged in hosts
    if (strlen($REMOTE_HOST) && strlen($_odd_hosts))
    foreach $xxx ($LIST_DELIM, $_odd_hosts)
    if ( ! strncmp($xxx, $REMOTE_HOST, strlen($REMOTE_HOST)))
    abort "Sorry, your host is on the blacklist"
    endif
        done
    endif
    ## second: hosts on diskless machines, that aren't allowed
    if (strlen($_diskless_hosts))
        foreach $xxx ($LIST_DELIM, $_diskless_hosts)
    if ( ! strncmp($xxx, $DOSEMU_HOST, strlen($DOSEMU_HOST)))
        abort "Sorry, your host is on the blacklist"
    endif
    done
    endif
    ## X param settings
    if ( $USING_X || (strchr($DOSEMU_OPTIONS."L") >0) )
        # running as xdos or DEXE (which may force X later)
        if ($_X_keycode == -1) # we do not test for 'auto', because older
            # DOSEMU don't know it and we want older
            # versions also running with this global.conf
            $_X_keycode = "keycode_auto"
        else
            if ($_X_keycode) $_X_keycode = "keycode" else $_X_keycode = " " endif
        endif
        if ($_X_sharecmap) $_X_sharecmap = "sharecmap" else $_X_sharecmap = "" endif
        if ($_X_aspect_43) $_X_aspect_43 = "aspect_43" else $_X_aspect_43 = " "
        endif
        if ($_X_lin_filt) $_X_lin_filt = "lin_filt" else $_X_lin_filt = " " endif
        if ($_X_bilin_filt) $_X_bilin_filt = "bilin_filt" else $_X_bilin_filt = " " endif
        if ($ X mode13fact != 2)
            $_X_mode13fact = "mode13fact ", $_X_mode13fact
        else
            $_X_mode13fact = " "
        endif
    endif

```

```

endif
_X_gamma = "gamma ",      (int(_X_gamma * 100))
if (strlen(_X_font)) _X_font = 'font " ', _X_font, ' "' endif
if (strlen(_X_winsize))
$Xxx = (strstr(_X_winsize,","))
      _X_winsize = "winsize (", strdel(_X_wlnsize,$xxx,999), "),( "
      strsplit(_X_wjnsize,$xxx+1,999), ")"
endif
$xxx = _X_vesamode
_X_vesamode = ""
if (strlen($xxx))
  foreach $yyy (" ", $xxx)
    $zzz = (strchr($yyy,","))
    _X_vesamode = "vesamode (", strdel($yyy,$zzz.999), "),( "
    strsplit($yyy,$zzz+1,999), ")"
  done
endif
if (strlen(_X_mgrab_key))
  _X_mgrab_key = 'mgrab_key " ', _X_mgrab_key, ' "'
endif
X {
  title _X_title icon_name _X_icon_name
  updatefreq _X_updfreq blinkrate _X_blinkrate
fixed_aspect _X_fixed_aspect vgaemu_memsize _X_vgaemu_memsize
lfb _X_lfb pm_interface _X_pm_jnterface mitshm _X_mitshm
$_x keycode $$_X_sharecmap $$_X_aspect_43 $$_X_Iin_filt $$_X_bilin_filt
  $$_X_font $$_X_winsize $$_X_gamma
  $$_X vesamode $$_X_mode13fact $$_X_mgrab_key
}
endif
ifndef guest
  ## /etc/dosemu.users defined 'guest' for this login
  define restricted # force restricted setting
  $_hogthreshold = (1) # force 'nice'
  $_rawkeyboard = (off) # force normal keyboard
  if ($BEING_ON eq "console") video { vga } endif
  sound emu off
  $DISABLE_MOUSE = (1)
  $DISABLE_SERIAL = (1)
else
  ## other then guest

```

```

# video settings
if ($BEING_ON eq "console")
    if (!strlen($_video)) $_video = "plainvga" endif
    if ($_console) $_console = "console" else $_console = " " endif
    if ($_graphics)
        $_graphics = "graphics"
        $_console = "console"
        $_vbios_seg = "vbios_seg ", ($_vbios_seg)
        $_vbios_size = "vbios_size ", ($_vbios_size)
        $_vmemsize = "memsize ", ($_vmemsize)
    if ($_videoportaccess) allowvideoportaccess on endif
    else
        $_graphics = " "; $_vbios_seg = " "; $_vbios_size = " "; $_vmemsize = " "
    endif
if (strlen($_chipset)) $_chipset = "chipset ",$_chipset endif
if ($_dualmon) $_dualmon = "dualmon" else $_dualmon = " " endif
video {
    $$$_video $$$_console $$$_graphics $$$_chipset
    $$$_vbios_seg $$$_vbios_size $$$_vmemsize $$$_dualmon
}
else
    if ($_dualmon) video { dualmon } endif
endif
## sound settings
if ($_sound)
    sound_emu { sb_base $_sb_base sb_irq $_sb_irq sb_dma $_sb_dma
sb_mixer $_sb_mixer mpu_base $_mpu_base }
else
    sound_emu off
endif
## serial
if (($DISABLE_SERIAL) && strlen($ ttylocks))
ttylocks { directory $_ttylocks }
endif
if (!($DISABLE_SERIAL) && strlen($_com1))
    if ($_mouse_dev eq "com1")
if (!$DISABLE_MOUSE) serial { mouse com 1 device $_Com1 } endif
$_mouse = " "
else
    if ($_com1 eq "virtual")
serial { com 1 virtual }

```



```

else
serial { com 1 device $_com1 }
endif
endif
endif
if (($DISABLE_SERIAL) && strlen($_com2))
if ($_mouse_dev eq "com2")
if (!$DISABLE_MOUSE) serial { mouse com 2 device $_com2 } endif
$_mouse = " "
else
if ($_com2 eq "virtual")
serial { com 2 virtual }
else
serial { com 2 device $_com2 }
endif
endif
endif
if (($DISABLE_SERIAL) && strlen($_com3))
if ($_mouse_dev eq "com3")
if (!$DISABLE_MOUSE) serial { mouse com 3 device
$_com3 } endif
$_mouse = " "
else
if ($_com3 eq "virtual")
serial { com 3 virtual }
else
serial { com 3 device $_com3 }
endif
endif
endif
if (($DISABLE_SERIAL) && strlen($_com4))
if ($_mouse_dev eq "com4")
if (!$DISABLE_MOUSE) serial { mouse com 4 device $_com4 } endif
$_mouse = " "
else
if ($_com4 eq "virtual")
serial { com 4 virtual }
else
serial { com 4 device $_com4 }
endif
endif
endif

```

```

endif
## mouse settings
if      ((!$DISABLE_MOUSE)      &&      strlen($_mouse)      &&
(strstr($_mouse_dev,"com")<O))
    if ($_mouse_baud)
        $_mouse_baud = "baudrate ", $_mouse_baud
    else
        $ mouse baud = " "
    endif
if (strlen($_mouse_dev))
$_mouse_dev = "device ", $_mouse_dev, " internaldriver"
endif
mouse { $$_mouse $$_mouse_dev $$_mouse_flags $$_mouse_baud }
endif
endif # end 'not guest'
hogthreshold $_hogthreshold
## keyboard setting
If ($BEING_ON ne "console") $_rawkeyboard = (off) endif
if ($_layout eq "auto")
    keyboard { layout auto keybint $_keybint rawkeyboard $_rawkeyboard }
else
    if ( strstr($_layout, "load") <0 )
        # we use the builtin keytables
        if (strlen($_layout)) $_layout = "layout " .$_layout
        else $_layout = "layout us" endif
        keyboard    {    $$_layout    keybint    $_keybint    rawkeyboard
$_rawkeyboard }
    else
        # we have to load a keytable
        $yyy = " "
        foreach $xxx ($LIST_DELIM, $_layout)
if ($xxx ne "load")
            $yyy = $xxx
        endif
    done
    if (!strlen($yyy))
        abort "no keytable name in $_layout"
    endif
    shell("test -f ", $DOSEMU_LIB_DIR, "/keymap/", $yyy)
    if ( $DOSEMU_SHELL_RETURN)

```

```

        abort "keytable ",$yyy, "not found in, $DOSEMU_LIB_DIR,
"/keymap/*"
        endif
        $_layout 'include "keymap/', $yyy, ' " ' ;
        $$_layout
        keyboard { keybint $_keybint rawkeyboard $_rawkeyboard }
    endif
endif
secure off
$_disks = " "
ifdef restricted
    ## /etc/dosemu.users defined 'restricted' for this login
    if (!strlen($_secure)) secure on endif
    define c_normal
    if (strchr($_secure,"d")>=O) dexe { secure } endif
    ifdef guest
        if (strchr($_secure,"g")>=O) secure on endif
    else
        if (strchr($_secure,"n")>=O) secure on endif
        $_disks = $_hdimager
    endif
    if (strchr($_secure,"O")>=O) secure off endif
    dpmi off
    speaker emulated
    ipxsupport off
    # printer ( options "%s" command "lpr" timeout 20 )
else
    ## /etc/dosemu.users does allow full access for this login
    dexe { allowdisk }
    $_disks = $_hdimager
    if (!strlen($_vbootfloppy))
    bootC      # default boot mode is 'from C:'
    else
        bootA      # default boot mode is 'from A:'
        $yyy = " "; $zzz = " ";
    foreach $xxx ($LIST_DELIM, $_vbootfloppy)
        if (strlen($yyy))
            $zzz = $xxx
        else
            $yyy = $xxx
        endif
    endif
endif

```

```

done
    $_vbootfloppy = $DOSEMU_LIB_DIR, "/", $yyy
bootdisk { heads 2 sectors 18 tracks 80 threeinch file $_vbootfloppy }
    if (strlen($zzz))
$_disks = $_hdimage
    endif
endif
if (strlen($_floppy_a))
    $zzz = strsplit($_floppy_a, strstr($_floppy_a, ":"), 999);
    if (strlen($zzz))
$zzz = strsplit($zzz, 1, 999)
$_floppy_a = strdel($_floppy_a, strstr($_floppy_a, ":"), 999);
    else
        $zzz = "/dev/fd0"
    endif
$xxx = shell("test -r " . $zzz);
if ($DOSEMU_SHELL_RETURN)
warn "**** Warning: floppy ", $zzz, " not accessible, disabled";
    else
floppy { device $$zzz $_floppy_a }
    endif
endif
if (strlen($_floppy_b))
$zzz = strsplit($_floppy_b, strstr($_floppy_b, ":"), 999);
if (strlen($zzz))
$zzz = strsplit($zzz, 1, 999)
    $_floppy_b = strdel($_floppy_b, strstr($_floppy_b, ":"), 999);
    else
        $zzz = "/dev/fd1"
    endif
$xxx = shell("test -r ", $zzz);
if ($DOSEMU_SHELL_RETURN)
warn "**** Warning: floppy ", $zzz, " not accessible, disabled";
    else
floppy { device $$zzz $_floppy_b }
        endif
    endif
# misc stuff
dpmi $_dpmi
if (strlen($_irqpassing))
    $yyy = "irqpassing { "

```

```

        foreach $xxx (" ", $_irqpassing)
            $yyy = $yyy, "use_sigio ", $xxx
        done
        $yyy = $yyy, " } ";
        $$yyy
    else
        irqpassing off
    endif
    if (strlen($_hardware_ram))
        hardware_ram ( $$_hardware_ram )
    endif
    if (strlen($_speaker))
        $_speaker = "speaker ", $_speaker;
        $$_speaker
    else
        speaker off
    endif
    ipxsupport $_jpxsupport
    if ($_novell_hack) pktdriver novell_hack endif
    vnet $_vnet
    if (strlen($_printer))
        foreach $xxx ($LIST_DELIM, $_printer)
            $xxx = " ' -p". $xxx, " %s' ";
            printer { options $$xxx command "lpr" timeout
$_printer_timeout }
        done
    endif
    if (strlen($_ports)) ports { $$_ports } endif
    endif
## setting up hdimages
$_mounted_devices = " "
if (strlen($_disks) && (!defined(c_dexerun)))
    foreach $xxxx ($LIST_DELIM, $_disks)
        $xu_pref = " "
        if ((strchr($xxxx, "*") + 1)&&(strlen($xxxx) == strchr($xxxx, "*")+
1))
            $xxx_pref = strdel($xxxx, strlen($xxxx) -1, 1);
            $xxxx = shell("cd '/var/lib/dosemu/", $xxx_pref, " ' 2>/dev/null && echo -n *")
        if ($DOSEMU_SHELL_RETURN)
            $xxxx = strcat($xxx_pref, "*")
            $xxx_pref = " "

```

```

        endif
    endif
    foreach $xxx ($LIST_DELIM, $xxxx)
        $xxx = strcat($xxx_pref. $xxx)
        if (!strncmp($xxx, "/dev/", 4))
            if (!strlen($_mounted_devices))
                # we first get all mounted devices
                $_mounted_devices = shell(" df I awk '^devV/{print $1}' tr '\n' ' ' ");
                # we try to get the swap device too
                $_mounted_devices = $_mounted_devices, shell("awk '!/^#/  && /dev/ &&
                /swap/{print $1}' /etc/fstab | tr '\n' ' ' ");
            endif
            $yyy = strdel($xxx, strstr($xxx, ":ro"), 999);
            $zzz = strsplit($xxx, strstr($xxx, ":ro"), 999);
            $yyyy = strcat($yyy. ' ');
            if (strstr($_mounted_devices, $yyyy) <0)
                # device not in use
                if (strlen($zzz))
                    disk { partition $yyy readonly }
            else
                disk { partition $yyy };
            endif
            else
                abort "**** device ", $yyy, " in use, cannot do partition access"
            endif
        else
            $yyy = strdel($xxx, strstr($xxx, ":ro"), 999);
            $zzz = strsplit($xxx, strstr($xxx, ":ro"), 999);
            $yyy = $DOSEMU_HDIMAGE_DIR, "/", $yyy
            shell("test -d ", $yyy, "")
            if (!$DOSEMU_SHELL_RETURN)
                if (strlen($zzz))
                    disk { directory $yyy readonly };
                else
                    disk { directory $yyy };
                endif
            else
                disk { image $yyy }
            endif
        endif
    endif
done

```

```

done
endif
## setting up ASPI devices
ifndef restricted
if (strlen($_aspi))
foreach $xxx ($LIST_DELIM, $_aspi)
$zz = (1);
$yy2 ""; $yy3 = (-1);
foreach $yyy (":", $xxx)
    $zzz = "$yy", $zz, " = $yyy";
    $zz = ($zz + 1);
    $$zzz
done;
    api { $yy1 devicetype $yy2 target $yy3 };
done
endif
endif
## setting up the features list
if ( ($DOSEMU_VERSION_CODE >= ((98<<16) | (3<<8) | 3))
&& ($DOSEMU_VERSION_CODE < (99<<16)) )
    || ($DOSEMU_VERSION_CODE > ((99 << 16) (5<< 8))) )
if (strlen($features))
    foreach $xxx ($LIST_DELIM, $_features)
        $yyy = strdel($xxx, strstr($xxx, ":"), 999);
        $zzz = strsplit($xxx, strstr($xxx, ":"), 999);
        if (strlen($zzz))
            $zzz = strsplit($zzz, 1, 999);
        else
            $zzz = (0);
        endif
        feature { $yyy = $zzz };
    done
endif
endif
# ... here we are and need disallow 'shell' any further
# so we can restore PATH
$PATH = $_DOSEMU_ORIG_PATH
## define the allowed classes for subsequent .dosrc and -I parsing ifdef restricted
undef c-all
ifdef guest
undef c_normal

```

```
define c_dexe
define c_nice
define c_x
endif
endif
endif
#####
```

TÀI LIỆU THAM KHẢO

1. Nguyễn Quang Huy, *Nhập môn hệ điều hành Linux*, NXB Khoa học kỹ thuật, 2005.
2. Lê Tuấn, *Unix hệ điều hành và một số vấn đề quản trị mạng*, NXB Khoa học kỹ thuật, 2003.
3. Roderick W. Smith, *Linux+ Study Guide*, SYBEX Inc, 2005.

4. Stephen Stafford & Alex Weeks, *The Linux System Administrator's Guide*, 2003.
5. Sun Certified System Administrator for Solaris 10 Study Guide.
6. Tiêu Đông Nhơn, *Giáo Trình Hệ Điều Hành Linux*, NXB ĐH Quốc gia TP.HCM, 2008