



HƯỚNG DẪN GIẢNG DẠY



CHƯƠNG TRÌNH KỸ THUẬT VIÊN
Ngành **MẠNG & PHẦN CỨNG**
Học phần IV

CHỨNG CHỈ
QUẢN TRỊ MẠNG LINUX



MỤC LỤC

MỤC LỤC	2
MỤC TIÊU	11
ĐỐI TƯỢNG HỌC VIÊN	11
PHÂN BỐ BÀI GIẢNG	11
BÀI 1 Giới Thiệu Hệ Điều Hành Linux	13
Tóm tắt	13
I. Vài dòng lịch sử về Linux	14
II. Lịch sử phát triển của Linux	15
III. Những ưu điểm của Linux	16
III.1. Khả năng tương thích với các hệ mở.....	16
III.2. Hỗ trợ ứng dụng.....	16
III.3. Lợi ích cho giới chuyên nghiệp điện toán.....	16
IV. Khuyết điểm của Linux	16
IV.1. Hỗ trợ kỹ thuật.....	16
IV.2. phần cứng	17
V. Kiến trúc của hệ điều hành Linux	17
V.1. Hạt nhân (Kernel)	17
V.2. Shell	18
V.3. Các tiện ích	18
V.4. Chương trình ứng dụng.....	18
VI. Các đặc tính cơ bản của Linux	18
VI.1. Đa tiến trình.....	18
VI.2. Tốc độ cao.....	18
VI.3. Bộ nhớ ảo.....	19
VI.4. Sử dụng chung thư viện.....	19
VI.5. Sử dụng các chương trình xử lý văn bản.....	19
VI.6. Sử dụng giao diện cửa sổ	19
VI.7. Network Information Service (NIS).....	19
VI.8. Lập lịch hoạt động chương trình, ứng dụng.....	19
VI.9. Các tiện ích sao lưu dữ liệu	20
VI.10. Hỗ trợ nhiều ngôn ngữ lập trình.	20
BÀI 2 Cài Đặt Hệ Điều Hành Linux	21
Tóm tắt	21
I. Yêu cầu phân cứng	22
II. Đĩa cứng và phân vùng đĩa trong Linux	22
III. Quản lý ổ đĩa và partition trong Linux	22
IV. Khởi động chương trình cài đặt	23
IV.1. Boot từ CD-ROM.....	23
IV.2. Boot từ đĩa khởi động Windows	23
IV.3. Boot từ đĩa mềm khởi động Linux	23
V. Các bước cài đặt hệ điều hành Linux	24
V.1. Chọn phương thức cài đặt	24
V.2. Chọn chế độ cài đặt	24
V.3. Chọn ngôn ngữ hiển thị trong quá trình cài đặt.....	24
V.4. Cấu hình bàn phím.....	25



V.5.	Chọn cấu hình mouse	25
V.6.	Lựa chọn loại màn hình.....	25
V.7.	Lựa chọn loại cài đặt	26
V.8.	Chia Partition.....	27
V.9.	Lựa chọn Automatically partition	27
V.10.	Chia Partition bằng Disk Druid	28
V.11.	Cài đặt chương trình Boot Loader.....	29
V.12.	Cấu hình mạng	30
V.13.	Cấu hình Firewall	31
V.14.	Chọn ngôn ngữ hỗ trợ trong Linux	31
V.15.	Cấu hình khu vực địa lý của hệ thống.....	31
V.16.	Đặt mật khẩu cho người quản trị.....	32
V.17.	Cấu hình chứng thực	32
V.18.	Chọn các chương trình và Package cài đặt	33
V.19.	Định dạng filesystem và tiến hành cài đặt.....	34
VI.	Cấu hình thiết bị.....	34
VI.1.	Bộ nhớ (RAM)	34
VI.2.	Vị trí lưu trữ tài nguyên.....	34
VI.3.	Hỗ trợ USB.....	35
VI.4.	Network Card	35
VI.5.	Cài đặt modem	35
VI.6.	Cài đặt và cấu hình máy in.....	36
VII.	Sử dụng hệ thống.....	37
VII.1.	Đăng nhập.....	37
VII.2.	Một số lệnh cơ bản.....	38
VII.3.	Sử dụng trợ giúp man	38
VIII.	Khởi động hệ thống.....	39
VIII.1.	Các bước khởi động hệ thống:.....	39
IX.	Shutdown và Reboot hệ thống	41
X.	Sử dụng runlevel.....	41
XI.	Phục hồi mật khẩu cho user quản trị.....	41
XII.	Tìm hiểu boot loader	42
XII.1.	GRUB boot loader	42
XII.2.	LILO boot loader.....	44
BÀI 3 Hệ Thống Tập Tin.....		46
Tóm tắt		46
I.	Cấu trúc hệ thống tập tin	47
I.1.	Loại tập tin.....	48
I.2.	Liên kết tập tin	48
II.	Cấu trúc cây thư mục.....	49
III.	Các thao tác trên hệ thống tập tin và đĩa	51
III.1.	Mount và umount một hệ thống tập tin	51
III.2.	Định dạng filesystem	53
III.3.	Quản lý dung lượng đĩa.....	53
III.4.	Duy trì hệ thống tập tin với lệnh fsck.....	54
IV.	Các thao tác trên tập tin và thư mục	54



IV.1.	Thao tác trên thư mục	54
IV.2.	Tập tin	56
IV.3.	Các tập tin chuẩn trong Linux.....	58
IV.4.	Đường ống (Pipe).....	60
IV.5.	Lệnh tee	60
V.	Lưu trữ tập tin/thư mục	60
V.1.	Lệnh gzip/gunzip	60
V.2.	Lệnh tar	60
VI.	Bảo mật hệ thống tập tin	61
VI.1.	Quyền hạn.....	61
VI.2.	Lệnh chmod, chown, chgrp	63
Bài 4 Cài Đặt Phần Mềm		65
Tóm tắt		65
I.	Chương trình RPM.....	66
II.	Đặc tính của RPM	66
III.	Lệnh rpm	66
III.1.	Cài đặt phần mềm bằng rpm	66
III.2.	Loại bỏ phần mềm đã cài đặt trong hệ thống	67
III.3.	Nâng cấp phần mềm	68
III.4.	Truy vấn các phần mềm.....	68
III.5.	Kiểm tra các tập tin đã cài đặt.....	69
III.6.	Cài đặt phần mềm file nguồn *.tar, *.tgz.....	69
Bài 5 Giới Thiệu Các Trình Tiện Ích		71
Tóm tắt		71
I.	Trình soạn thảo vi	72
I.1.	Một số hàm lệnh của vi	72
I.2.	Chuyển chế độ lệnh sang chế độ soạn thảo	72
I.3.	Chuyển chế độ soạn thảo sang chế độ lệnh	72
II.	Trình tiện tích mail.....	74
III.	Tiện ích tạo đĩa mềm boot.....	75
IV.	Trình tiện ích setup	75
V.	Trình tiện ích fdisk	76
VI.	Trình tiện ích iptraf	77
VII.	Trình tiện ích lynx	77
VIII.	Trình tiện ích mc	78
Bài 6 Quản Trị Người Dùng Và Nhóm		79
Tóm tắt		79
I.	Superuser	80
II.	Thông tin của User	80
II.1.	Tập tin /etc/passwd	80
II.2.	Username và UserID	81
II.3.	Mật khẩu người dùng	82
II.4.	Group ID.....	82
II.5.	Home directory	82
III.	Quản lý người dùng.....	82
III.1.	Tạo tài khoản người dùng	82
III.2.	Thay đổi thông tin của tài khoản	83



III.3.	Tạm khóa tài khoản người dùng	84
III.4.	Hủy tài khoản	84
IV.	Nhóm người dùng	84
IV.1.	Tạo nhóm	84
IV.2.	Thêm người dùng vào nhóm	84
IV.3.	Hủy nhóm	85
IV.4.	Xem thông tin về user và group	85
	BÀI 7 Quản Lý Tài Nguyên Đĩa Cứng	86
	Tóm tắt	86
I.	Giới thiệu QUOTA	87
II.	Thiết lập Quota.....	87
II.1.	Chỉnh sửa tập tin /etc/fstab	87
II.2.	Thực hiện quotacheck.....	88
II.3.	Phân bổ quota	88
III.	Kiểm tra và thống kê hạn ngạch	89
IV.	Thay đổi Grace Periods	89
	BÀI 08 Cấu Hình Mạng.....	90
	Tóm tắt	90
I.	Đặt tên máy.....	91
II.	Cấu hình địa chỉ IP cho NIC	91
II.1.	Xem địa chỉ IP	91
II.2.	Thay đổi địa chỉ IP	91
II.3.	Tạo nhiều địa chỉ IP trên card mạng	92
II.4.	Lệnh netstat.....	93
III.	Thay đổi default gateway.....	94
III.1.	Mô tả đường đi (route) thông qua script file	94
III.2.	Xóa route trong bảng định tuyến	95
IV.	Truy cập từ xa	95
IV.1.	xinetd.....	95
IV.2.	Tập tin /etc/services	96
IV.3.	Khởi động xinetd	97
V.	Telnet	97
V.1.	Khái niệm telnet.....	97
V.2.	Cài đặt.....	97
V.3.	Cấu hình.....	98
V.4.	Bảo mật dịch vụ telnet.....	99
VI.	Secure Remote Access – SSH (Secure Shell)	100
VI.1.	Cài đặt SSH Server trên Server Linux.....	100
VI.2.	Sử dụng SSH Client trên Linux	100
VI.3.	Quản trị hệ thống Linux thông qua SSH client for Windows:.....	100
VII.	Dynamic Host Configuration Protocol.....	101
VII.1.	Một số đặc điểm cần lưu ý trên DHCP Server	101
VII.2.	Ưu điểm của việc sử dụng DHCP	101
VII.3.	Cấu hình DHCP Server	101
VII.4.	Khởi động dịch vụ DHCP:	102
	BÀI 9 SAMBA.....	103
	Tóm tắt	103



I.	Cài đặt SAMBA.....	104
II.	Khởi động dịch vụ SAMBA	104
III.	Cấu hình Samba Server	104
III.1.	Đoạn [global]	105
III.2.	Đoạn [homes]	105
III.3.	Chia sẻ máy in dùng SMB	106
III.4.	Chia sẻ thư mục	106
IV.	Sử dụng SAMBA SWAT	106
IV.1.	Tập tin cấu hình SAMBA SWAT	106
IV.2.	Truy xuất SWAT từ Internet Explorer	107
IV.3.	Cấu hình SAMBA SWAT	108
V.	Khởi động Samba Server	108
VI.	Sử dụng SMB client	108
VII.	Mount thư mục chia sẻ.....	109
VIII.	Mount tự động tài nguyên từ SMB Server	109
IX.	Mã hoá mật khẩu.....	110
BÀI 10 Network File System.....		111
Tóm tắt		111
I.	Tổng quan về quá trình hoạt động của NFS.....	112
I.1.	Một số luật chung khi cấu hình NFS	112
I.2.	Một số khái niệm chính về NFS	112
II.	Cài đặt NFS.....	112
III.	Cấu hình NFS.....	113
III.1.	Cấu hình NFS Server	113
III.2.	Cấu hình NFS Client	114
III.3.	Kích hoạt file /etc/exports	115
III.4.	Troubleshooting NFS Server	115
BÀI 11 LẬP TRÌNH SHELL TRÊN LINUX.....		117
Tóm tắt		117
I.	Giới thiệu về SHELL Và Lập Trình SHELL.....	118
I.1.	Giới thiệu về Shell	118
I.2.	Lập cấu hình môi trường đăng nhập.....	119
II.	Mục đích và ý nghĩa của việc lập trình Shell	121
III.	Điều khiển Shell từ dòng lệnh.....	121
IV.	Điều khiển tập tin lệnh.....	122
V.	Cú pháp ngôn ngữ Shell	123
V.1.	Ghi chú, định shell thực thi, thoát chương trình	123
V.2.	Sử dụng biến	124
V.3.	Lệnh kiểm tra.....	126
V.4.	Biểu thức tính toán expr	127
V.5.	Kết nối lệnh, khối lệnh và lấy giá trị của lệnh	128
V.6.	Cấu trúc rẽ nhánh If.....	128
V.7.	Cấu trúc lựa chọn Case	130
V.8.	Cấu trúc lặp	130
V.9.	Lệnh break, continue, exit	132
V.10.	Các lệnh khác.....	133



V.11. Hàm(function).....	133
BÀI 12 Quản Lý Tiến Trình.....	135
Tóm tắt	135
I. Định nghĩa	136
II. Xem thông tin tiến trình.....	137
III. Tiến trình tiên cảnh(foreground process).....	138
IV. Tiến trình hậu cảnh(background process).....	138
V. Tạm dừng và đánh thức tiến trình	138
VI. Hủy một tiến trình.....	139
VII. Chương trình lập lịch at	139
VIII. Chương trình lập lịch batch.....	140
IX. Chương trình lập lịch crontab	140
BÀI 13 Domain Name System	142
Tóm tắt	142
I. Giới thiệu về DNS	143
II. Cách phân bổ dữ liệu quản lý domain name.....	146
III. Cơ chế phân giải tên.....	146
III.1. Phân giải tên thành IP	146
III.2. Phân giải IP thành tên máy tính	147
IV. Sự khác nhau giữa domain name và zone	148
V. Fully Qualified Domain Name (FQDN)	149
VI. Phân loại Domain Name Server	149
VI.1. Primary Name Server	149
VI.2. Secondary Name Server	149
VI.3. Caching Name Server	149
VII. Sự ủy quyền(Delegating Subdomains)	150
VIII. Resource Record (RR).....	150
VIII.1. SOA(Start of Authority).....	150
VIII.2. NS (Name Server).....	151
VIII.3. A (Address) và CNAME (Canonical Name).....	152
VIII.4. MX (Mail Exchange).....	152
VIII.5. PTR (Pointer)	153
IX. Hoạt động của Name Server trong Linux.....	153
X. Cài đặt BIND.....	153
X.1. Một số file cấu hình quan trọng	154
X.2. Cấu hình.....	154
XI. Kiểm tra hoạt động của DNS.....	157
XII. Cấu hình Secondary Name Server.....	158
XIII. Một số quy ước.....	158
XIV. Cấu hình sự ủy quyền cho các miền con	160
BÀI 13 File Transfer Protocol	161
Tóm tắt	161
I. Giới thiệu về FTP	162
I.1. Giao thức FTP	162
II. Chương trình FTP Server.....	165
III. Chương trình FTP client	166



IV.	Giới thiệu VsFTP	168
IV.1.	Những tập tin được cài đặt liên quan đến vsftpd	168
IV.2.	Khởi động và dừng vsftpd	168
IV.3.	Một số thông số cấu hình mặc định	168
IV.4.	Những tùy chọn cấu hình vsftpd	169
V.	Cấu hình Virtual FTP Server	171
V.1.	Logging.....	171
V.2.	Network	171
	BÀI 14 WEB SERVER.....	172
	Tóm tắt	172
I.	Giới thiệu về Web Server	173
I.1.	Giao thức HTTP	173
I.2.	Web Server và cách hoạt động	174
I.3.	Web client.....	175
I.4.	Web động.....	175
II.	Giới thiệu Apache	175
II.1.	Cài đặt Apache.....	176
II.2.	Tạm dừng và khởi động lại Apache	176
II.3.	Sự chứng thực, cấp phép, điều khiển việc truy cập.....	176
II.4.	Điều khiển truy cập.....	179
II.5.	Khảo sát log file trên apache.....	180
III.	Cấu hình Web Server.....	181
III.1.	Định nghĩa về ServerName	181
III.2.	Thư mục Webroot và một số thông tin cần thiết.....	182
III.3.	Cấu hình mạng.....	183
III.4.	Alias.....	184
III.5.	UserDir	184
III.6.	VirtualHost.....	185
	BÀI 15 MAIL SERVER.....	188
	Tóm tắt	188
I.	Những giao thức mail	189
I.1.	SMTP(Simple Mail Transfer Protocol).....	189
I.2.	Post Office Protocol.....	191
II.	Giới thiệu về hệ thống mail.....	193
II.1.	Mail gateway	193
II.2.	Mail Host	193
II.3.	Mail Server	194
II.4.	Mail Client.....	194
II.5.	Một số sơ đồ hệ thống mail thường dùng	194
III.	Những chương trình mail và một số khái niệm	195
III.1.	Mail User Agent (MUA).....	195
III.2.	Mail Transfer Agent (MTA)	195
III.3.	Mailbox	195
III.4.	Hàng đợi (queue)	196
III.5.	Alias.....	196
IV.	DNS và Sendmail.....	200
V.	Những tập tin cấu hình Sendmail	201



V.1.	Tập tin /etc/sendmail.cf	201
V.2.	Macro	202
V.3.	Sendmail macro	203
V.4.	Tùy chọn (Option).....	203
V.5.	Định nghĩa các mailer.....	204
V.6.	Rule	204
V.7.	Rule set	205
VI.	Tập tin /etc/aliases.....	206
VII.	Cấu hình Mail Server với Sendmail	206
VIII.	Một số file cấu hình trong sendmail.....	207
VIII.1.	File /etc/mail/access	207
VIII.2.	File /etc/mail/local-host-names	207
VIII.3.	File /etc/mail/virtusertable.....	208
VIII.4.	File /etc/mail/mailertable.....	208
VIII.5.	File /etc/mail/domaintable.....	209
IX.	Cấu hình POP Mail Server	209
X.	Cài đặt và cấu hình Webmail - Openwebmail	209
X.1.	Cài đặt và cấu hình Open Webmail.....	210
X.2.	Cài đặt Open Webmail từ Source code.....	211
BÀI 16 PROXY SERVER		215
Tóm tắt		215
I.	Firewall.....	216
I.1.	Giới thiệu về Firewall.....	216
I.2.	Những chính sách Firewall.....	216
I.3.	Các loại Firewall và cách hoạt động.....	217
II.	Squid Proxy	219
II.1.	Giới thiệu Squid.....	219
II.2.	Những giao thức hỗ trợ trên Squid	219
II.3.	Trao đổi cache.....	219
II.4.	Cài đặt Squid Proxy.....	219
II.5.	Cấu hình.....	220
II.6.	Khởi động Squid.....	223
BÀI 17 Linux Security		224
Tóm tắt		224
I.	Log File.....	225
II.	Giới hạn user	225
III.	Network security.....	225
III.1.	Host Based security	225
III.2.	Port based security.....	226
BÀI 18 Webmin.....		239
Tóm tắt		239
I.	Giới thiệu Webmin	240
II.	Cài đặt Webmin	240
II.1.	Cài đặt từ file nhị phân	240
II.2.	Cài đặt Webmin từ file nguồn *.tar.gz	240
III.	Cấu hình Webmin	241
III.1.	Đăng nhập vào Webmin Server	241



III.2.	Cấu hình Webmin.....	241
III.3.	Cấu hình Webmin qua Web Browser	242
III.4.	Quản lý Webmin User	245
III.5.	Webmin cho Users(Usermin)	245
III.6.	Sử dụng Usermin	246
III.7.	Cấu hình hệ thống qua Webmin.....	248
III.8.	Cấu hình Server và Daemon.....	249
III.9.	Cấu hình mạng thông qua Webmin.....	250
III.10.	Cấu hình Hardware trên Webmin	251
III.11.	Linux Cluster trên Webmin	252
III.12.	Các thành phần khác(Others) trên Webmin	253
ĐỀ THI CUỐI HỌC PHẦN.....		254
I.	Cấu trúc đề thi.....	254
II.	Đề thi mẫu.....	256
II.1.	Đề thi mẫu cuối môn - Hệ Điều Hành Linux	256
II.2.	Đề thi cuối môn - Dịch Vụ Mạng Linux	258
ĐỀ THI CUỐI HỌC PHẦN.....		260
I.	Mẫu Đề thi lý thuyết.....	260
II.	Mẫu đề thi thực hành	267
ĐỀ THI KIỂM TRA CHUYÊN MÔN GIÁO VIÊN.....		269



MỤC TIÊU

Sau khi hoàn thành khóa học, học viên sẽ có khả năng:

- Cài đặt và sử dụng hệ điều hành Linux (phiên bản mới nhất của RedHat) và thực thi được các thao tác tạo tập tin, thư mục, quản lý người dùng, cấp quyền hạn sử dụng tài nguyên, soạn thảo văn bản bằng các công cụ, chia sẻ tài nguyên thông qua dịch vụ Samba, đặt hạn ngạch để giới hạn sử dụng tài nguyên đĩa cứng.
- Cấu hình và quản trị các dịch vụ mạng trên hệ thống Linux như: DNS, FTP, WEB, MAIL, PROXY.
- Thiết lập một số cơ chế bảo mật hệ thống Linux thông qua các công cụ như: iptables, tcp_wrappers,...
- Tổ chức hệ thống cho phép người dùng có thể làm việc từ xa qua Web, SSH, Telnet, SFTP sử dụng các công cụ như: Webmin, Usermin, OpenSSH, TELNET.

ĐỐI TƯỢNG HỌC VIÊN

Học sinh, sinh viên, kỹ sư CNTT, những nhân viên quản trị mạng (cơ quan, xí nghiệp) muốn bổ sung kiến thức quản trị mạng trên môi trường Linux.

PHÂN BỐ BÀI GIẢNG

Thời lượng: 96LT + 120TH

STT	Bài học	Số tiết LT	Số tiết TH
1	Giới thiệu về Linux	3	
2	Cài đặt hệ điều hành RedHat Linux	5	5
3	Quản lý hệ thống tập tin	8	10
4	Cài đặt phần mềm	3	5
5	Giới thiệu các trình tiện ích	4	5
6	Quản trị người dùng	5	5
7	Quản lý tài nguyên đĩa cứng	3	5
8	Cấu hình mạng	5	10
9	SAMBA	4	5
10	NFS	3	5
11	Lập trình Shell trên Linux	5	5



12	Quản lý tiến trình	5	5
13	Dịch vụ DNS	5	10
14	Dịch vụ FTP	5	5
15	Dịch vụ Web	5	5
16	Dịch vụ Mail	8	10
17	Dịch vụ Proxy	5	5
18	Linux Security	10	10
19	Webmin	5	5
20	Ôn tập		5
Tổng số tiết		96	120



BÀI 1

Giới Thiệu Hệ Điều Hành Linux

Tóm tắt

Lý thuyết: 3 tiết - thực hành: 0 tiết

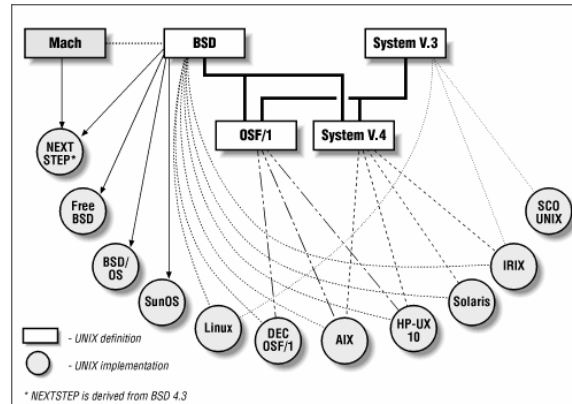
Mục tiêu	Các mục chính	Bài tập bắt buộc	Bài tập làm thêm
Bài học này giới thiệu sơ lược về lịch sử phát triển, kiến trúc của Linux, ưu và nhược điểm của Linux so với các hệ điều hành khác.	<ol style="list-style-type: none"> I. Vài dòng về lịch sử Linux. II. Lịch sử phát triển của Linux. III. Những ưu điểm của Linux. IV. khuyết điểm của Linux. 		



I. Vài dòng lịch sử về Linux

Giữa năm 1960, AT & T Bell Laboratories và một số trung tâm khác tham gia vào một cố gắng nhằm tạo ra một hệ điều hành mới được đặt tên là Multics (Multiplexed Information and Computing Service). Đến năm 1969, chương trình Multics bị bãi bỏ vì đó là một dự án quá nhiều tham vọng và do đó không khả thi. Thậm chí nhiều yêu cầu đối với Multics thời đó đến nay vẫn chưa có được trên các Unix mới nhất. Nhưng Ken Thompson, Dennis Richie và một số đồng nghiệp của Bell Labs đã không bỏ cuộc. Thay vì xây dựng một hệ điều hành làm nhiều việc một lúc như Multics, họ quyết định phát triển một hệ điều hành đơn giản chỉ làm tốt một công việc là chạy chương trình (run program). Hệ điều hành sẽ có rất nhiều các công cụ (tool) nhỏ, đơn giản, gọn nhẹ (compact) và chỉ làm tốt một công việc. Bằng cách kết hợp nhiều công cụ lại với nhau, họ sẽ có một chương trình thực hiện một công việc phức tạp. Đó cũng là cách thức người lập trình viết ra chương trình. Vào năm 1973, sử dụng ngôn ngữ C của Richie. Thompson đã viết lại toàn bộ hệ điều hành Unix và đây là một thay đổi quan trọng của Unix. Do đó, Unix từ chỗ là một hệ điều hành cho một máy PDP-xx trở thành hệ điều hành của các máy khác với một cố gắng tối thiểu để chuyển đổi. Khoảng 1977 bản quyền của UNIX được giải phóng và hệ điều hành UNIX trở thành một thương phẩm. Hai dòng UNIX: System V của AT&T, Novell và Berkeley Software Distribution (BSD) của Đại học Berkeley.

- **System V:** Các phiên bản UNIX cuối cùng do AT&T xuất bản là System III và một vài phát hành (releases) của System V. Hai bản phát hành gần đây của System V là Release 3.2 (SVR 3.2) và Release 4.2 (SVR 4.2). Phiên bản SVR 4.2 là phổ biến nhất từ máy PC cho tới máy tính lớn.
- **BSD:** Từ 1970 Computer Science Research Group của University of California tại Berkeley (UCB) xuất bản nhiều phiên bản UNIX, được biết đến dưới tên Berkeley Software Distribution, hay BSD. Cải tiến của PDP-11 được gọi là 1BSD và 2BSD. Trợ giúp cho các máy tính của Digital Equipment Corporation VAX được đưa vào trong 3BSD. Phát triển của VAX được tiếp tục với 4.0BSD, 4.1BSD, 4.2BSD và 4.3BSD.
- Trước 1992, UNIX là tên thuộc sở hữu của AT&T. từ 1992, khi AT&T bán bộ phận Unix cho Novell, tên Unix thuộc sở hữu của X/Open foundation. Tất cả các hệ điều hành thỏa mãn một số yêu cầu đều có thể gọi là Unix. Ngoài ra, Institute of Electrical and Electronic Engineers (IEEE) đã thiết lập chuẩn "An Industry-Recognized Operating System Interface Standard based on the UNIX Operating System". Kết quả cho ra đời POSIX.1 (cho giao diện C) và POSIX.2 (cho hệ thống lệnh trên Unix). Tóm lại, vấn đề chuẩn hóa UNIX vẫn còn rất xa kết quả cuối cùng. Nhưng đây là quá trình cần thiết có lợi cho sự phát triển của ngành tin học nói chung và sự sống còn của hệ điều hành UNIX nói riêng.



II. Lịch sử phát triển của Linux

- Năm 1991, Linus Torvalds, sinh viên của Đại học Tổng hợp Helsinki Phần Lan bắt đầu xem xét Minix, một phiên bản của Unix làm ra với mục đích nghiên cứu cách tạo ra một hệ điều hành Unix chạy trên máy PC với bộ vi xử lý Intel 80386.
- Ngày 25/8/1991, Linus cho ra version 0.01 và thông báo trên comp.os.minix về dự định của mình về Linux.
- 1/1992, Linus cho ra version 0.02 với shell và trình biên dịch C. Linux không cần Minix nữa để biên dịch lại hệ điều hành của mình. Linus đặt tên hệ điều hành của mình là Linux.
- 1994, phiên bản chính thức 1.0 được phát hành.
- Linux là một hệ điều hành dạng UNIX (Unix-like Operating System) chạy trên máy PC với bộ điều khiển trung tâm (CPU) Intel 80386 trở lên, hay các bộ vi xử lý trung tâm tương thích AMD, Cyrix. Linux ngày nay còn có thể chạy trên các máy Macintosh hoặc SUN Space. Linux thỏa mãn chuẩn POSIX.1.
- Linux được viết lại toàn bộ từ con số không, tức là không sử dụng một dòng lệnh nào của Unix để tránh vấn đề bản quyền của Unix. Tuy nhiên, hoạt động của Linux hoàn toàn dựa trên nguyên tắc của hệ điều hành Unix. Vì vậy, nếu một người nắm được Linux thì sẽ nắm được UNIX. Nên chú ý rằng giữa các Unix sự khác nhau cũng không kém gì giữa Unix và Linux.
- Linux là hệ điều hành phân phát miễn phí, phát triển trên mạng Internet, tựa Unix và được sử dụng trên máy tính cá nhân (PCs). Linux đã phát triển nhanh chóng và trở nên phổ biến trong thời gian ngắn. Nó nhanh chóng được nhiều người sử dụng vì một trong những lý do là không phải trả tiền bản quyền. Mọi người có thể dễ dàng download từ Internet hay mua tại các hiệu bán CD.
- Linux là hệ điều hành có hiệu năng cao, trong tất cả các máy tính có cấu hình cao hay thấp. Hệ điều hành này hỗ trợ các máy tính sử dụng 32 cũng như 64 bit và rất nhiều phần mềm khác nhau.
- Quá trình phát triển của Linux được tăng tốc bởi sự giúp đỡ của chương trình GNU (GNU's Not Unix). Đó là chương trình phát triển các Unix có khả năng chạy trên nhiều nền tảng khác nhau. Đến hôm nay, cuối 2001, phiên bản mới nhất của Linux kernel là 2.6.11.3, có khả năng điều khiển các máy đa bộ vi xử lý và rất nhiều các tính năng khác.



III. Những ưu điểm của Linux

Trong số những hệ điều hành thông dụng ngày nay, Linux là hệ điều hành miễn phí được sử dụng rộng rãi nhất. Với các PC IBM, Linux cung cấp một hệ thống đầy đủ với những chức năng đa nhiệm (multitasking) và đa người dùng (multiuser) lập sẵn, tận dụng được sức mạnh xử lý của máy 386 và cao hơn.

Linux có sẵn bộ giao thức TCP/IP giúp bạn dễ dàng kết nối Internet. Linux cũng có Xfree86 cung cấp cho bạn một giao diện đồ họa GUI đầy đủ. Những phần này bạn không cần phải mất tiền mua chỉ cần tải xuống từ Internet.

III.1. Khả năng tương thích với các hệ mở

Khả năng tương thích của một hệ điều hành giúp bạn chuyển nó từ một nền này sang một nền khác mà vẫn hoạt động tốt. Trước kia UNIX chỉ hoạt động trên một nền duy nhất, đó là máy điện toán mini DEC PDP-7. Hiện nay, UNIX chạy được trên bất kỳ nền nào, từ máy tính xách tay cho đến những máy tính lớn dạng mainframe. Nhờ tính tương thích này, các máy điện toán chạy UNIX trên nhiều nền khác nhau có thể liên lạc với nhau một cách chính xác và hữu hiệu với những loại nền khác.

III.2. Hỗ trợ ứng dụng

Hiện nay, Linux có hàng nghìn ứng dụng, bao gồm các chương trình báo biểu, cơ sở dữ liệu, xử lý văn bản... Ngoài ra, Linux cũng có hàng loạt trò chơi giải trí trên nền văn bản hoặc đồ họa.

III.3. Lợi ích cho giới chuyên nghiệp điện toán

Đến với Linux, giới điện toán sẽ có hàng loạt 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++, ...

IV. Khuyết điểm của Linux

IV.1. Hỗ trợ kỹ thuật

Có lẽ điều trở ngại nhất của Linux là không có một công ty nào chịu trách nhiệm phát triển hệ điều hành Linux này. Nếu có điều gì trục trặc, bạn không thể gọi miễn phí cho một bộ phận hỗ trợ kỹ thuật nào cả.

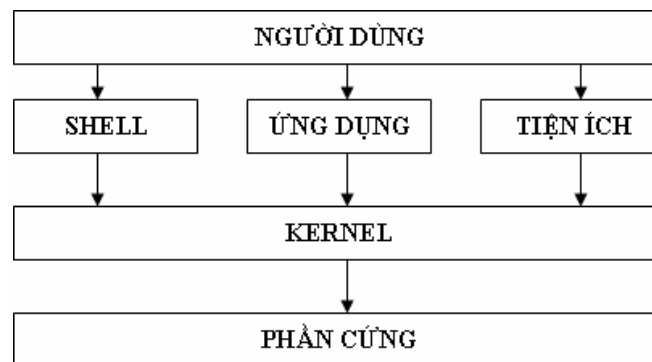
Thiếu nguồn trợ giúp kỹ thuật không chỉ đối với Linux mà cả với những ứng dụng Linux. Mặc dù, hiện có vài chương trình mang tính thương mại dành cho Linux, song đa phần lại là chương trình miễn phí do một nhóm nhỏ biên soạn rồi đưa lên mạng cho cả thế giới sử dụng chung.



IV.2. phần cứng

Một điều bất tiện nữa là thực sự Linux không dễ cài đặt và rất nhiều thành phần không tương thích với một vài phần cứng nào đó. Các nhà phát triển Linux là những người sống rải rác trên hành tinh này, do đó không thể có một chương trình được đảm bảo chất lượng như thông lệ. Các nhà phát triển cảm thấy chương trình của mình dùng được là tung ra cho mọi người cùng xài chứ không có một thời gian thử nghiệm chương trình. Hơn nữa, các phần cứng mà Linux hỗ trợ tùy thuộc vào loại máy móc mà các nhà phát triển sử dụng khi soạn thảo đoạn mã. Chính vì thế mà Linux không thể chạy trên tất cả mọi nền phần cứng của PC hiện nay.

V. Kiến trúc của hệ điều hành Linux



V.1. Hạt nhân (Kernel)

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

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

- Bảo vệ vùng nhớ giữa các tiến trình, điều này không cho phép một tiến trình làm tắt toàn bộ hệ thống.
- Chỉ tải các chương trình khi có yêu cầu.



V.2. Shell

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

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

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

V.3. Các tiện ích

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

V.4. Chương trình ứng dụng

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

VI. Các đặc tính cơ bản của Linux

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

VI.1. Đa tiến trình

Là đặc tính cho phép người dùng thực hiện nhiều tiến trình đồng thời. Ví dụ bạn vừa in, vừa soạn văn bản, vừa nghe nhạc... cùng một lúc. Máy tính sử dụng chỉ một CPU nhưng xử lý đồng thời nhiều tiến trình cùng lúc. Thực chất là tại một thời điểm CPU chỉ xử lý được một mệnh lệnh, việc thực hiện cùng lúc nhiều công việc là giả tạo bằng cách làm việc xen kẽ và chuyển đổi trong thời gian nhanh. Do đó người dùng cứ ngỡ là thực hiện đồng thời.

VI.2. Tốc độ cao

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



VI.3. Bộ nhớ ảo

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

VI.4. Sử dụng chung thư viện

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

VI.5. Sử dụng các chương trình xử lý văn bản

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

VI.6. Sử dụng giao diện cửa sổ

Giao diện cửa sổ dùng Hệ thống X Window, có giao diện như hệ điều hành Windows. Với hệ thống này người dùng rất thuận tiện khi làm việc trên hệ thống. X window System hay còn gọi tắt là X được phát triển tại viện Massachusetts Institute of Technology. Nó được phát triển để tạo ra môi trường làm việc không phụ thuộc phần cứng. X chạy dưới dạng client –server. Hệ thống X window hoạt động qua hai bộ phận :

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

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

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

VI.7. Network Information Service (NIS)

Dịch vụ NIS cho phép chia sẻ các tập tin password và group trên mạng. NIS là một hệ thống cơ sở dữ liệu dạng client-server, chứa các thông tin của người dùng và dùng để chứng thực người dùng. NIS xuất phát từ hãng Sun Microsystems với tên là Yellow Pages.

VI.8. Lập lịch hoạt động chương trình, ứng dụng

Chương trình lập lịch trong Linux xác định các ứng dụng, script thực thi theo một sự sắp xếp của người dùng như: at, cron, batch.



VI.9. Các tiện ích sao lưu dữ liệu

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

VI.10. Hỗ trợ nhiều ngôn ngữ lập trình.

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



BÀI 2

Cài Đặt Hệ Điều Hành Linux

Tóm tắt

Lý thuyết: 5 tiết - Thực hành: 5 tiết.

Mục tiêu	Các mục chính	Bài tập bắt buộc	Bài tập làm thêm
Giới thiệu cho học viên cách cài đặt hệ điều hành Linux, cài đặt các thiết bị, tìm hiểu nguyên lý hoạt động, chương trình khởi động hệ điều hành Linux.	<ol style="list-style-type: none"> I. Yêu cầu phần cứng. II. Đĩa cứng và phân vùng đĩa trong Linux. III. Quản lý ổ đĩa và partition trong Linux. IV. Khởi động chương trình cài đặt. V. Các bước cài đặt hệ điều hành Linux. VI. Cấu hình thiết bị. VII. Sử dụng hệ thống. VIII. Khởi động hệ thống. IX. Shutdown và Reboot hệ thống. X. Sử dụng runlevel. XI. Phục hồi mật khẩu cho user quản trị. XII. Tìm hiểu boot loader. 	Bài tập 02 (sách bài tập)	



I. Yêu cầu phần cứng

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

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

II. Ổ đĩa cứng và phân vùng đĩa trong Linux

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

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

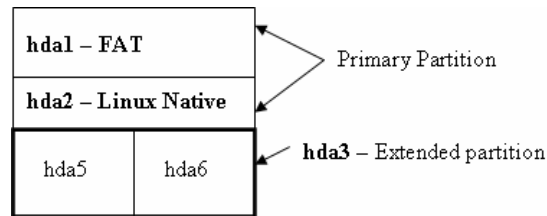
III. Quản lý ổ đĩa và partition trong Linux

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

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

Ví dụ :

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



Như hình vẽ trên là các partition của ổ cứng thứ nhất hda: có 2 partition chính ký hiệu là hda1 và hda2, một partititon mở rộng là hda3. Trong partition mở rộng hda3 có 2 partition logic có ký hiệu là hda6 và hda5. Trong Linux bắt buộc phải có tối thiểu 2 partition sau:

- Partition chính chứa thư mục gốc (/) và hạt nhân (gọi là Linux Native partition)
- Partition swap được dùng làm không gian hoán đổi dữ liệu khi vùng nhớ chính được sử dụng hết. Kích thước của phần swap sử dụng tùy thuộc hệ thống mình sử dụng nhiều hay ít ứng dụng. Thông thường thì kích thước vùng swap bằng kích thước bộ nhớ chính.

IV. Khởi động chương trình cài đặt

IV.1. Boot từ CD-ROM

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

IV.2. Boot từ đĩa khởi động Windows

BIOS của máy bạn không hỗ trợ boot được từ CD, bạn có thể khởi động từ đĩa khởi động DOS. Sau khi khởi động, đưa CD cài đặt vào ổ CD-ROM. Giả sử ổ CD của bạn là ổ E:. Bước kế bạn thực hiện.

Cd Dosutils Autoboot

IV.3. Boot từ đĩa mềm khởi động Linux

CD cài đặt Linux có chứa tập tin image giúp khởi động cài đặt Linux từ đĩa mềm. Trên RedHat Linux 7.x Image này lưu trong thư mục: <cdrom_write>\images\bootnet.img.

Trên RedHat 9.0 và Fedora core thì tập tin <cdrom_write>\images\bootdisk.img

Để bung tập tin image này ra đĩa mềm chúng ta dùng chương trình rawrite có trong thư mục dosutils của đĩa cài đặt. Trên môi trường Windows:

```
<cdrom_write>\dosutils\rawrite
```

```
Enter disk image soure file name : ..\bootnet.img
```

```
Enter the target disk device : A
```

```
Please insert formatted diskette into device A: and press – ENTER -- : enter
```

Trên môi trường Linux ta có thể dùng lệnh:

```
#dd if=/mnt/cdrom/images/<image name> of=/dev/fd0
```

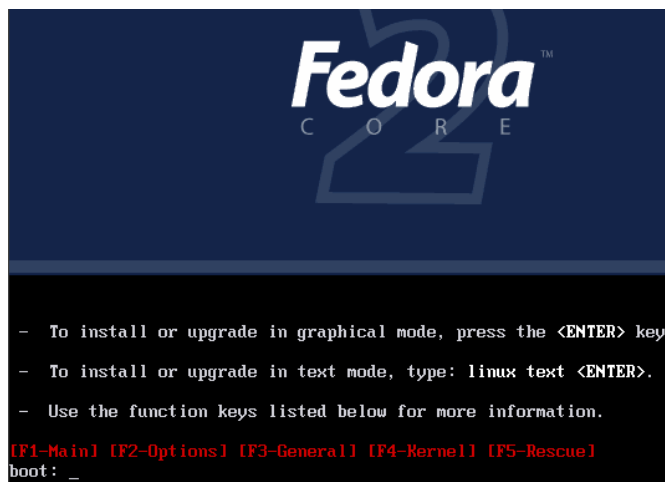
V. Các bước cài đặt hệ điều hành Linux

V.1. Chọn phương thức cài đặt

Nguồn cài đặt từ :

- CD-Rom: Có thể khởi động từ CD-ROM hoặc khởi động bằng đĩa mềm boot.
- Đĩa cứng: Cần sử dụng đĩa mềm boot(dùng lệnh dd hoặc mkbootdisk để tạo đĩa mềm boot).
- NFS image: Sử dụng đĩa khởi động mạng. Kết nối tới NFS sever.
- FTP: Sử dụng đĩa khởi động mạng. Cài trực tiếp qua kết nối FTP.
- HTTP: Sử dụng đĩa khởi động mạng. Cài trực tiếp qua kết nối HTTP.

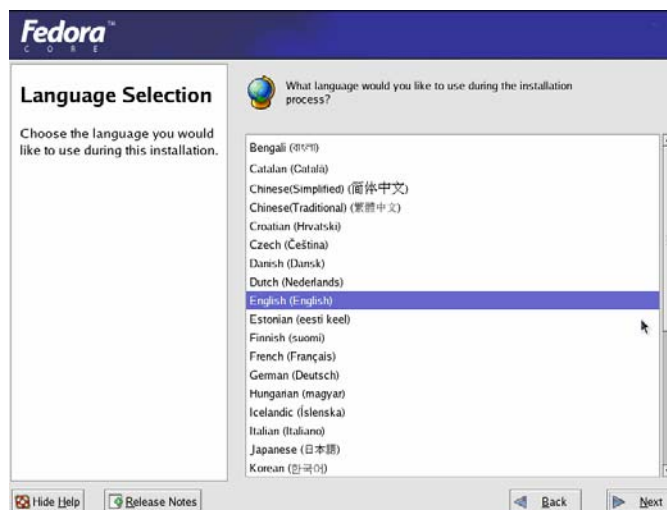
V.2. Chọn chế độ cài đặt



Chúng ta có thể chọn các chế độ:

- Linux text: Chương Hệ Điều Hành Linux đặt dưới chế độ text(Text mode).
- [Enter] : Chương Hệ Điều Hành Linux đặt dưới chế độ đồ họa(Graphical mode)

V.3. Chọn ngôn ngữ hiển thị trong quá trình cài đặt



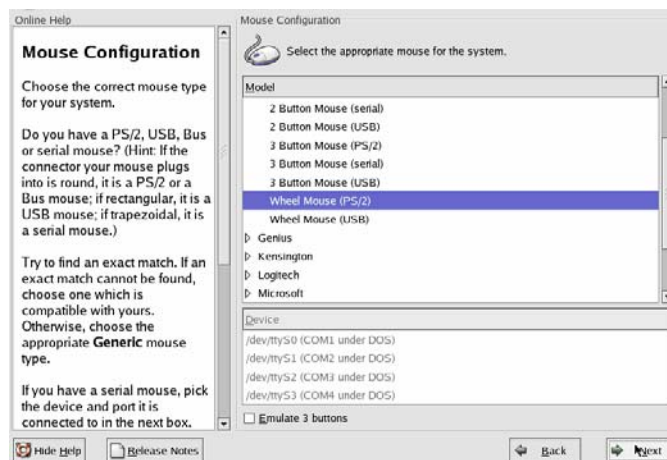
Chọn ngôn ngữ “English” rồi chọn Next

V.4. Cấu hình bàn phím



Chọn loại bàn phím của mình, chọn Next

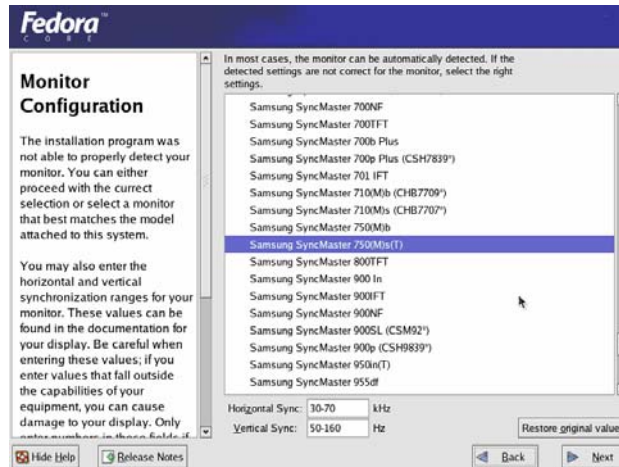
V.5. Chọn cấu hình mouse



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

V.6. Lựa chọn loại màn hình

Thông thường tại bước này hệ điều hành sẽ tự động nhận đúng loại màn hình hiển thị nếu không thì ta phải cấu hình lại màn hình hiển thị trong hộp thoại bên phải.



Chọn Next.

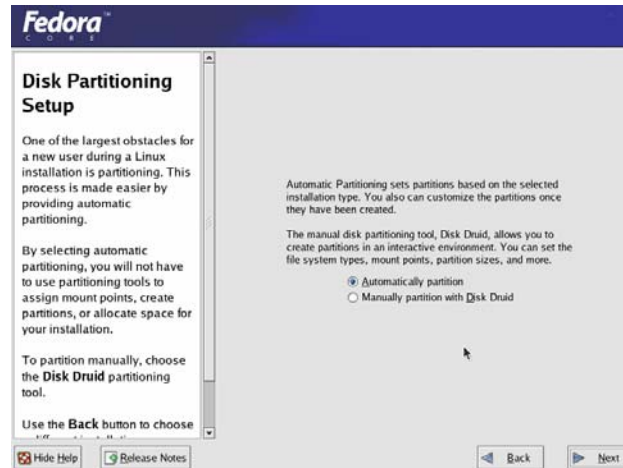
V.7. Lựa chọn loại cài đặt



Một số loại cài đặt thông dụng:

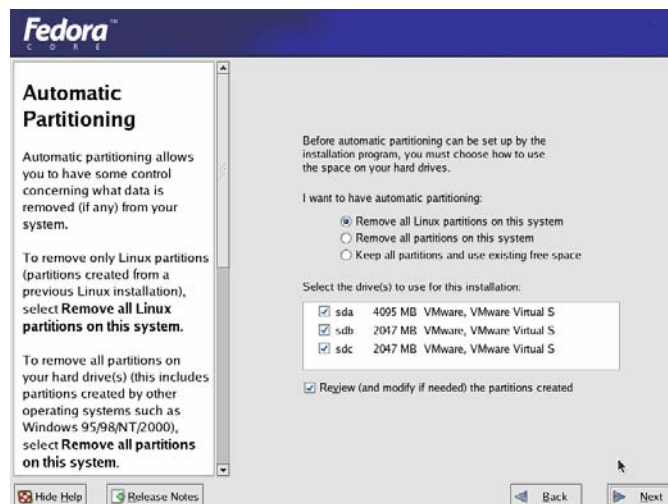
- **Workstation:** Cài đặt hệ điều hành phục vụ cho công việc của một máy trạm.
- **Server:** Cài đặt hệ điều hành phục vụ cho máy chủ.
- **Custom:** có thể tích hợp các tùy chọn trên một cách tùy ý.

V.8. Chia Partition

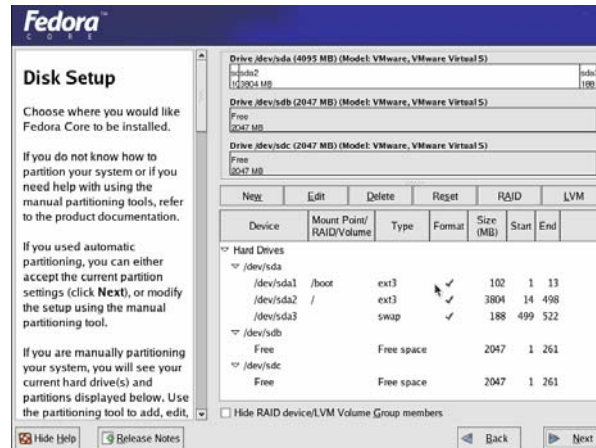


- **Automatically partition.:** cho phép hệ thống tự động phân vùng ổ đĩa hợp lý để cài hệ điều hành (thông thường theo cách này thì hệ thống sẽ tạo ra hai phân vùng: /boot, /, swap)
- **Manually partition with Disk Druid:** Chia partition bằng tiện ích Disk Druid. Đây là cách chia partition dưới dạng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

V.9. Lựa chọn Automatically partition



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

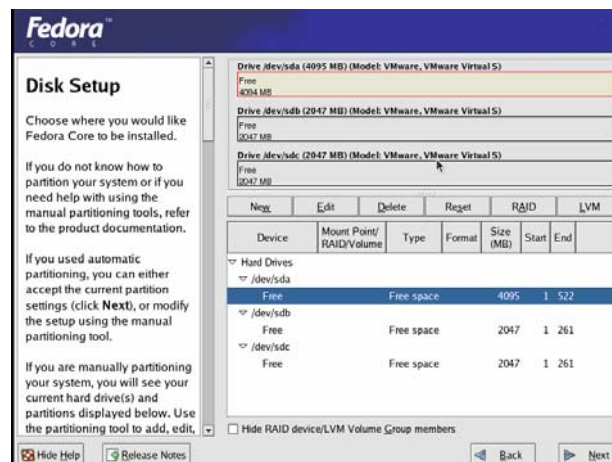


V.10. Chia Partition bằng Disk Druid

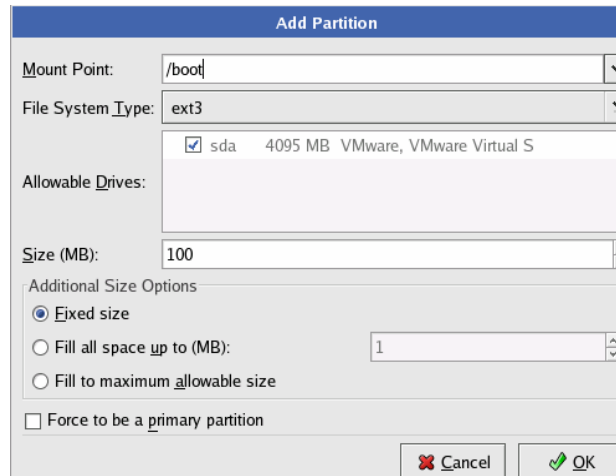
Trong bước 8 ta chọn Manually partition with Disk Druid để thực hiện phân chia phân vùng sử dụng tiện ích Disk Druid.

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

Chi tiết các partition gồm kích thước, loại hệ thống tập tin, thư mục được mount vào được mô tả trong hình sau:



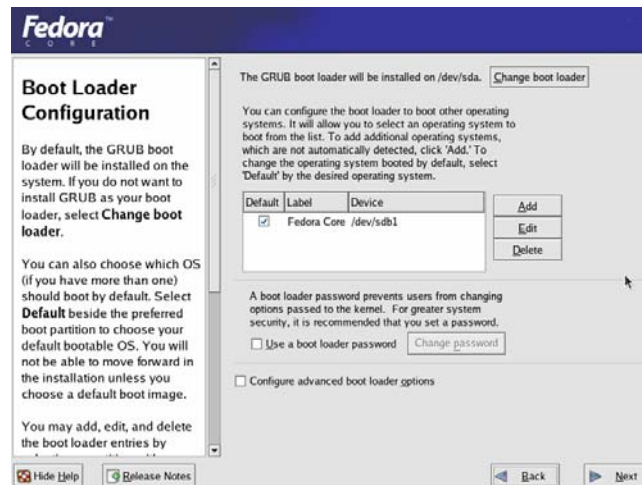
- **New:** Tạo một partition mới, chỉ định tên phân vùng(mount point), loại filesystem(ext3) và kích thước(size) tính bằng MByte(tùy chọn).



- **Edit:** Thay đổi lại các tham số của phân vùng được chọn.
- **Delete:** Xóa phân vùng được chọn.
- **Reset:** Phục hồi lại trạng thái đĩa như trước khi thao tác.
- **Make RAID:** Sử dụng với RAID (Redundant Array of Independent Disks) khi ta có ít nhất 3 đĩa cứng.

V.11. Cài đặt chương trình Boot Loader

Boot Loader là chương trình cho phép bạn chọn các hệ điều hành để khởi động qua menu. Khi chúng ta chọn, thì chúng xác định các tập tin cần thiết để khởi động hệ điều hành và giao quyền điều khiển lại cho hệ điều hành. Boot Loader có thể được cài vào Master Boot record hoặc vào sector đầu tiên của partition.

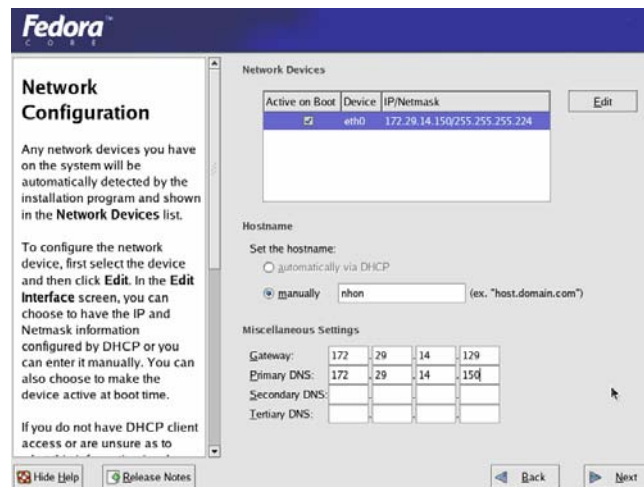


Linux cho phép bạn sử dụng chương trình Boot Loader là GRUB hoặc LILO. Cả 2 Boot Loader đều có thể hỗ trợ quản lý nhiều hệ điều hành trên một hệ thống.

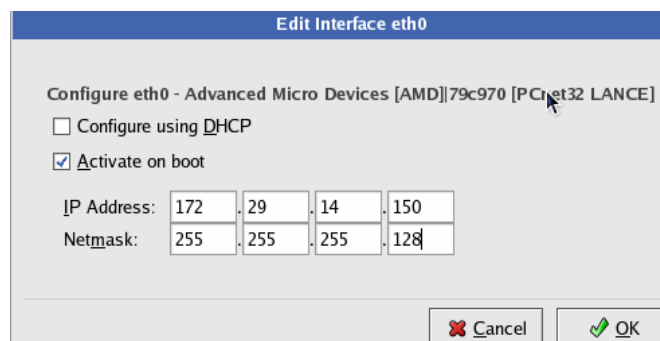
- Bạn chọn cài Boot Loader vào Master Boot Record (MBR) khi chưa có chương trình Boot Loader nào (Ví dụ như của Windows) được cài, hoặc bạn chắc chắn chương boot loader của bạn có thể khởi động được các hệ điều hành khác trong máy của mình. Khi cài lên MBR thì các chương trình Boot Loader trước đó sẽ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

- Chọn cài Boot loader vào sector đầu tiên của partition cài đặt khi bạn đã có chương trình Boot Loader tại MBR và không muốn thay thế nó. Trong trường hợp này, chương trình Boot Loader kia nắm quyền điều khiển trước và trở đến chương trình Boot Loader của Linux khi có yêu cầu khởi động hệ điều hành này.
- Bạn không cài chương trình Boot loader, khi đó bạn phải sử dụng đĩa mềm boot để khởi động hệ điều hành.
- Ta có thể đặt mật khẩu cho boot loader thông qua nút Change password.

V.12. Cấu hình mạng



Configure using DHCP: Bạn có thể chọn cấu hình TCP/IP động qua dịch vụ DHCP hoặc cấu hình cụ thể. Khi cấu hình cụ thể, bạn phải nhập những thông số cấu hình mạng trong mục chọn edit:



- IP Address: Chỉ định địa chỉ IP của host cài đặt.
- Netmask Address: subnet mask cho địa chỉ IP trên.

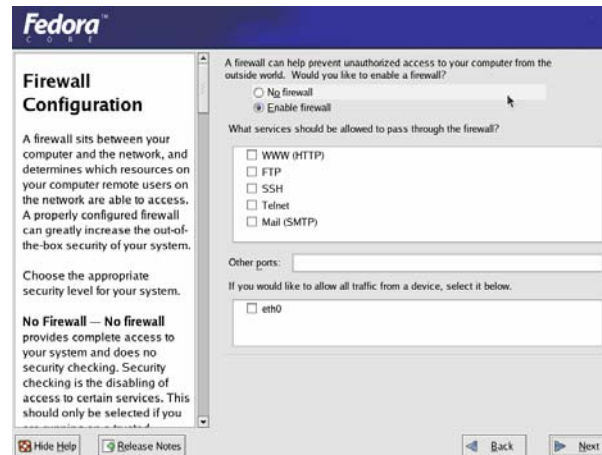
Active on boot: Card mạng được kích hoạt khi hệ điều hành khởi động.

Host name: Nếu bạn có tên dns đầy đủ thì khai báo tên đầy đủ. Trong trường hợp bạn không kết nối vào mạng, bạn cũng đặt tên cho máy thông qua mục manually. Nếu không tên nào được điền vào thì giá trị mặc nhiên sử dụng là localhost

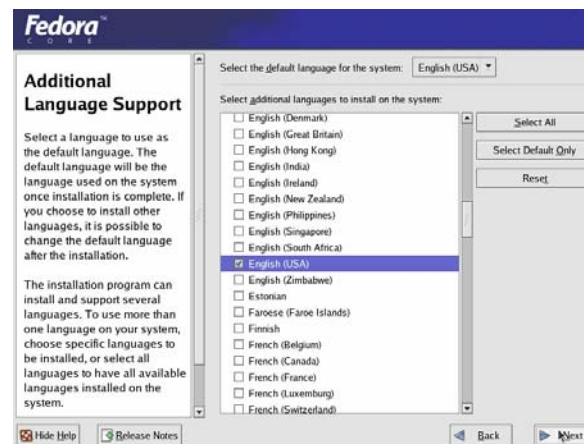
Miscellaneous Settings: để chỉ định địa chỉ gateway và Primary DNS, và một số thông số khác. Các trường không có giá trị thì các trường đó không được sử dụng trong hệ thống.

V.13. Cấu hình Firewall

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



V.14. Chọn ngôn ngữ hỗ trợ trong Linux



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

V.15. Cấu hình khu vực địa lý của hệ thống

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

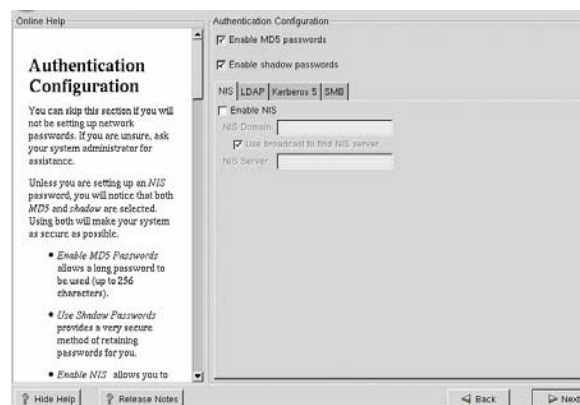


V.16. Đặt mật khẩu cho người quản trị

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



V.17. Cấu hình chứng thực



Nếu bạn không sử dụng password mạng có thể bỏ qua cấu hình này nhưng vẫn sử dụng chế độ chọn mặc nhiên (chọn Enable MD5 passwords và Enable shadow passwords)

Enable MD5 passwords: cho phép password sử dụng tới 256 ký tự thay vì chỉ tới 8 ký tự



Enable shadow passwords: cung cấp cơ chế lưu trữ password an toàn. Password được lưu trữ trong tập tin /etc/shadow và chỉ có root mới được đọc.

Enable NIS: cho phép một nhóm máy trong một NIS domain sử dụng chung tập tin passwd và group. Chọn các tham số sau :

- + NIS domain: Xác định NIS domain mà máy này tham gia
- + Use broadcast to find NIS server: Cho phép sử dụng thông điệp quảng bá để tìm NIS server.
- + NIS Server : Xác định NIS server.
- + Enable LDAP: Hệ thống của bạn sử dụng LDAP cho một vài hoặc tất cả các phép chứng thực.
- + LDAP Server : Xác định LDAP server (dùng địa chỉ IP)
- + LDAP Base DN: cho phép tìm kiếm thông tin người dùng dựa trên DN(Distinguished Name)
- + Use TLS (Transport Layer Security) lookups: tùy chọn này cho phép LDAP gọi tên người dùng và password mã hóa tới LDAP server trước khi chứng thực.

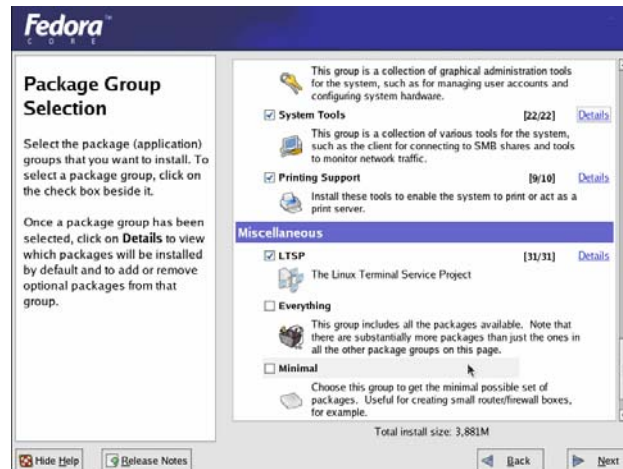
Enable Kerberos: là hệ thống cung cấp các dịch vụ chứng thực trên mạng. Các lựa chọn :

- + Realm: cho phép bạn truy xuất tới mạng sử dụng Kerberos.
- + KDC: cho phép bạn truy xuất tới Key Distribution Center (KDC).
- + Admin Server: cho phép bạn truy xuất tới server chạy kadmind
- + Enable SMB Authentication: Cài PAM để dùng một Samba server chứng thực cho các client.
- + SMB Server: Xác định samba server mà các máy trạm kết nối tới để chứng thực.
- + SMB Workgroup: Xác định workgroup mà samba server được cấu hình tham gia.

V.18. Chọn các chương trình và Package cài đặt

Bạn chọn các chương trình cần cài đặt, nếu ta chọn **everything** là cài tất cả các chương trình, chọn **Minimal** là chỉ cài một số chương trình hoặc phần mềm thông dụng.

Nếu bạn nắm rõ các package cần thiết cho các chương trình mình mong muốn thì chọn **Select individual packages**. Ta có thể chọn **Details** để chọn chi tiết các thành phần trong từng phần mềm hoặc nhóm các công cụ.



V.19. Định dạng filesystem và tiến hành cài đặt



VI. Cấu hình thiết bị

VI.1. Bộ nhớ (RAM)

System RAM được BIOS nhận biết khi khởi động, Linux kernel có khả năng nhận biết được tất cả các loại RAM(EDO, DRAM, SDRAM, DDRAM).

VI.2. Vị trí lưu trữ tài nguyên

Để cho phép các thiết bị phần cứng trong máy tính có thể giao tiếp trực tiếp với tài nguyên hệ thống, đặc biệt là CPU thì hệ thống sẽ định vị dưới dạng lines và channels cho mỗi thiết bị như: IRQ(interrupt Request Lines), Input/Output Address and Direct Memory Access channels(DMA).

- IRQ cho phép thiết bị yêu cầu CPU time, IRQ có giá trị từ 0 ->15
- IO address chỉ định địa chỉ trong bộ nhớ, CPU sẽ giao tiếp với thiết bị bằng cách đọc và ghi bộ nhớ trên địa chỉ này.
- DMA cho phép thiết bị truy xuất bộ nhớ hệ thống như ghi và xử lý dữ liệu mà không cần truy xuất CPU.



Kernel lưu trữ thông tin tài nguyên này trong thư mục /proc, các tập tin ta cần quan tâm:

- + /proc/dma
- + /proc/interrupt
- + /proc/ioports
- + /proc/pci

Tuy nhiên ta có thể sử dụng các công cụ lspci, dmesg để có thể xem thông tin IRQ, I/O, DMA...

Thiết bị	I/O port	IRQ
/dev/ttyS0	0x03F8	4
/dev/ttyS0	0x02F8	3
/dev/lp0	0x378	7
/dev/lp1	0x278	5
Soundcard	0x220	
Ethernet card	0x300	10
Ethernet card	0x340	9

Ta có thể cấu hình các thông tin trên bằng cách thay đổi thông tin trong tập tin /etc/modules.conf

VI.3. Hỗ trợ USB

Hầu hết các phiên bản linux sau này có khả năng nhận biết (Detect) USB device, một khi USB được cắm vào USB port thì nó được USB controller điều khiển, Linux hỗ trợ rất nhiều USB controller (ta có thể tham khảo trong tài liệu USB howto), thiết bị USB được Linux kernel nhận biết qua tập tin /dev/sda1

VI.4. Network Card

Kernel của linux hỗ trợ hầu hết NIC, để xem chi tiết thông tin hiện tại của card mạng ta sử dụng các lệnh sau đây: Dmesg, lspci, /proc/interrupts, /sbin/lsmmod, /etc/modules.conf

VI.5. Cài đặt modem

Trong phần này ta tìm hiểu cách cài đặt Serial modem, ta tìm hiểu các serial port được nhận biết trên Linux

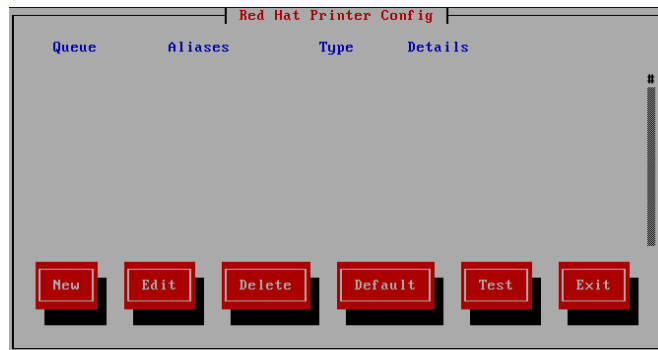
Dos	Linux
COM1	/dev/ttyS0
COM2	/dev/ttyS1
COM3	/dev/ttyS2

Sau đây là một số bước cài đặt serial modem:

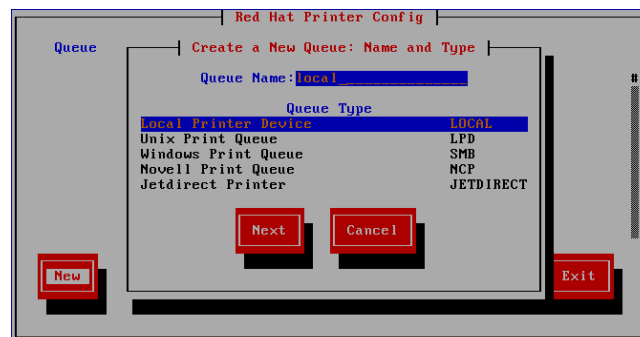
- + Bước 1: Dùng lệnh setserial để scan serial device.
- + Bước 2: Dùng lệnh ls -s /dev/ttyS1 /dev/modem
- + Bước 3: cấu hình Dial profile thông qua công cụ wvdial cung cấp script wvdialconfig để ta scan những thông tin cần thiết cho modem và ghi vào file /etc/wvdial.conf (trong phần này ta chỉ quan tâm về vấn đề cài đặt modem cho nên đây là một bước tham khảo thêm)

VI.6. Cài đặt và cấu hình máy in

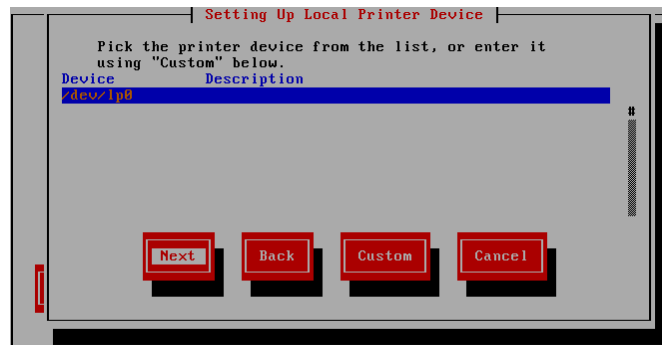
Trước khi cài đặt máy in ta cần cài thêm package system-config-printer-0.6.98-1(Fedora Core). Sau đó ta dùng lệnh #system-config-printer

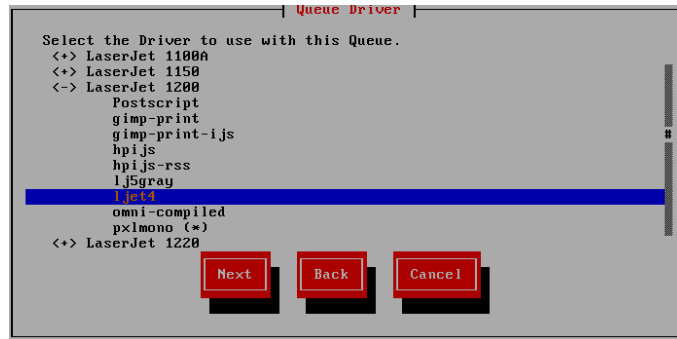


Chọn New để cài đặt máy in

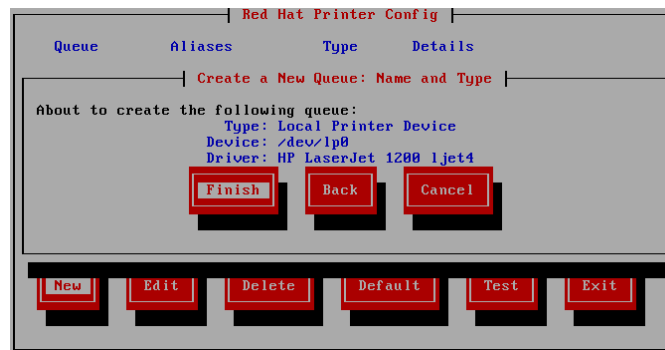


Đặt tên Printer và chọn Queue Type





Chọn **Queue Driver** để chỉ định loại máy in



VII. Sử dụng hệ thống

VII.1. Đăng nhập

Linux là hệ điều hành đa người dùng, tại một thời điểm nhiều người có thể cùng sử dụng hệ thống để làm việc. Mỗi người dùng có một tài khoản trong hệ thống. Tài khoản này dùng để quản lý và phân biệt các người dùng với nhau.

Để sử dụng hệ thống, trước hết bạn phải đăng nhập vào. Khi bạn kết nối tới máy thì màn hình hiển thị dòng

- + Login :
- + Password:

Có 2 dạng dấu nhắc lệnh:

- + Dạng \$ dùng cho người dùng thường.
- + Dạng # dùng cho người dùng quản trị (root).

Khi login vào hệ thống, chúng ta thấy dấu nhắc lệnh xuất hiện có dạng:

[tên-đăng-nhập@tên-máy thư-mục-hiện-hành]dấu-nhắc-lệnh

Ví dụ:

[root@server root]#

- Từ dấu nhắc lệnh ta có thể sử dụng lệnh theo cú pháp như sau: Tên-lệnh [tùy-chọn] [tham-số]



- + Tùy chọn có dạng: <math>-<math>
- + Nếu có nhiều tùy chọn thì ta dùng dấu khoảng trắng để làm dấu ngăn cách hoặc kết hợp nhiều tùy chọn

Ví dụ :

```
[root@server root]#ls -a -l /etc
```

- Linux cho phép chúng ta kết hợp nhiều lựa chọn chỉ dùng một dấu - . Như ví dụ trên ta có thể dùng lệnh ls -al /etc thay cho ls -a -l /etc
- Chuyển sang user khác: Đang làm việc chúng ta có thể chuyển sang người dùng khác mà không phải logout ra. Trong trường hợp này bạn dùng lệnh su.
 - + \$su [tên-user] : chuyển sang user mới
- Nếu tên-user không có thì mặc định là chuyển qua root
- Thông thường khi chúng ta chuyển sang user khác thì biến môi trường của hệ thống vẫn giữ nguyên theo user cũ. Để sử dụng biến môi trường của user mới chúng ta dùng thêm tham số - trong lệnh su.

Ví dụ: #su - [user]

VII.2. Một số lệnh cơ bản

Tên lệnh	Ý nghĩa
date	Hiển thị ngày giờ hệ thống
who	Cho biết các người dùng đang đăng nhập vào hệ thống
tty	Xác định tập tin tty mà mình đang login vào.
cal	Lịch
finger	Hiển thị các thông tin của các người dùng như họ tên, địa chỉ ...
chfn	Thay đổi thông tin của người dùng
head	Xem nội dung tập tin từ đầu tập tin
tail	Xem nội dung từ cuối tập tin
hostname	Xem, đổi tên máy
passwd	Đổi mật khẩu cho user

VII.3. Sử dụng trợ giúp man

Trong MS DOS để biết cú pháp hay ý nghĩa của một lệnh chúng ta hay dùng giúp đỡ của lệnh bằng cách đánh tham số /? vào phía sau lệnh, còn Windows có bộ Help cho phép bạn tìm kiếm các thông tin liên quan đến một vấn đề nào đó. Linux cung cấp cho bạn một hệ thống thư viện giúp đỡ bạn tìm các thông tin theo từ khóa bạn nhập vào. Dù không có giao diện bằng Window, nhưng các tài liệu giúp đỡ này rất có ích đối với người sử dụng đặc biệt khi sử dụng các lệnh. Các bạn sẽ biết các lệnh trong Linux sử dụng rất nhiều tùy chọn mà chúng ta không thể nhớ hết được, Linux cung cấp trình trợ giúp man

```
$man [từ-khóa]
```

Ví dụ: Tìm kiếm các thông tin về lệnh ls

```
$man ls
```

Bạn dùng phép điều khiển lên, xuống để xem trang man. Nếu muốn xem từng trang dùng phím space. Để thoát khỏi man: chọn phím q



Man phân dữ liệu mình lưu trữ thành những đoạn (session) khác nhau với các chủ đề khác nhau là

Session	Tên chủ đề	Ý nghĩa
1	User command	các lệnh thông thường của hệ điều hành
2	system call	các hàm thư viện kernel của hệ thống
3	subroutines	các hàm thư viện lập trình
4	devices	các hàm truy xuất tập tin và xử lý thiết bị
5	File format	các hàm định dạng tập tin
6	games	các hàm liên quan đến trò chơi
7	Miscell	các hàm khác
8	Sys. admin	các hàm quản trị hệ thống

Xác định cụ thể thông tin của một chủ đề nào, chúng ta dùng lệnh man như sau:

\$man [session] [từ-khóa]

Ví dụ : man 3 printf :Xem các thông tin về hàm printf dùng trong lập trình

Nếu chúng ta không xác định session thì session mặc nhiên là 1

VIII. Khởi động hệ thống

VIII.1.Các bước khởi động hệ thống:

- **Bước 1:** Khi một máy PC bắt đầu khởi động, bộ vi xử lý sẽ tìm đến cuối vùng bộ nhớ hệ thống của BIOS và thực hiện các chỉ thị ở đó.
- **Bước 2:** BIOS sẽ kiểm tra hệ thống, tìm và kiểm tra các thiết bị và tìm kiếm đĩa chứa trình khởi động. Thông thường, BIOS sẽ kiểm tra ổ đĩa mềm, hoặc CDROM xem có thể khởi động từ chúng được không, rồi đến đĩa cứng. Thứ tự của việc kiểm tra các ổ đĩa phụ thuộc vào các cấu hình trong BIOS.
- **Bước 3:** Khi kiểm tra ổ đĩa cứng, BIOS sẽ tìm đến MBR và nạp vào vùng nhớ hoạt động chuyển quyền điều khiển cho nó.
- **Bước 4:** MBR chứa các chỉ dẫn cho biết cách nạp trình quản lý khởi động GRUB/LILO cho Linux hay NTLDR cho Windows NT/2000. MBR sau khi nạp trình quản lý khởi động, sẽ chuyển quyền điều khiển cho trình quản lý khởi động.
- **Bước 5:** Boot loader tìm kiếm boot partition và đọc thông tin cấu hình trong file grub.conf hoặc lilo.conf và hiển thị Operating Systems kernel có sẵn trong hệ thống để cho phép chúng ta lựa chọn OS kernel boot.

Ví dụ về grub.conf

```
default=0
timeout=10
splashimage=(hd0,0)/grub/splash.xpm.gz
title Fedora Core (2.6.8-1.521)
    root (hd0,0)
    kernel /vmlinuz-2.6.8-1.521 ro root=LABEL=/
    initrd /initrd-2.6.8-1.521.img
title Windows 2000
    rootnoverify (hd0,1)
```



chainloader +1

- **Bước 6:** Sau khi chọn kernel boot trong file cấu hình của boot loader, hệ thống tự động load chương trình /sbin/init để số kiểm tra hệ thống tập tin (file system check) sau đó đọc file /etc/inittab để xác định mức hoạt động(runlevel). Các **Linux runlevel**

Mode/runlevel	Thư mục lưu script file(Directory)	Mô tả mode hoạt động
0	/etc/rc.d/rc0.d	Là mức shutdown hệ thống(halt)
1	/etc/rc.d/rc1.d	Chỉ dành cho một người dùng thường dùng để sửa lỗi hệ thống tập tin.(còn gọi là single user mode)
2	/etc/rc.d/rc2.d	Không sử dụng(user-definable)
3	/etc/rc.d/rc3.d	Sử dụng cho nhiều người dùng nhưng chỉ giao tiếp dưới dạng Text(Full multi-user mode no GUI interface)
4	/etc/rc.d/rc4.d	Không sử dụng(user-definable)
5	/etc/rc.d/rc5.d	Sử dụng cho nhiều người dùng và có thể cung cấp giao tiếp đồ họa.(Full multiuser mode)
6	/etc/rc.d/rc6.d	Mức reboot hệ thống

- **Bước 7:** Sau khi xác định runlevel(thông qua biến initdefault), chương trình /sbin/init sẽ thực thi các file startup script được đặt trong các thư mục con của thư mục /etc/rc.d. Script sử dụng runlevel 0->6 để xác định thư mục chứa file script chỉ định cho từng runlevel như: /etc/rc.d/rc0.d -> /etc/rc.d/rc6.d. Ta tham khảo một số file script trong thư mục /etc/rc.d/rc3.d/

```

K01yum      K50snmptd  S09isdn    S40snortd
S90mysql
K05innd     K50tux     S10network S44acpid
S90xfs
K05saslauthd K50vsftpd  S12syslog  S55cups
S95anacron
K15postgres K54dovecot S13irqbalance
S55named    S95atd
K20nfs      K70aep1000 S13portmap S55sshd
S97messagebus
K24irda     K70bcm5820 S14nfslock
S56rawdevices S97rhnsd
K25squid    K74ntpd    S20random  S56xinetd
S99local
K34yppasswdd K74ypserv  S24pcmcia
S78mysqld   S99webmin
K35smb      K74ypxfrd  S25netfs   S80sendmail
K35vncserver K92iptables S26apmd    S85gpm
K35winbind  S00microcode_ctl S28autofs
S85httpd
K50snmpd    S05kudzu   S40smartd  S90crond
    
```




Ta cần lưu ý tên tập tin bắt đầu bằng từ khóa “S” có nghĩa rằng tập tin này sẽ được thực thi lúc khởi động hệ thống, ngược lại tập tin bắt đầu bằng từ khóa “K” nghĩa rằng tập tin đó được thực thi khi hệ thống shutdown, số theo sau các từ khóa “S” và “K” để chỉ định trình tự khởi động các script, kế tiếp là tên file script cho từng dịch vụ .

- **Bước 8:** Nếu như ở bước 4 runlevel 3 được chọn lựa thì hệ thống sẽ chạy chương trình login để yêu cầu đăng nhập cho từng user trước khi sử dụng hệ thống, nếu runlevel 5 được chọn lựa thì hệ thống sẽ load X terminal GUI application để yêu cầu đăng nhập cho từng user.

IX. Shutdown và Reboot hệ thống

- Để shutdown hệ thống ta thực hiện một trong các cách sau:
 - + [root@server root]# init 0
 - + [root@server root]# shutdown -hy t (shutdown hệ thống sau khoảng thời gian t giây)
 - + [root@server root]# halt
 - + [root@server root]# poweroff
- Để reboot hệ thống ta có thể thực hiện một trong các cách sau:
 - + [root@server root]# init 6
 - + [root@server root]# reboot
 - + [root@server root]# shutdown -ry 10 (chỉ định 10 phút sau hệ thống sẽ reboot hệ thống)

X. Sử dụng runlevel

Chuyển đổi runlevel: Runlevel được hiểu là các mức hoạt động của hệ thống, để chuyển đổi các mức hoạt động này ta dùng lệnh **init #runlevel_number**. Ví dụ ta muốn chuyển sang mức 1 ta dùng lệnh **init 1** lúc này dấu nhắc shell của hệ thống ở dạng **bash-2.05b#**, ta có thể dùng lệnh **startx** để chuyển sang **runlevel 5** (tương đương với lệnh init 5). Đặt runlevel mặc định cho hệ thống ta dùng trình tiện ích mc để hiệu chỉnh thông số runlevel X(0->6)

id:X:initdefault:

XI. Phục hồi mật khẩu cho user quản trị

Trong trường hợp ta để mất mật khẩu của user quản trị(root user), lúc có nhiều cách để phục hồi mật khẩu cho user này:

- + Ta có thể dùng lệnh đĩa mềm khởi động (ta có thể dùng lệnh mkbootdisk hoặc dd để tạo đĩa này,...)
- + Dựa vào boot loader LILO hoặc GRUB(ta chỉ sử dụng cách này trong trường hợp ta có thể edit được boot loader khi khởi động, nếu không ta phải dùng cách 1)

Ta thực hiện điển hình cách 2(dựa vào grub boot loader) như sau:



- + Khởi động máy.
- + Khi GRUB Screen hiển thị ta chọn phím e để edit boot loader(nếu ta có đặt mật khẩu cho GRUB thì nhập mật khẩu vào).

```
GRUB version 0.93 (638K lower / 63424K upper memory)

root (hd0,0)
kernel /boot/vmlinuz-2.4.22-1.2115.nptl ro root=LABEL=/ rhgb
initrd /boot/initrd-2.4.22-1.2115.nptl.img
```

- + Chọn mục kernel /boot.... Sau đó bấm phím e để edit mục này và thêm từ khóa -s để vào runlevel 1 sau đó bấm phím enter

```
grub edit> kernel /boot/vmlinuz-2.4.22-1.2115.nptl ro root=LABEL=/ rhgb -s
```

- + Sau khi thực thi bước 3 ta bấm phím b để boot hệ thống vào runlevel 1 và thực hiện lệnh passwd để thay đổi mật khẩu của user root.

```
sh-2.05b# passwd
Changing password for user root.
New password:
BAD PASSWORD: it is too simplistic/systematic
Retype new password:
passwd: all authentication tokens updated successfully.
sh-2.05b# _
```

- + dùng lệnh init 6 để reboot hệ thống.

XII. Tìm hiểu boot loader

XII.1. GRUB boot loader

XII.1.1 Tổng quan

GRUB là trình khởi động máy tính, nó có nhiệm vụ tải nhân và khởi động hệ thống Linux cũng như một số hệ điều hành khác: FreeBSD, NetBSD, OpenBSD, GNU HURD, DOS, Windows 95, 98, Me, NT, 2000 và XP...

Năm 1995, Erich Boley thiết kế GRUB. Năm 1999, Gordon Matzigkeit và Yoshinori K. Okuji kế thừa GRUB thành gói phần mềm GNU chính thức.

- GRUB hỗ trợ nhiều hệ điều hành bằng cách khởi động trực tiếp nhân hệ điều hành hoặc bằng cách nạp chuỗi (chain-loading).
- GRUB hỗ trợ nhiều hệ thống tập tin: BSD FFS, DOS FAT16 và FAT32, Minix fs, Linux ext2fs và ext3fs, ReiserFS, JSF, XFS, và VSTa fs.
- GRUB cung cấp giao diện dòng lệnh linh hoạt lẫn giao diện thực đơn, đồng thời cũng hỗ trợ tập tin cấu hình.



XII.1.2 Tập tin cấu hình

Đoạn thứ nhất: mô tả các chỉ thị tổng quát như:

- + Hệ điều hành mặc định (default)
- + Thời gian chờ đợi người dùng nhập dữ liệu trước khi thực hiện lệnh mặc định (timeout=10), tính bằng giây.
- + Ta cũng có thể chọn màu để hiển thị trình đơn (color green/black light-gray/blue)

Đoạn thứ hai: cho biết các thông số để khởi động hệ Linux:

- + Tiêu đề trên trình đơn là Red Hat Linux (title)
- + Hệ điều hành này sẽ khởi động từ partition đầu tiên của ổ đĩa thứ nhất – / (hda0,0:ổ đĩa thứ nhất, partition thứ nhất). Và cần phải mount partition này trước.
- + Tập tin vmlinuz đang được chứa trong thư mục root và filesystem root đang nằm trên partition thứ năm của đĩa cứng thứ nhất (/dev/hdc5)
- + Dòng lệnh boot nhắc phải nạp ngay hệ điều hành đã được khai báo ở trên.

Đoạn thứ ba: cho biết các thông số về hệ điều hành thứ hai đang được cài đặt trong hệ thống.

- + Tiêu đề là Windows
- + Hệ điều hành đang chiếm partition thứ nhất của ổ đĩa thứ hai (hda1,0). Có điều với lệnh rootnoverify, GRUB không cần chú ý kiểm tra xem partition này có được mount hay không.
- + Cấu lệnh chainloader + 1 đã sử dụng +1 làm tên tập tin cần khởi động như một mào xích trong tiến trình: +1 có nghĩa là sector thứ nhất của partition đang xét
- + Bạn có thể dùng lệnh man grub.conf để tìm hiểu thêm về tập tin cấu hình này.
- + Lưu ý: Từ GRUB muốn chuyển sang LILO thực hiện các bước sau:
- + Trong thư mục /etc có tập tin lilo.conf.anaconda. Từ tập tin này copy thành tập tin lilo.conf
- + Thực thi lệnh lilo

XII.1.3 Bảo mật cho GRUB

Dùng tính năng mật khẩu của GRUB để chỉ cho phép người quản trị dùng các hoạt động tương tác (như biên tập đề mục thực đơn và vào giao diện dòng lệnh). Để sử dụng tính năng này, cần chạy lệnh password trong tập tin cấu hình: password --md5 <PASSWORD>

Khi đó GRUB không cho phép điều khiển tương tác nào (<e> và <c>), cho đến khi gõ phím <p> và nhập đúng mật khẩu. Tùy chọn --md5 cho GRUB biết rằng PASSWORD ở định dạng MD5. Nếu không sử dụng tùy chọn này, GRUB cho rằng PASSWORD ở dạng văn bản thuần túy.

XII.1.4 Khởi động GRUB từ ntldr

- Cài GRUB lên sector khởi động của một phân vùng (chẳng hạn như /boot, ở /dev/hda2).
- Chép sector khởi động đó vào đĩa mềm hoặc một hệ thống tập tin trên đĩa cứng, thí dụ cho đĩa mềm (sau khi đã được gán tại /mnt/floppy):

```
dd if=/dev/hda2 of=/mnt/floppy/bootsect.lnx bs=512 count=1
```

- Tên của tập tin bootsect.lnx phải theo quy định 8.3 để ntldr có thể nhận diện được.



- Khởi động lại Windows và chép tập tin bootsect.lnx vào thư mục gốc trên đĩa C:
- Thay đổi thuộc tính chỉ-đọc của tập tin C:\boot.ini, nếu cần, bằng Windows Explorer hoặc bằng dòng lệnh (C:\attrib -s -r c:\boot.ini, và sau khi thực hiện xong C:\attrib +s +r c:\boot.ini). Mở tập tin boot.ini bằng một trình biên tập, chẳng hạn như Notepad, thêm dòng c:\bootsect.lnx="Linux" vào tập tin đó.

```
[bootloader]
timeout=5
default=multi(0)disk(0)rdisk(0)partition(1)\WINNT
[operatingsystems]
multi(0)disk(0)rdisk(0)partition(1)\WINNT="Microsoft Windows 2000 Professional"/fastdetect
c:\bootsect.lnx="Linux"
```

Mỗi khi thay đổi sector khởi động dùng để tạo bootsect.lnx, cần phải cập nhật bản mới của tập tin này.

XII.2. LILO boot loader

LILO là một boot manager nằm trọn gói chung với các bản phát hành Red Hat, và là boot manager mặc định cho Red Hat 7.1 trở về trước.

XII.2.1 Thiết lập cấu hình LILO.

LILO đọc thông tin chứa trong tập tin cấu hình /etc/lilo.conf để biết xem hệ thống máy bạn có những hệ điều hành nào, và các thông tin khởi động nằm ở đâu. LILO được lập cấu hình để khởi động một đoạn thông tin trong tập tin /etc/lilo.conf cho từng hệ điều hành. Sau đây là ví dụ về tập tin /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/initrd-2.4.0-0.43.6.img
- Read-only
- Root=/dev/hda5

Đoạn 3:

- Other=/dev/hda1



- Label=dos

Đoạn thứ nhất:

- + Cho biết LILO cần xem xét vào MBR (boot=/dev/hda1)
- + Kiểm tra tập tin map
- + Nó còn cho biết LILO có thể cài đặt một tập tin đặc biệt (/boot/boot.b) như là một sector khởi động mới
- + Thời gian chờ trước khi nạp hệ điều hành mặc định (default=xxx) được khai báo thông qua dòng timeout=50 (5 giây) – thời gian tính bằng 1/10 của giá trị”.
- + Nạp thông tin trong quá trình khởi động từ tập tin /boot/message
- + Dòng LBA32 cho biết cấu hình của đĩa cứng: cho biết đĩa cứng của bạn hỗ trợ LBA32, thông thường dòng này có giá trị linear (bạn không nên đổi lại dòng này nếu bạn không hiểu rõ ổ đĩa cứng của bạn, bạn có thể tìm hiểu đĩa cứng của bạn có hỗ trợ LBA32 hay không bằng cách xem trong BIOS)

Đoạn thứ hai:

- + Cung cấp thông tin khởi động cho hệ điều hành linux
- + Dòng image báo cho LILO biết vị trí của kernel Linux
- + Dòng label hiện diện ở cả 2 đoạn cho biết tên của hệ điều hành nào sẽ xuất hiện tại trình đơn khởi động của LILO.
- + Dòng root xác định vị trí root file system của Linux

Đoạn thứ ba: Dòng other cho biết partition của một hệ điều hành nữa đang ở hda1 của ổ đĩa cứng.

Lưu ý: Từ LILO muốn chuyển sang GRUB thực hiện cài đặt như sau:

```
#!/sbin/grub-install [tên_ổ_đĩa]
```

Ví dụ: #/sbin/grub-install /dev/had



BÀI 3 Hệ Thống Tập Tin

Tóm tắt

Lý thuyết: 8 tiết - Thực hành: 10 tiết.

Mục tiêu	Các mục chính	Bài tập bắt buộc	Bài tập làm thêm
Bài học này giới thiệu các khái niệm cơ bản về hệ thống tập tin, cấu trúc hệ thống tập tin, các loại tập tin được hỗ trợ trên linux, cách tạo và quản lý các hệ thống tập tin, sử dụng các lệnh liên quan đến hệ thống tập tin, thư mục.	<ol style="list-style-type: none"> I. Cấu trúc hệ thống tập tin. II. Cấu trúc cây thư mục. III. Các thao tác trên hệ thống tập tin và đĩa. IV. Các thao tác trên tập tin và thư mục. V. Lưu trữ tập tin/thư mục. VI. Bảo mật hệ thống tập tin. 	Bài tập 3.1 (sách bài tập - Hệ thống tập tin)	



I. Cấu trúc hệ thống tập tin

- Mỗi hệ điều hành có cách tổ chức lưu trữ dữ liệu riêng. Ở mức vật lý, đĩa được định dạng từ các thành phần sector, track, cylinder. Ở mức logic, mỗi hệ thống sử dụng cấu trúc riêng, có thể dùng chỉ mục hay phân cấp để có thể xác định được dữ liệu từ mức logic tới mức vật lý. Cách tổ chức như vậy gọi là hệ thống tập tin (file system).
- Chẳng hạn như Windows sử dụng hệ thống tập tin FAT16, FAT32, WinNT sử dụng NTFS để tăng cường bảo mật hệ thống tập tin.
- Hệ thống tập tin là một phần cơ bản của hệ điều hành Linux.
- Một hệ thống tập tin là thiết bị mà nó đã được định dạng để lưu trữ tập tin và thư mục.
- Hệ thống tập tin Linux bao gồm: đĩa mềm, CD-ROM, những partition của đĩa cứng. Những hệ thống tập tin thường được tạo trong quá trình cài đặt hệ điều hành. Nhưng bạn cũng có thể thay đổi cấu trúc hệ thống tập tin khi thêm thiết bị hay chỉnh sửa những partition đã tồn tại. Như vậy, việc biết và hiểu cấu trúc hệ thống tập tin trong Linux thật là quan trọng.
- Linux hỗ trợ rất nhiều loại hệ thống tập tin như: ext2, ext3, MS-DOS, proc. Hệ thống tập tin cơ bản của Linux là ext2 và ext3 (hiện tại là ext3). Hệ thống tập tin này cho phép đặt tên tập tin tối đa 256 ký tự và kích thước tối đa là 4terabytes. MS-DOS dùng để truy cập trực tiếp những tập tin MS-DOS. Bên cạnh đó, Linux còn hỗ trợ vfat cho phép đặt tên tập tin dài đối với những tập tin MS-DOS và những partition FAT32. Proc là một hệ thống tập tin ảo (/proc) nghĩa là không dành dung lượng đĩa phân phối cho nó. Ngoài ra còn có những hệ thống tập tin khác như iso9660, UMSDOS, Network File System (NFS).
- Các thành phần của hệ thống tập tin:
 - + Superblock
 - + Inode
 - + Storageblock

Super Block: là một cấu trúc được tạo tại vị trí bắt đầu hệ thống tập tin. Nó lưu trữ thông tin về hệ thống tập tin như: Thông tin về block-size, free block, thời gian gắn kết(mount) cuối cùng của tập tin

Inode (256 byte): Lưu những thông tin về những tập tin và thư mục được tạo ra trong hệ thống tập tin. Nhưng chúng không lưu tên tập tin và thư mục thực sự. Mỗi tập tin tạo ra sẽ được phân bổ một inode lưu thông tin sau:

- + Loại tập tin và quyền hạn truy cập tập tin
- + Người sở hữu tập tin.
- + Kích thước của tập tin và số hard link đến tập tin.
- + Ngày và thời gian chỉnh sửa tập tin lần cuối cùng.
- + Vị trí lưu nội dung tập tin trong hệ thống tập tin.

Storageblock: Là vùng lưu dữ liệu thực sự của tập tin và thư mục. Nó chia thành những Data Block. Dữ liệu lưu trữ vào đĩa trong các data block. Mỗi block thường chứa 1024 byte. Ngay khi tập tin chỉ có 1 ký tự thì cũng phải cấp phát 1 block để lưu nó. Không có ký tự kết thúc tập tin.



- + Data Block của tập tin thông thường lưu inode của tập tin và nội dung của tập tin
- + Data Block của thư mục lưu danh sách những entry bao gồm inode number, tên của tập tin và những thư mục con.

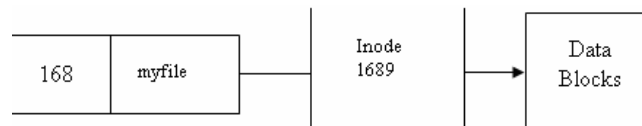
1.1. Loại tập tin.

Trong linux tập tin dùng cho việc lưu trữ dữ liệu. Nó bao gồm cả thư mục và các thiết bị lưu trữ. Một tập tin dữ liệu, hay một thư mục đều được xem là tập tin. Khái niệm tập tin còn mở rộng dùng cho các thiết bị như máy in, đĩa cứng ... ngay cả bộ nhớ chính cũng được coi như là một tập tin, các tập tin trong linux được chia ra làm 3 loại chính:

- Tập tin chứa dữ liệu bình thường
- Thư mục
- Tập tin thiết bị

Tập tin dữ liệu: Đây là tập tin theo định nghĩa truyền thống, nó là dữ liệu lưu trữ trên các thiết bị lưu trữ như đĩa cứng, CD-ROM ... Bạn có thể đưa bất cứ dữ liệu nào vào tập tin này như đoạn source chương trình, tập tin văn bản hay tập tin thực thi dạng mã máy, các lệnh của Linux cũng như tất cả các tập tin được tạo ra bởi người dùng.

Tập tin thư mục: Thư mục không chứa dữ liệu, mà chỉ chứa các thông tin của những tập tin và thư mục con trong nó. Thư mục chứa hai trường của một tập tin là tên tập tin và inode number.



Tập tin thiết bị: Hệ thống Unix và Linux xem các thiết bị như là các tập tin. Ra vào dữ liệu trên các tập tin này chính là ra vào dữ liệu cho thiết bị. Ví dụ khi chúng ta muốn chép dữ liệu ra ổ đĩa A: thì sẽ chép vào tập tin /dev/fd0 hoặc khi chúng ta thực hiện việc in thì dữ liệu vào máy in được đưa vào tập tin tương ứng cho máy in.

1.2. Liên kết tập tin

Link (Liên kết) một liên kết, hiểu theo cách đơn giản nhất, là tạo ra một tên tập tin thứ hai cho một tập tin. Ví dụ, bạn có một tập tin /usr/lib/testfile và muốn có một tập tin giống như vậy trong thư mục /usr/tim, bạn không cần phải copy nó mà chỉ cần tạo một liên kết với lệnh sau:

```
#ln /usr/bill/testfile /usr/tim/testfile
```

Cú pháp của lệnh ln:

```
$ln <nguồn> <đích>
```

Lý do cơ bản của việc tạo liên kết là nhân tập tin lên nhiều lần. Trong ví dụ trên, cả hai tập tin chính là một. Do đó, nếu có bất kỳ sự thay đổi nào trên một tập tin sẽ ảnh hưởng ngay đến tập tin còn lại.

Hard Link: là một liên kết trong cùng hệ thống tập tin với hai inode entry tương ứng trở đến cùng một nội dung vật lý (cùng inode number vì chúng trở đến cùng dữ liệu). Nếu bạn muốn thấy điều này, dùng lệnh sau:



```
$ ls -i testfile
```

```
14253 testfile
```

Sau đó tạo một liên kết có một tên khác và hiển thị thông tin của inode entry.

```
$ ln testfile test2
```

```
$ ls -i testfile test2
```

```
14253 testfile 14253 test2
```

Cả hai tập tin đều có inode number giống nhau

Symbolic Link: Là một liên kết khác mà không sử dụng inode entry cho việc liên kết. Bạn sử dụng liên kết này khi muốn tạo ra những driver thiết bị, như /dev/modem thay cho /dev/cua1. Tùy chọn -s của lệnh ln cho phép tạo ra một symbolic link.

Ví dụ:

```
$ ls -i bigfile
```

```
6253 bigfile
```

```
$ ln -s bigfile anotherfile
```

```
$ ls -i bigfile anotherfile
```

```
6253 bigfile 8358 anotherfile
```

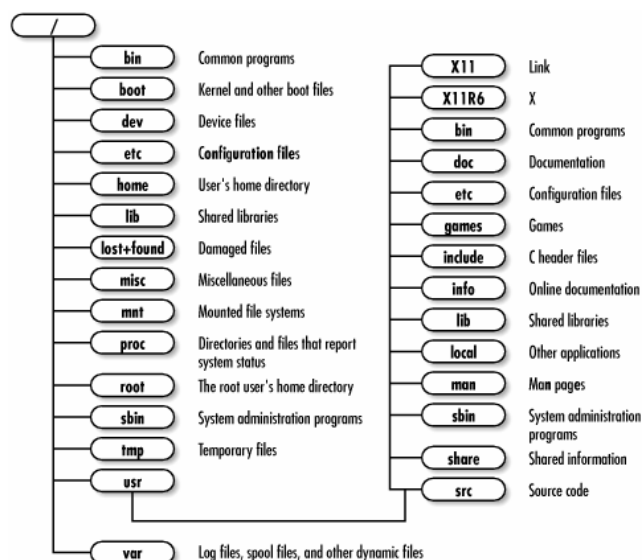
Như bạn thấy, nội dung inode number của các tập tin khác nhau. Liệt kê một thư mục sẽ thấy symbolic link:

```
lrwxrwxrwx 1 root root 6 Sep 16:35 anotherfile -> bigfile
```

```
-rw-rw-r-- 1 root root 2 Sep 17:23 bigfile
```

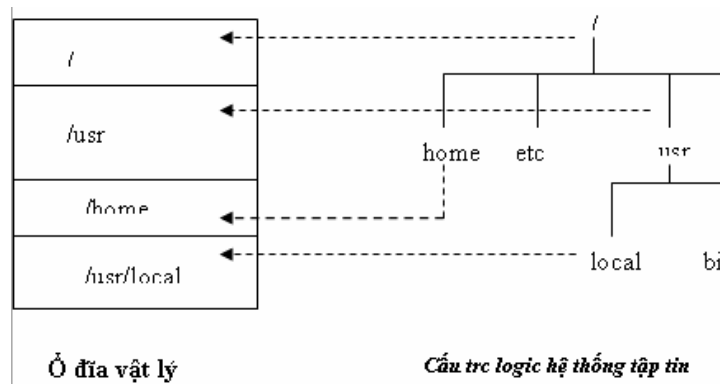
Lưu ý: khi xóa tập tin gốc, nội dung của tập tin hard link không bị ảnh hưởng nhưng nội dung tập tin symbolic link không xem được.

II. Cấu trúc cây thư mục





Hệ thống tập tin Linux có cấu trúc như hình vẽ trên. Trong Linux không có khái niệm ổ đĩa như trong Windows, tất cả các tập tin thư mục bắt đầu từ thư mục gốc (/). Linux sử dụng dấu "." chỉ thư mục hiện hành và dấu ".." chỉ thư mục cha của thư mục hiện hành.



Như hình vẽ trên thư mục gốc được mount vào partition thứ nhất, /usr được mount vào partition thứ 2... Những dữ liệu ghi vào thư mục /home sẽ ghi vào partition thứ 3. Tương tự, dữ liệu của thư mục /usr/local ghi vào partition 4, dữ liệu của thư mục /usr không phải thư mục con /usr/local thì ghi vào partition 2.

Linux sử dụng các tập tin chỉ đến các partition trên ổ đĩa vật lý. Những tập tin này là những tập tin thiết bị, nằm trong thư mục /dev. Tập các tập tin này có dạng đầu tin là ký tự xác định loại ổ đĩa như: đĩa mềm là fd, đĩa cứng là hd, đĩa scsi là sd ... tiếp theo là số thứ tự ổ đĩa: Ổ đĩa thứ nhất dùng ký hiệu a, thứ 2 ký hiệu là b ... và sau cùng là số thứ tự partition.

Ví dụ: tập tin chỉ đến các thiết bị :

- + ổ mềm thứ nhất : /dev/fd0
- + partition thứ nhất của ổ đĩa cứng đầu tin : /dev/hda1
- + partition thứ 3 của đĩa cứng thứ 2 : /dev/hdb3.

Các thư mục cơ bản trên Linux

Thư mục	Chức năng
/bin, /sbin	Chứa các tập tin nhị phân hỗ trợ cho việc boot và thực thi các lệnh cần thiết.
/boot	Chứa linux kernel, file ảnh hỗ trợ load hệ điều hành
/lib	Chứa các thư viện chia sẻ cho các tập tin nhị phân trong thư mục /bin và /sbin, chứa kernel module.
/usr/local	Chứa các thư viện, các phần mềm để chia sẻ cho các máy khác trong mạng.
/tmp	Chứa các file tạm
/dev	Chứa các tập tin thiết bị (như CDROM, floppy), và một số file đặc biệt khác.
/etc	Chứa các tập tin cấu hình hệ thống
/home	Chứa các thư mục lưu trữ home directory của người dùng
/root	Lưu trữ home directory cho user root
/usr	Lưu trữ tập tin của các chương trình đã được cài đặt trong hệ thống.
/var	Lưu trữ log file, hàng đợi của các chương trình ứng dụng, mailbox của người dùng.
/mnt	Chứa các mount point của các thiết bị được mount vào trong hệ thống.



```
/proc | Lưu trữ thông tin về kernel
```

Các thư mục có thể sử dụng làm mount point cho các thiết bị riêng: như: /boot, /home, /root, /tmp, /usr, /usr/local, /opt, /var.

III. Các thao tác trên hệ thống tập tin và đĩa

III.1. Mount và umount một hệ thống tập tin

Muốn mount một hệ thống tập tin vào cây thư mục, bạn phải có một partition vật lý như CD-ROM, đĩa mềm... Và một điều kiện nữa là thư mục mà bạn muốn mount (mount point) vào phải là thư mục có thật. Nó phải có trước khi mount một hệ thống tập tin.

Lưu ý: muốn biết thư mục hiện hành đang ở hệ thống tập tin nào, bạn dùng lệnh `df`. Lệnh này sẽ hiển thị hệ thống tập tin và khoảng trống còn lại trên đĩa.

III.1.1 Mount hệ thống tập tin có tính tương tác

Để mount một hệ thống tập tin, bạn dùng lệnh mount theo cú pháp sau:

```
#mount <tên-thiết-bị> <điểm-mount>
```

Trong đó: Tên-thiết-bị: là thiết bị vật lý như /dev/cdrom (CD-ROM), /dev/fd0 (đĩa mềm), /dev/hda1 ...
điểm-mount: là vị trí thư mục, trong cây thư mục, mà bạn muốn mount vào

Một số tùy chọn của lệnh mount:

- + -f: làm cho tất cả mọi thứ đều hiện ra như thật, song nó chỉ gây ra động tác giả.
- + -v: chế độ chi tiết, cung cấp thêm thông tin về những gì mount định thực hiện.
- + -w: mount hệ thống tập tin với quyền đọc và ghi.
- + -r: mount hệ thống tập tin chỉ có quyền đọc mà thôi.
- + -t loại: xác định lại hệ thống tập tin đang được mount. Những loại hợp lệ là minux, ext2, ext3, msdos, hpfs, proc, nfs, umsdos, iso9660, vfat.
- + -a: mount tất cả những hệ thống tập tin được khai báo trong /etc/fstab.
- + -o remount <fs> chỉ định việc mount lại 1 filesystem nào đó.

Ví dụ:

mount cdrom:

```
#mount /dev/cdrom
```

mount một hệ thống tập tin:

```
#mount /dev/hda6 /usr
```

remount filesystem.

```
#mount -o remount /home
```



III.1.2 Mount một hệ thống tập tin khi khởi động

Một khi đã làm việc ổn định, thường thì Linux sử dụng một số hệ thống tập tin hay dùng và ít khi thay đổi. Do đó, bạn có thể xác định danh sách các hệ thống tập tin nào Linux cần phải mount khi khởi động và cần phải umount khi đóng tắt. Các hệ thống tập tin này được liệt kê trong tập tin cấu hình /etc/fstab.

Tập tin /etc/fstab liệt kê các hệ thống tập tin cần được mount theo từng dòng, mỗi dòng một hệ thống tập tin. Những trường trong mỗi dòng phân cách nhau bằng khoảng trống hoặc khoảng tab.

Các field	Mô tả
Hệ thống tập tin	Xác định thiết bị hoặc hệ thống tập tin cần mount
Mount point	Xác định điểm mount cho hệ thống tập tin. Đối với các hệ thống tập tin đặc biệt như swap, bạn dùng chữ none, có tác dụng làm cho tập tin swap hoạt động như nhìn vào cây thư mục không thấy.
Type	Chỉ ra loại hệ thống tập tin như msdos, vfat, iso9660, ext2...
Mount options	Danh sách các tùy chọn được ngăn cách nhau bởi dấu phẩy
Dump frequency	Xác định khoảng thời gian để lệnh dump sao chép (backup) hệ thống tập tin. Nếu trường này trống, dump sẽ giả định rằng hệ thống tập tin này không cần backup.
Pass number	Khai báo cho lệnh fsck biết thứ tự kiểm tra các hệ thống tập tin khi khởi động hệ thống. Hệ thống tập tin gốc (/) phải có giá trị 1. Tất cả hệ thống tập tin khác phải có giá trị 2. Nếu không khai báo, khi khởi động, máy sẽ không kiểm tra tính nhất thống của hệ thống tập tin.

Như vậy, khi muốn mount các hệ thống tập tin lúc khởi động, bạn nên sử dụng tập tin /etc/fstab thay vì dùng lệnh mount.

Sau đây là ví dụ về tập tin /etc/fstab:

```

LABEL=/                /                ext3 defaults 1 1
LABEL=/boot            /boot            ext3 defaults 1 2
none                   /dev/pts         devpts gid=5,mode=620 0 0
none                   /dev/shm         tmpfs defaults 0 0
none                   /proc            proc defaults 0 0
none                   /sys             sysfs defaults 0 0
/dev/sda3              swap             swap defaults 0 0
/dev/cdrom             /mnt/cdrom       udf,iso9660 noauto,owner,kudzu,ro 0 0
/dev/fd0               /mnt/floppy      auto noauto,owner,kudzu 0 0
//nhon/data           /mnt/smb         smbfs credentials=/etc/cred 0 0
~
    
```



III.1.3 Umount một hệ thống tập tin

Sau khi làm quen với việc gắn những hệ thống tập tin vào cây thư mục Linux. Kế đến, bạn có thể tháo một hệ thống tập bằng lệnh umount. Bạn cần umount một hệ thống tập tin vì nhiều lý do như: kiểm tra hay sửa chữa hệ thống tập tin với lệnh fsck; khi gặp vấn đề về mạng; umount đĩa mềm hay CD-ROM...Lệnh umount có 3 dạng:

- + #umount thiết-bị <điểm-mount>
- + #umount -a
- + #umount -t loại-fs

Lưu ý : lệnh umount không umount những hệ thống tập tin đang sử dụng.

Ví dụ:

```
#cd /mnt
```

```
#umount /mnt
```

Lúc này máy sẽ báo lỗi là hệ thống tập tin đang bận(busy). Do đó, muốn umount /mnt bạn phải di chuyển đến một thư mục khác và một hệ thống tập tin khác

III.2. Định dạng filesystem

Dùng lệnh mkfs để định dạng cho mọi hệ thống tập tin(ext2, ext3,...)

Cú pháp lệnh:

```
#mkfs -t <fstype> <filesystem>
```

Ví dụ: mkfs -t ext2 /dev/hda1 (tương đương với lệnh mkfs.ext2 /dev/hda1)

III.3. Quản lý dung lượng đĩa

Để quản lý và theo dõi dung lượng đĩa ta có thể sử dụng nhiều cách khác nhau, thông thường ta dùng hai lệnh df và fdisk. Cú pháp lệnh:

```
df <option>, fdisk <option> <parameters>
```

Ví dụ:

Theo dõi các thông tin về file system được mount trong hệ thống.

Liệt kê file system trong hệ thống:

```
[root@server /]# df -l
Filesystem      1K-blocks      Used Available Use% Mounted on
/dev/sda1        2838408    2376896    309732   89% /
none              30608         0         30608    0% /dev/shm
[root@server /]#
```

In theo dạng (MB,GB)

```
[root@server /]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1        2.7G  2.3G  303M   89% /
none              30M   0     30M   0% /dev/shm
[root@server /]#
```

Liệt kê các partition trong hệ thống



```

root@server /l# fdisk -l
Disk /dev/sda: 3221 MB, 3221225472 bytes
255 heads, 63 sectors/track, 391 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *           1           358     2875603+   83  Linux
/dev/sda2                359         391       265072+   82  Linux swap
root@server /l#
    
```

Ta có thể tham khảo chi tiết thông tin về hai lệnh trên ta dùng lệnh man df

III.4. Duy trì hệ thống tập tin với lệnh fsck

Người quản trị hệ thống chịu trách nhiệm duy trì tính nhất quán của các hệ thống tập tin. Công việc thường làm là thỉnh thoảng kiểm tra xem có tập tin nào hỏng không. Linux sẽ tự động kiểm tra hệ thống tập tin lúc khởi động nếu chúng có giá trị lớn hơn 0 và được xác định trong trường pass number của tập tin /etc/fstab. Để thực hiện những công việc trên, người quản trị dùng lệnh fsck(file system check), Cú pháp như sau:

#fsck [tùy-chọn] hệ-thống-tập-tin

Bảng sau đây mô tả các tùy chọn:

Tùy chọn	Mô tả
-A	Duyệt khắp tập tin /etc/fstab và cố gắng kiểm tra tất cả các hệ thống tập tin chỉ trong một lần duyệt. Nếu đã chọn -A, bạn không cần chỉ ra hệ thống tập tin.
-V	Chế độ chi tiết. Cho biết lệnh fsck đang làm gì.
-t loại-fs	Xác định loại hệ thống tập tin cần kiểm tra
-a	Tự động sửa chữa những hỏng hóc trong hệ thống tập tin mà không cần hỏi
-l	Liệt kê tất cả các tên tập tin trong hệ thống tập tin
-r	Hỏi trước khi sửa chữa hệ thống tập tin
-s	Liệt kê các superbloc trước khi kiểm tra hệ thống tập tin.

IV. Các thao tác trên tập tin và thư mục

IV.1. Thao tác trên thư mục

IV.1.1 Đường dẫn tương đối và tuyệt đối

Đường dẫn trong Linux sử dụng là dấu / thay cho dấu \ được sử dụng trong Windows.

Để xác định một tập tin hay thư mục chúng ta dùng đường dẫn tuyệt đối hay tương đối. Đường dẫn tuyệt đối là đường dẫn đầy đủ đi từ thư mục gốc (/) của cây thư mục. Ví dụ : /home/hv, /usr/local/vd.txt



Trong một số trường hợp sử dụng các tập tin và thư mục là con của thư mục mình đang làm việc, lúc đó chúng ta sử dụng đường dẫn tương đối. Đường dẫn tương đối được tính từ thư mục hiện hành. Ví dụ chúng ta đang ở thư mục /home/hv khi gõ lệnh cat test.txt là chúng ta xem tập tin test.txt trong thư mục /home/hv.

Linux dùng ký hiệu “.” chỉ thư mục hiện hành và ký hiệu “..” chỉ thư mục cha của thư mục hiện hành. Ví dụ thư mục hiện hành là /usr/bin, đường dẫn ../local tương đương /usr/local

Chương trình thực thi trong Linux có 2 dạng chính là tập tin lệnh và tập tin binary. Tập tin lệnh là tập tin lưu các lệnh của shell tương tự tập tin bat trong DOS. Còn tập tin binary chứa mã máy tương tự tập tin .exe hoặc .com trong Windows. Trong Linux tên tập tin không có khái niệm mở rộng. Người ta thường sử dụng phần mở rộng để nói lên tính chất, ý nghĩa của tập tin chứ không để xác định chương trình thực thi tập tin. Ví dụ .txt chỉ tập tin dạng text, .conf chỉ tập tin cấu hình. Tập tin muốn thực thi được thì phải gán quyền thực thi(x).

Khi thực thi chương trình phải xác định đường dẫn chính xác hoặc sử dụng đường dẫn trong biến môi trường PATH. Do vậy, muốn thực thi tập tin trong thư mục hiện hành phải dùng ./tên-file

IV.1.2 Lệnh pwd

Lệnh pwd cho phép xác định vị trí thư mục hiện hành.

Ví dụ :

```
[natan@netcom bin]$ pwd
/usr/local/bin
```

IV.1.3 Lệnh cd

Lệnh cd cho phép thay đổi thư mục.

Cú pháp:

```
$cd [thư-mục]
thư-mục: là nơi cần di chuyển vào.
```

Ví dụ: \$cd /etc

IV.1.4 Lệnh ls

Lệnh ls cho phép liệt kê nội dung thư mục.

Cú pháp: ls [tùy chọn] [thư mục]

ls -x hiển thị trên nhiều cột.

ls -l hiển thị chi tiết các thông tin của tập tin.

ls -a hiển thị tất cả các tập tin kể cả tập tin ẩn.

Ví dụ: \$ ls -l /etc

```
-rw-r--r-- 1 root root 920 Jun 25 2001 im_palette-small.pal
-rw-r--r-- 1 root root 224 Jun 25 2001 im_palette-tiny.pal
-rw-r--r-- 1 root root 5464 Jun 25 2001 imrc
```



```
-rw-r--r--  1 root  root   10326 Apr 12 08:42 info-dir
lrwxrwxrwx  1 root  root    11 Apr 12 07:52 init.d -> rc.d/init.d
```

Ý nghĩa các cột từ trái sang phải

- + Cột 1: ký tự đầu tiên : - chỉ tập tin bình thường, **d** chỉ thư mục, **l** chỉ link và phía sau có dấu -> chỉ tới tập tin thật.
- + Các ký tự còn lại chỉ quyền truy xuất
- + Cột thứ 2: Chỉ số liên kết đến tập tin này.
- + Cột thứ 3, 4 : Người sở hữu và nhóm sở hữu
- + Cột thứ 5 : Kích thước tập tin, thư mục
- + Cột thứ 6 : Chỉ ngày giờ sửa chữa cuối cùng
- + Cột thứ 7 : Tên tập tin, thư mục

Bạn muốn xem thông tin 1 hay nhiều tập tin có thể dùng

```
$ls -l tập-tin1 tập-tin2 ...
```

IV.1.5 Lệnh mkdir

Lệnh mkdir cho phép tạo thư mục.

Cú pháp:

```
$mkdir [tùy-chọn] [thư-mục]
```

Ví dụ: \$mkdir /home/web

IV.1.6 Lệnh rmdir

Lệnh cho phép xóa thư mục rỗng

Cú pháp:

```
$rmdir [tùy-chọn] [thư-mục]
```

Ví dụ: \$rmdir /home/web

IV.2. Tập tin

IV.2.1 Lệnh cat

Lệnh cat dùng hiển thị nội dung của tập tin dạng văn bản. Để xem tập tin chúng ta chọn tên tập tin làm tham số.

Cú pháp:

```
$cat [tên-tập-tin]
```

Ví dụ: \$cat myfile

Lệnh cat còn cho phép bạn xem nhiều tập tin cùng lúc

```
$cat file1 file2 ...
```




Cat cũng được dùng để tạo và soạn thảo văn bản dạng text. Trong trường hợp này chúng ta sử dụng dấu > hay >> đi theo sau. Nếu tập tin cần tạo đã tồn tại, dấu > sẽ xóa nội dung của tập tin và ghi nội dung mới vào, dấu >> sẽ ghi nối nội dung mới vào sau nội dung cũ của tập tin.

```
$cat > <tên-tập-tin> [Enter]
```

```
> Các-dòng-dữ-liệu-của-tập-tin
```

```
> ...
```

```
[Ctrl-d :kết thúc]
```

IV.2.2 Lệnh more

Lệnh more cho phép xem nội dung tập tin theo từng trang màn hình.

Cú pháp:

```
$more [tên-tập-tin]
```

Ví dụ:

```
$more /etc/passwd
```

IV.2.3 Lệnh cp

Lệnh cp cho phép sao chép tập tin

Cú pháp:

```
$cp <tập-tin-nguồn> <tập-tin-đích>
```

Ví dụ: \$cp /etc/passwd /root/passwd

IV.2.4 Lệnh mv

Lệnh mv cho phép thay đổi tên tập tin và di chuyển vị trí của tập tin

Cú pháp:

```
$mv <tên-tập-tin-cũ> <tên-tập-tin-mới>
```

Ví dụ: \$cp /etc/passwd /root/pwd

IV.2.5 Lệnh rm

Lệnh rm cho phép xóa tập tin, thư mục.

Cú pháp:

```
$rm [tùy-chọn] [tên-tập-tin/thư-mục]
```

Các tùy chọn hay dùng:

-r : xóa thư mục và tất cả các tập tin và thư mục con

-I : xác nhận lại trước khi xóa

IV.2.6 Lệnh find

Cho phép tìm kiếm tập tin thỏa mãn điều kiện.



Cú pháp:

#find [đường-dẫn] [biểu-thức-tìm-kiếm]

- đường-dẫn: là đường dẫn thư mục tìm kiếm
- biểu-thức-tìm-kiếm : tìm các tập tin hợp với điều kiện tìm .

Tìm 1 tập tin xác định :

#find [thư-mục] -name [tên-tập-tin] -print

Ngoài ra, bạn có thể sử dụng những kí hiệu sau:

“*” : viết tắt cho một nhóm ký tự

“?” : viết tắt cho một ký tự

Có thể sử dụng man để có các lựa chọn tìm kiếm đầy đủ hơn

IV.2.7 Lệnh grep

Lệnh grep cho phép tìm kiếm một chuỗi nào đó trong nội dung tập tin.

Cú pháp :

#grep [biểu-thức-tìm-kiếm] [tên-tập-tin]

Tìm trong tập tin có tên [tên-tập-tin] những dữ liệu thỏa mãn [biểu-thức-tìm-kiếm]

Ví dụ : grep “nva” /etc/passwd

Tìm kiếm trong tập tin /etc/passwd và hiển thị các dòng có xuất hiện chuỗi “nvan”.

IV.2.8 Lệnh touch

Là lệnh hỗ trợ việc tạo và thay đổi nội dung tập tin

Cú pháp : touch <option> file

Ví dụ: #touch file1.txt file2.txt (tạo hai tập tin file1.txt và file2.txt)

IV.2.9 Lệnh dd

Sao chép và chuyển đổi file.

Ví dụ:

```
dd if=/mnt/cdrom/images/boot.img of=/dev/fd0
```

(if là input file, of là output file)

IV.3. Các tập tin chuẩn trong Linux

Khi khởi động chương trình Linux, nó giao tiếp với người dùng qua việc hiển thị thông tin ra màn hình. Thông tin hiển thị màn hình có thể là dữ liệu của chương trình hay lỗi phát sinh khi có lỗi xảy ra. Người dùng giao tiếp với chương trình qua các ký tự gõ vào bàn phím. Luồng dữ liệu vào từ bàn phím gọi là nhập chuẩn. Luồng dữ liệu ra màn hình gọi là xuất chuẩn còn luồng dữ liệu thông báo lỗi là lỗi chuẩn.



Trong Linux, các luồng giao tiếp chuẩn được xem như các tập tin dữ liệu và được đánh số theo thứ tự: Tập tin nhập (file input) chuẩn là 0, tập tin xuất (file output) chuẩn là 1 và tập tin lỗi chuẩn là 2. Các số này được gọi là tập tin mô tả (file descriptor).

Sử dụng chương trình cat để soạn thảo, chúng ta gõ:

```
$ cat > filename
```

...<nhập nội dung cho tập tin>

<Ctrl-d>.

Tất cả các dữ liệu chúng ta đưa vào từ bàn phím được xem là tập tin nhập chuẩn. Dùng lệnh ls bạn sẽ nhận được dữ liệu ra màn hình, đó là tập tin xuất chuẩn chuẩn.

Một thông báo lỗi xuất hiện ở màn hình khi chúng ta gõ lệnh sai hoặc truy xuất vào các tập tin hay thư mục không có quyền chính là tập tin lỗi chuẩn. Ví dụ như bạn gõ lệnh listn thì sẽ xuất hiện lỗi invalid command.

IV.3.1 Chuyển hướng (redirection)

Chuyển tiếp là hình thức thay đổi luồng dữ liệu của các nhập, xuất và lỗi chuẩn. Khi sử dụng chuyển tiếp, nhập chuẩn có thể nhận dữ liệu từ tập tin thay vì bàn phím, xuất và lỗi chuẩn có thể xuất ra tập tin hay máy in...

Có 3 loại chuyển hướng:

- + Chuyển hướng nhập(Input redirection)
- + Chuyển hướng xuất(Output redirection)
- + chuyển hướng lỗi(Error redirection)

IV.3.2 Chuyển hướng nhập:

Theo qui ước thì các lệnh lấy dữ liệu từ thiết bị nhập chuẩn(bàn phím). Để lệnh lấy dữ liệu từ tập tin chúng ta dùng ký hiệu < :

Cú pháp:

```
$lệnh < tập_tin
```

Dấu "<" chỉ hướng chuyển dữ liệu.

Ví dụ \$cat < abc.txt hoặc \$cat 0< abc.txt

IV.3.3 Chuyển hướng xuất:

Kết quả của các lệnh thông thường được hiển thị trên màn hình. Để xuất kết quả này ra tập tin bạn dùng dấu ">"

Cú pháp: \$lệnh > tập-tin

Ví dụ: Liệt kê nội dung thư mục và chuyển kết quả ra tập tin

```
$ls -l > tm.txt
```

Để chèn thêm dữ liệu vào cuối tập tin đã tồn tại bạn dùng dấu ">>" thay cho dấu ">"

Cú pháp: \$lệnh >> tập-tin



Ví dụ: \$cat a.txt >> sum.txt

IV.4. Đường ống (Pipe)

Pipe là còn gọi là truyền thẳng. Nó là cách truyền dữ liệu sử dụng kết hợp 2 chuyển tiếp. Pipe sử dụng kết xuất của một chương trình làm dữ liệu nhập cho một chương trình khác.

Ví dụ: \$ls -l | more

Kết quả của lệnh ls không xuất ra màn hình mà chuyển cho lệnh more xử lý như dữ liệu đầu vào.

IV.5. Lệnh tee

Hoạt động chuyển tiếp và đường ống là đặc điểm của hệ điều hành UNIX. Tuy nhiên bạn cũng có thể sử dụng 1 lệnh của Linux để làm việc này. Đó là lệnh tee, nó sẽ giảm bớt các kết quả gián tiếp của chuỗi đường ống.

Ví dụ: \$sort baocao | tee baocaostt | lp

Đầu tiên lệnh tee gửi nhập chuẩn của nó đến xuất chuẩn của nó, trong trường hợp này gửi xuất của sort đến nhập của lp. Thứ hai tee sao chép 1 bản nhập chuẩn vào tập tin baocaostt.

V. Lưu trữ tập tin/thư mục

V.1. Lệnh gzip/gunzip

gzip dùng để nén tập tin, còn gunzip dùng để giải nén các tập tin đã nén. Cú pháp của gzip và gunzip như sau:

\$gzip [tùy-chọn] [tên-tập-tin]

\$gunzip [tùy-chọn] [tên-tập-tin]

gzip tạo tập tin nén với phần mở rộng .gz

Các tùy chọn dùng cho gunzip và gzip:

-c	Chuyển các thông tin ra màn hình
-d	Giải nén, gzip -d tương đương gunzip
-h	Hiển thị giúp đỡ.

Ví dụ:

#gzip /etc/passwd

#gunzip /etc/passwd.gz

V.2. Lệnh tar

Lệnh này dùng để gom và bung những tập tin/thư mục. Nó sẽ tạo ra một tập tin có phần mở rộng .tar

Cú pháp: #tar [tùy-chọn] [tập-tin-đích] [tập-tin-nguồn/thư-mục-nguồn ...]



Trong đó:

- + `-cvf` : gom tập tin/ thư mục
- + `-xvf` : bung tập tin / thư mục
- + `tập-tin-đích`: tập tin `.tar` sẽ được tạo ra.
- + `tập-tin-nguồn/thư-mục-nguồn`: những tập tin và thư mục cần gom.

Ví dụ:

```
#tar -cvf /home/backup.tar /etc/passwd /etc/group
```

```
#tar -xvf /home/backup.tar
```

VI. Bảo mật hệ thống tập tin

VI.1. Quyền hạn

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

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

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

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

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

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

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

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

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



Ký tự đầu tiên của quyền là ký tự “-” cho biết đó là một tập tin bình thường. Nếu ký tự d thay thế cho dấu “-” thì myfile là một thư mục. Ngoài ra còn có c cho thiết bị ngoại vi dạng ký tự (như bàn phím), b cho thiết bị ngoại vi dạng block (như ổ đĩa cứng).

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

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

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

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

Bit 2	bit 1	bit 0
read	write	excute

Ví dụ:

chỉ có quyền đọc : 100 có giá trị là 4

có quyền đọc và thực thi : 101 có giá trị là 5

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

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

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

read , write và excute : 4+2+1=7

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



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

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

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

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

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

VI.2. Lệnh chmod, chown, chgrp

VI.2.1 Lệnh chmod

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

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

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

Một số ví dụ : gán quyền trên tập tin myfile

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

Xóa quyền read trên group và others : \$ chmod go-w myfile

Cấp quyền x cho mọi người:

\$ chmod ugo+x myfile hoặc



`$chmod a+x myfile` hoặc

`$ chmod +x myfile`

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

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

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

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

Ví dụ: Cấp quyền cho tập tin myfile

Quyền	Lệnh
<code>-wrxr-xr-x</code>	<code>\$chmod 755 myfile</code>
<code>-r-xr--r --</code>	<code>\$chmod 522 myfile</code>
<code>-rwxrwxrwx</code>	<code>\$chmod 777 myfile</code>

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

VI.2.2 Lệnh chown

Lệnh `chown` dùng để thay đổi người sở hữu trên tập tin, thư mục

Cú pháp: `$chown [tên-user:tên-nhóm] [tên-tập-tin/thư-mục]`

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

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

VI.2.3 Lệnh chgrp

Lệnh `chgrp` dùng để thay đổi nhóm sở hữu của một tập tin, thư mục

Cú pháp: `$chgrp [nhóm-sở-hữu] [tên-tập-tin/thư-mục]`



Bài 4 Cài Đặt Phần Mềm

Tóm tắt

Lý thuyết: 3 tiết - Thực hành: 5 tiết.

Mục tiêu	Các mục chính	Bài tập bắt buộc	Bài tập làm thêm
Giới thiệu học viên cơ chế cài đặt, tổ chức, quản lý phần mềm trên môi trường Linux	I. Chương trình RPM II. Đặc tính của RPM III. Lệnh rpm	Bài tập 4.1 (Sách bài tập – Cài đặt phần mềm)	



I. Chương trình RPM

RedHat Package Manager (RPM) là hệ thống quản lý package (gói phần mềm) được Linux hỗ trợ cho người dùng. Nó cung cấp cho người dùng nhiều tính năng để duy trì hệ thống của mình. Người dùng có thể cài đặt, xóa hoặc nâng cấp các package trực tiếp bằng lệnh. RPM có một cơ sở dữ liệu chứa các thông tin của các package đã cài và các tập tin của chúng, nhờ vậy RPM cho phép bạn truy vấn các thông tin, cũng như xác thực các package trong hệ thống. Nếu bạn sử dụng XWindow, có thể dùng chương trình KDE-PRM hoặc Gnome-RPM thay thay cho việc sử dụng lệnh.

Trong quá trình nâng cấp package, RPM thao thác trên tập tin cấu hình rất cẩn thận, do vậy mà bạn không bao giờ bị mất các lựa chọn trước đó của mình. Trên phương diện các nhà phát triển, nó cho phép các nhà phát triển đóng gói chương trình nguồn của phần mềm thành các package dạng nguồn hoặc binary đưa tới người dùng.

II. Đặc tính của RPM

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

- **Khả năng nâng cấp phần mềm:** Với RPM bạn có thể nâng cấp các thành phần riêng biệt của hệ thống mà không cần phải cài lại. Khi có một phiên bản mới của hệ điều hành dựa trên RPM (như RedHat Linux chẳng hạn) thì bạn không phải cài lại hệ thống mà chỉ cần nâng cấp thôi. RPM cho phép nâng cấp hệ thống một cách tự động, thông minh. Các tập tin cấu hình được gìn giữ cẩn thận qua các lần nâng cấp, vì thế bạn không sợ thay đổi các tùy chọn sẵn có của hệ thống được nâng cấp.
- **Truy vấn thông tin hiệu quả:** RPM cũng được thiết kế cho mục đích truy vấn các thông tin về các package trong hệ thống. Bạn có thể tìm kiếm thông tin các package hoặc các tập tin cài đặt trong toàn bộ cơ sở dữ liệu. Bạn cũng có thể hỏi tập tin cụ thể thuộc về package nào và nó ở đâu. Package RPM có các tập tin chứa các thông tin rất hữu ích về package này và nội dung của package. Các tập tin này cho phép người dùng tìm kiếm thông tin dễ dàng trong một package riêng lẻ.
- **Thẩm tra hệ thống (System Verification):** Một đặc tính rất mạnh của RPM là cho phép bạn thẩm tra lại các package. Nếu bạn nghi ngờ một tập tin nào bị xóa hay bị thay thế trong package, bạn có thể kiểm tra lại rất dễ dàng. Bạn cần phải chú ý đến các dấu hiệu bất bình thường của hệ thống, nên kiểm tra và cài lại nếu cần thiết.

III. Lệnh rpm

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

III.1. Cài đặt phần mềm bằng rpm

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



rpm -ivh tên-tập-tinRPM

Ví dụ:

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

foo #####

Một số trường hợp lỗi khi cài đặt.

- Package đã cài rồi.
- Xung đột với tập tin cũ đã tồn tại.
- Package phụ thuộc vào package khác.

Ví dụ: package đã được cài đặt trước

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

foo package foo-1.0-1 is already installed

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

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

Ví dụ:

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

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

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

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

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

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

Ví dụ: Package phụ thuộc vào package khác

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

failed dependencies:

bar is needed by foo-1.0-1

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

III.2. Loại bỏ phần mềm đã cài đặt trong hệ thống

Xóa package thì đơn giản hơn cài. Lệnh xóa.

rpm -e tên-package

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

Ví dụ:

rpm -e foo

removing these packages would break dependencies:

foo is needed by bar-1.0-1



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

III.3. Nâng cấp phần mềm

Upgrade cũng tương tự như cài đặt mới.

```
# rpm -Uvh tên-tập-tinRPM
```

Ví dụ:

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

```
foo #####
```

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

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

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

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

Trong trường hợp này bạn thêm tham số `--oldpackage`

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

```
foo #####
```

III.4. Truy vấn các phần mềm

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

```
# rpm -q tên-package
```

Ví dụ:

```
# rpm -q foo
```

```
foo-2.0-1 //kết quả truy vấn
```

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

- + `-a` : Truy vấn tất cả các package.
- + `-f <tập-tin>`: Truy vấn những package chứa tập-tin. Khi xác định tập tin bạn phải chỉ rõ đường dẫn (ví dụ : /usr/bin/lis)
- + `-p <tên-tập-tin-package>` : Truy vấn package tên-tập-tin-package

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



- + -i : xác định các thông tin về package bao gồm : tên, mô tả, phiên bản, kích thước, ngày tạo, ngày cài đặt, nhà sản xuất ...
- + -l : Hiển thị những tập tin trong package.
- + -s : Hiển thị trạng thái của các tập tin trong package.
- + -d : hiển thị danh sách tập tin tài liệu cho package (ví dụ man, README, info file ...)
- + --c : hiển thị danh sách tập tin cấu hình.

III.5. Kiểm tra các tập tin đã cài đặt

Kiểm tra xem tập tin đã cài đặt với các tập tin gốc của package. Các thông tin dùng kiểm tra là : kích thước, MD5 checksum, quyền hạn, loại tập tin, người sở hữu, nhóm sở hữu tập tin.

- + rpm -V tên-package :Kiểm tra tất cả các tập tin trong package.
- + rpm -vf tên-file : Kiểm tra tập tin tên-file
- + rpm -Va :Kiểm tra tất cả các package đã cài.
- + rpm -Vp tên-tập-tin-RPM :Kiểm tra một package với tập tin package xác định, thường sử dụng trong trường hợp cơ sở dữ liệu của RPM bị hỏng.

Khi kiểm tra nếu không có lỗi thì không có hiển thị, nếu không thì sẽ thông báo ra. Định dạng của dòng thông báo gồm 8 ký tự và tên tập tin. Mỗi ký tự biểu diễn cho kết quả của việc so sánh một thuộc tính của tập tin với thuộc tính lưu trong cơ sở dữ liệu RPM. Dấu chấm (.) nghĩa là đã kiểm tra xong. Những ký tự đại diện cho các lỗi kiểm tra.

- + 5 – MD5 checksum
- + S – kích thước tập tin
- + L – liên kết mềm
- + T - thời gian cập nhật tập tin
- + D - thiết bị
- + U – người sở hữu
- + G – nhóm sở hữu
- + M - quyền truy xuất và loại tập tin.
- + ? – không tìm thấy tập tin

III.6. Cài đặt phần mềm file nguồn *.tar, *.tgz

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

Bước 1: Giải nén file tar.

Ví dụ:

```
[root@bigboy tmp]# tar -xvzf linux-software-1.3.1.tar.gz
linux-software-1.3.1/
linux-software-1.3.1/plugins-scripts/
```



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

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

Ví dụ:

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

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



Bài 5

Giới Thiệu Các Trình Tiện Ích

Tóm tắt

Lý thuyết: 4 tiết - Thực hành: 5 tiết.

Mục tiêu	Các mục chính	Bài tập bắt buộc	Bài tập làm thêm
Giới thiệu các trình tiện ích, công cụ được sử dụng phổ biến trên môi trường Unix/Linux, trợ giúp cho học viên sử dụng để tổ chức và quản trị hệ thống hiệu quả hơn.	I. Trình soạn thảo vi II. Trình tiện tích mail III. Tạo đĩa mềm boot IV. Trình tiện ích setup V. Trình tiện ích fdisk VI. Trình tiện ích iptraf VII. Trình tiện ích lynx VIII. Trình tiện ích mc	Bài tập 5.1 (sách bài tập – Trình tiện ích)	



I. Trình soạn thảo vi

Vim là chương trình soạn thảo chuẩn trên các hệ điều hành Unix. Nó là chương trình soạn thảo trực quan, hoạt động dưới 2 chế độ : Chế độ lệnh (command mode) và chế độ soạn thảo (input mode). Để soạn thảo tập tin mới hoặc xem hay sửa chữa tập tin cũ bạn dùng lệnh:

```
$vi [tên-tập-tin]
```

Khi thực hiện, vi sẽ hiện lên màn hình soạn thảo ở chế độ lệnh. Ở chế độ lệnh, chỉ có thể sử dụng các phím để thực hiện các thao tác như: Dịch chuyển con trỏ, lưu dữ liệu, mở tập tin mới...Do đó, bạn không thể soạn thảo văn bản. Nếu muốn soạn thảo văn bản, bạn phải chuyển từ chế độ lệnh sang chế độ soạn thảo. Chế độ soạn thảo giúp bạn sử dụng bàn phím để soạn nội dung văn bản.

I.1. Một số hàm lệnh của vi

- vi tập tin --> bắt đầu dòng 1
- vi +n tập tin --> bắt đầu ở dòng n
- vi +/pattern --> bắt đầu ở pattern
- vi -r tập tin --> phục hồi tập tin sau khi hệ thống treo

I.2. Chuyển chế độ lệnh sang chế độ soạn thảo

Dưới đây là nhóm lệnh để chuyển sang chế độ soạn thảo. Tùy theo yêu cầu mà bạn sử dụng hợp lệ.

- i trước dấu con trỏ
- I trước ký tự đầu tiên trên dòng
- a sau dấu con trỏ
- A sau ký tự đầu tiên trên dòng
- o dưới dòng hiện tại
- O trên dòng hiện tại
- r thay thế 1 ký tự hiện hành
- R thay thế cho đến khi nhấn <ESC>

I.3. Chuyển chế độ soạn thảo sang chế độ lệnh

Dùng phím ESC (escape), sau đó sử dụng các nhóm lệnh thích hợp sau:

I.3.1 Nhóm lệnh di chuyển con trỏ

- h sang trái một khoảng trắng
- e sang phải một khoảng trắng
- <space> - nt -
- w sang phải 1 từ
- b sang trái 1 từ
- k lên một dòng



- j xuống một dòng
- <return> - nt -
-) cuối câu
- (đầu câu
- } đầu đoạn văn
- { cuối đoạn văn
- ^-w đến ký tự đầu tiên chèn vào
- ^-u cuộn lên 1/2 màn hình
- ^-d kéo xuống 1/2 màn hình
- ^-z kéo xuống 1 màn hình
- ^-b kéo lên 1 màn hình

Lưu ý: dấu “^” viết tắt cho phím Ctrl

I.3.2 Nhóm lệnh xóa

- Dw 1 từ
- do đến đầu dòng
- d\$ cuối dòng
- 3dw 3 từ
- dd dòng hiện hành
- 5dd 5 dòng
- x xóa 1 ký tự

I.3.3 Nhóm lệnh thay thế

- cw Thay thế 1 từ
- 3cw Thay thế 3 từ
- cc Dòng hiện hành
- 5cc 5 dòng

I.3.4 Nhóm lệnh tìm kiếm

- */and Từ kế tiếp của and
- *?and Từ kết thúc là and
- */nThe Tìm dòng kế bắt đầu bằng The
- n Lặp lại lần dò tìm sau cùng

I.3.5 Nhóm lệnh tìm kiếm và thay thế

- :s/text1/text2/g Thay text1 thành text2
- :1,\$s/tập tin/thư mục Thay tập tin bằng thư mục từ hàng 1 đến cuối.
- :g/one/s//1/g Thay thế one bằng 1

I.3.6 Copy and paste

- Để copy ta dùng lệnh y và để paste dùng lệnh p



- y\$: copy từ vị trí hiện tại của cursor đến cuối dòng.
- yy : copy toàn bộ dòng tại vị trí cursor.
- 3yy : copy ba dòng liên tiếp.

I.3.7 Undo

Thao tác undo cho phép chúng ta hủy thao tác hiện tại và quay về thao tác trước đó, trong vi thực hiện bằng phím u.

I.3.8 Thao tác trên tập tin

- :w ghi vào tập tin
- :x lưu và thoát khỏi chế độ soạn thảo
- :wq lưu và thoát khỏi chế độ soạn thảo
- : w <filename> lưu vào tập tin mới
- :q thoát nếu không có thay đổi nội dung tập tin
- :q! thoát không lưu nếu có thay đổi tập tin
- :r mở tập tin đọc .

II. Trình tiện ích mail

Trình tiện ích này do Linux cung cấp để hỗ trợ cho việc gửi và nhận mail.

\$mail

Lệnh này sẽ hiển thị nội dung các mail trong mailbox theo thứ tự vào trước ra sau. Sau khi hiển thị mỗi mail sẽ hiện lên dấu “?” để chờ lệnh của người sử dụng. các thao tác cơ bản sau:

- newline Hiển thị mail kế, nếu không còn thì thoát khỏi lệnh.
- + giống như newline
- p In thông báo
- s [tập tin] lưu mail vào tập tin khác hoặc mailbox
- w [tập tin] giống như s nhưng không lưu đầu thông báo
- d xóa mail
- q thoát khỏi tiện ích
- x thoát khỏi tiện ích mà không thay đổi mail
- ![lệnh] thực hiện [lệnh] Unix

Gửi mail: Đưa vào lệnh mail với địa chỉ của người sử dụng. Ví dụ :

```
$ mail dung@fibi.hcm.vn
```

```
<nội dung>
```

```
^D
```

Mail sẽ được gửi cho người sử dụng có tên là dung ở công ty fibi vùng hcm.vn. Có thể cùng một lúc gửi một thông báo cho nhiều người

```
$ mail dung@fibi.hcm.vn trung@fibi.hanoi.vn
```



Nhận mail : Khi login vào hệ thống nếu có thư hệ thống sẽ thông báo “ You have mail” khi đó có thể đánh \$mail để nhận mail. Tương tự ta có thể dùng các tiện ích như: sendmail, pine thông qua trợ giúp man.

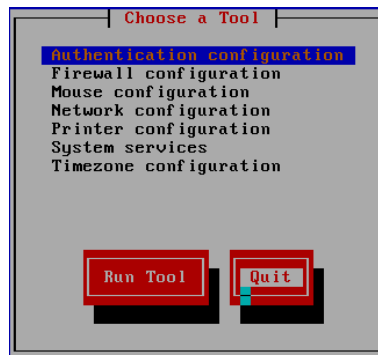
III. Tiện ích tạo đĩa mềm boot

Ta có thể sử dụng lệnh mkbootdisk để tạo đĩa mềm khởi động hệ thống. Các bước thực hiện như sau:

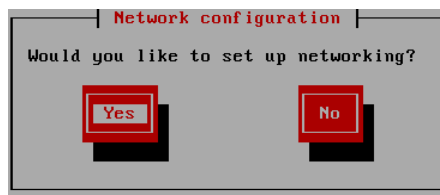
- Đăng nhập vào hệ thống bằng user root.
- Xem phiên bản kernel của Linux dùng lệnh `ls /lib/modules/` hoặc lệnh `uname -r` (trong ví dụ này Linux kernel là 2.2.12-20).
- Sử dụng lệnh `/sbin/mkbootdisk 2.2.12-20` từ dấu nhắc shell
- Đưa đĩa mềm vào ổ đĩa khi được hệ thống yêu cầu (Insert a disk in /dev/fd0. Any information on the disk will be lost.)

IV. Trình tiện ích setup

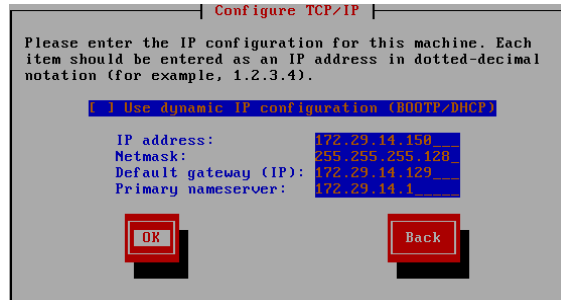
Là trình tiện ích hỗ trợ cài đặt thiết bị, filesystem, thiết lập cấu hình mạng, dịch vụ hệ thống, từ dấu nhắc lệnh ta enter vào lệnh **setup**, dialog chọn công cụ sẽ được hiển thị.



Ta có thể dùng chương trình này để cài đặt thống cấu hình TCP/IP cho hệ thống, từ giao diện trên ta chọn item Network Configuration -> Run Tool



Sau khi ta chọn Yes để thực hiện quá trình cấu hình thích hợp



Sau đó ta chọn Ok -> Exit. Có thể dùng lệnh `/etc/init.d/network restart` để cập nhật lại các thông số mạng.

V. Trình tiện ích fdisk

Là trình tiện ích cho phép quản lý ổ đĩa cứng như: tạo mới, xem thông tin và xóa các partition trong hệ thống. Cú pháp lệnh:

`#fdisk <device_name>`

Trong đó `<device_name>` có thể là `/dev/hda` hoặc `/dev/sda`. Sau đây là một số lệnh fdisk cơ bản.

Lệnh	Giải thích
P	Liệt kê danh sách các partition table
N	Tạo mới 1 partition
D	Xóa partition
Q	Thoát khỏi trình tiện ích
W	Tạo mới partition
A	Thiết lập boot partition
T	Thay đổi system partition ID
L	Liệt kê loại partition (bao gồm ID)

Sau đây là một số bước để tạo mới một partition với dung lượng 384M

Bước thực hiện	Giải thích
<code># fdisk /dev/hdb</code>	Khởi tạo tiện ích fdisk để thao tác lên Partition /dev/hdb
Command (m for help): p Disk /dev/hdb: 64 heads, 63 sectors, 621 cylinders Units = cylinders of 4032 * 512 bytes	Liệt kê danh sách các partition trong hệ thống.
Command (m for help): n Command action e extended p primary partition (1-4) p	Tạo mới một primary partition với kích thước 384MB
Partition number (1-4): 1 First cylinder (1-621, default 1):<RETURN> Using default value 1 Last cylinder or +size or +sizeM or +sizeK (1-621, default 621): +384M	

```

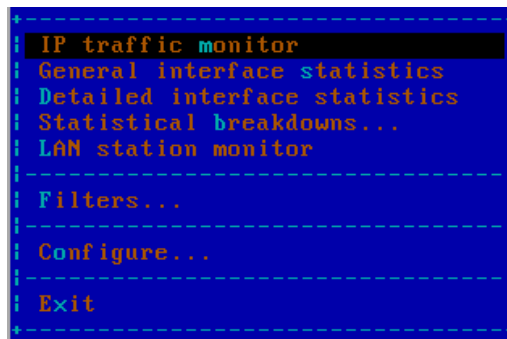
Command (m for help): p
Device Boot  Start    End  Blocks  Id
System
/dev/hdb1    1      196  395104
83 Linux
    
```

Lưu ý: Sau khi ta dùng fdisk để tạo một partition mới thì ta phải reboot lại hệ thống và dùng lệnh **mkfs -t ext3 <filesystem>** để định dạng lại partition đó trước khi sử dụng.

VI. Trình tiện ích iptraf

Là trình tiện ích hỗ trợ việc theo dõi và giám sát các traffic trên mạng, lưu ý rằng ta phải cài chương trình này từ đĩa CDROM bằng lệnh `rpm -ivh iptraf...rpm`

Sau đây là một số màn hình minh họa cho việc sử dụng tiện ích iptraf để theo dõi lưu lượng mạng. Từ dấu nhắc lệnh enter vào lệnh **iptraf**.



Tên tiện ích	Giải thích
IP traffic monitor	Theo dõi ip traffic và TCP connection
General interface statistics	Xem các thông tin tổng quát trên các interface
Detailed interface statistics	Xem thông tin chi tiết trên từng interface (tổng số byte gửi, tổng số byte nhận, ...)
Statistical breakdown ...	Thống kê các packet bị hủy bỏ trên các interface do một số sự cố mạng
LAN station monitor	Thống kê thông tin từ máy mạng gửi vào máy nội bộ.
Filters...	Cho phép thiết lập bộ lọc thông tin dựa theo các giao thức mạng TCP/UDP...
Configure...	Cấu hình các thông số cho trình tiện ích iptraf

VII. Trình tiện ích lynx

Lynx là một trong những trình duyệt Web có giao diện text. Lynx cho phép người dùng có thể sử dụng để truy xuất Web qua giao diện text thay vì sử dụng giao diện đồ họa của XWindows. Lynx có thể sử dụng trong console, terminal hoặc xterm. Cú pháp lệnh lynx:

`#lynx <URL>`



Ví dụ: #lynx webmail.tatavietnam.vn

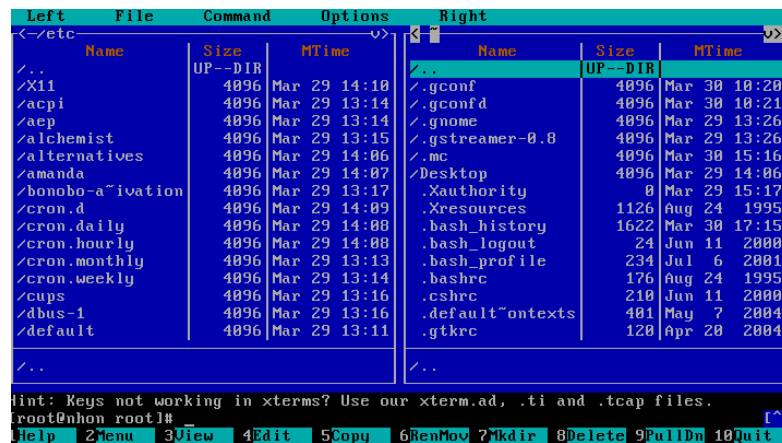
```
04/07/2005 06:57:32 pm +0700 - iso-8859-1 - Open WebMail
[openwebmail.gif]
Login
UserID: _____
Password: _____
Login [X] HTTP
Compression [ ] Auto
Login
Open WebMail version 2.51 Help?
(NORMAL LINK) Use right-arrow or <return> to activate.
Arrow keys: Up and Down to move. Right to follow a link; Left to go back.
H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list |
```

Ta có thể tham khảo màn hình chính của trình duyệt Web Browser để xem trợ giúp:

Ví dụ: chọn phím g để duyệt trang Web khác, phím o để hiệu chỉnh tùy chọn, phím p để in thông tin ra máy in....

VIII. Trình tiện ích mc

GNU Midnight Commander là chương trình quản lý và thao tác trên file và thư mục được sử dụng trên Unix/Linux, để sử dụng ta phải cài package mc, sau đó dùng lệnh mc để kích hoạt chương trình, mc có khả năng cung cấp tính năng truyền file thông qua ftp và ssh.





Bài 6

Quản Trị Người Dùng Và Nhóm

Tóm tắt

Lý thuyết: 5 tiết - Thực hành: 5 tiết.

Mục tiêu	Các mục chính	Bài tập bắt buộc	Bài tập làm thêm
Giới thiệu cơ chế tổ chức và quản trị người dùng trên Linux.	I. Superuser II. Thông tin của User III. Quản lý người dùng IV. Nhóm người dùng	Bài tập 6.1 (tham khảo “Sách bài tập”)	



I. Superuser

Trong hệ thống Linux, tài khoản root có quyền cao nhất được sử dụng bởi người quản trị. Sử dụng quyền root chúng ta thấy rất thoải mái vì chúng ta có thể thực hiện các thao tác mà không phải lo lắng gì đến vấn đề quyền truy cập vì root có quyền cao nhất trong hệ thống. Tuy nhiên, khi hệ thống bị sự cố do một lỗi làm nào đó, chúng ta mới thấy sự nguy hiểm khi làm việc với quyền root, do vậy chúng ta chỉ sử dụng tài khoản này vào các mục đích cấu hình, bảo trì hệ thống chứ không nên sử dụng vào mục đích hằng ngày. Bạn cần tạo các tài khoản (account) cho người sử dụng thường sớm nhất có thể được (đầu tiên là cho bản thân bạn). Với những server quan trọng và có nhiều dịch vụ khác nhau, bạn có thể tạo ra các superuser thích hợp cho từng dịch vụ để tránh dùng root cho các công tác này. Ví dụ như superuser cho công tác backup chỉ cần chức năng đọc (read-only) mà không cần chức năng ghi.

Tài khoản root này có quyền hạn rất lớn nên nó là mục tiêu mà các kẻ xấu muốn chiếm đoạt, chúng ta sử dụng tài khoản root phải cẩn thận, không sử dụng bừa bãi trên qua telnet hay kết nối từ xa mà không có công cụ kết nối an toàn.

Trong Linux, chúng ta có thể tạo tài khoản có tên khác nhưng có quyền của root, bằng cách tạo user có UserID bằng 0. Cần phân biệt bạn đang login như root hay người sử dụng thường thông qua dấu nhắc của shell.

```
login: natan
```

```
Password:****
```

```
[natan@NetGroup natan]$ su -
```

```
Password: ****
```

```
[root@NetGroup /root]#
```

Dòng thứ tư với dấu \$ cho thấy bạn đang kết nối như một người sử dụng thường (natan). Dòng cuối cùng với dấu # cho thấy bạn đang thực hiện các lệnh với root. Lệnh su user_name cho phép bạn thay đổi login dưới một tài khoản khác (user_name) mà không phải logout rồi login trở lại.

II. Thông tin của User

Mọi người muốn đăng nhập và sử dụng hệ thống Linux đều cần có 1 tài khoản. Việc tạo và quản lý tài khoản là vấn đề quan trọng mà người quản trị phải thực hiện. Trừ tài khoản root, các tài khoản khác do người quản trị tạo ra.

Mỗi tài khoản người dùng phải có một tên sử dụng (username) và mật khẩu (password) riêng. Tập tin /etc/passwd là tập tin chứa các thông tin về tài khoản người dùng của hệ thống.

II.1. Tập tin /etc/passwd

Tập tin /etc/passwd đóng vai trò sống còn đối với một hệ thống Unix/Linux. Mọi người đều có thể đọc được tập tin này nhưng chỉ có root mới có quyền thay đổi nó. Tập tin /etc/passwd được lưu dưới dạng văn bản như hầu hết các tập tin cấu hình khác của Linux. Chúng ta thử xem qua nội dung của tập tin passwd:

```
root:x:0:0:root:/root:/bin/bash
```




```
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:
news:x:9:13:news:/var/spool/news:
ftp:x:14:50:FTP User:/var/ftp:
nobody:x:99:99:Nobody:/:
nscd:x:28:28:NSCD Daemon:./bin/false
mailnull:x:47:47:./var/spool/mqueue:/dev/null
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/bin/false
xfs:x:43:43:X Font Server:/etc/X11/fs:/bin/false
nhung:x:525:526:nguyen tien hung:/home/nhung:/bin/bash
natan:x:526:527:./home/natan:/bin/bash
```

Mỗi tài khoản được lưu trong một dòng gồm 7 cột:

- Cột 1 : Tên người sử dụng.
- Cột 2 : Mã liên quan đến mật khẩu của tài khoản và “x” đối với Linux. Linux lưu mã này trong một tập tin khác /etc/shadow mà chỉ có root mới có quyền đọc.
- Cột 3:4 : Mã định danh tài khoản (user ID) và mã định danh nhóm (group ID).
- Cột 5 : Tên đầy đủ của người sử dụng. Một số phần mềm phá password sử dụng dữ liệu của cột này để thử đoán password.
- Cột 6 : thư mục cá nhân. (Home Directory)
- Cột 7 : Chương trình sẽ chạy đầu tiên sau khi người dùng đăng nhập vào hệ thống.

Dòng đầu tiên của tập tin /etc/passwd mô tả thông tin cho user root (chú ý là tất cả những tài khoản có user_ID = 0 đều là root), tiếp theo là các tài khoản khác của hệ thống (đây là các tài khoản không có thật và không thể login vào hệ thống), cuối cùng là các tài khoản người dùng thường.

II.2. Username và UserID

Tên người dùng là chuỗi ký tự xác định duy nhất một người dùng, người dùng sử dụng tên này khi đăng nhập cũng như truy xuất tài nguyên, trong Linux tên người dùng có sự phân biệt giữa chữ hoa và thường. Thông thường, tên người dùng thường sử dụng chữ thường. Để dễ dàng trong việc quản lý người dùng, ngoài tên người dùng Linux còn sử dụng khái niệm định danh người dùng (user_ID). Mỗi người dùng có một con số định danh riêng.

Linux sử dụng số định danh để kiểm soát hoạt động của người dùng. Theo qui định chung, những người dùng có định danh là 0 là người dùng quản trị (root). Các số định danh từ 1- 99 sử dụng cho các tài khoản hệ thống, định danh của người dùng bình thường sử dụng giá trị bắt đầu từ 100.



II.3. Mật khẩu người dùng

Mỗi người dùng có một mật khẩu riêng để sử dụng tài khoản của mình. Mọi người đều có quyền đổi mật khẩu của chính mình. Người quản trị thì có thể đổi mật khẩu của những người khác.

Unix truyền thống lưu các thông tin liên quan tới mật khẩu người dùng trong tập tin `/etc/passwd`. Tuy nhiên, mọi người dùng đều đọc được tập tin này do một số yêu cầu cho hoạt động bình thường của hệ thống (như chuyển User ID thành tên khi hiển thị trong lệnh `ls` chẳng hạn) và nhìn chung các người dùng đặt mật khẩu “yếu” do đó hầu hết các phiên bản Unix mới đều lưu mật khẩu (được mã hóa) thực sự trong một tập tin khác `/etc/shadow` và chỉ có root được quyền đọc tập tin này.

Chú ý: Theo cách xây dựng mã hóa mật khẩu, chỉ có 2 cách phá mật khẩu là vét cạn (brute force) và đoán. Phương pháp vét cạn, theo tính toán chặt chẽ, là không thể thực hiện nổi vì đòi hỏi thời gian tính toán quá lớn, còn đoán thì chỉ tìm ra những mật khẩu ngắn, hoặc “yếu”, ví dụ như những từ tìm thấy trong từ điển như `god`, `darling` ...

II.4. Group ID

Khái niệm Group ID để định danh nhóm của người dùng, thông qua Group ID này giúp ta có thể xác định người dùng đó thuộc nhóm nào, thông thường trên Linux GID được mặc định tạo ra khi ta tạo một user và có giá trị ≥ 500 .

II.5. Home directory

Khi người dùng login vào hệ thống được đặt làm việc tại thư mục cá nhân của mình. Thường thì mỗi người có một thư mục cá nhân riêng, người dùng có toàn quyền trên đó, nó dùng để chứa dữ liệu cá nhân và các thông tin hệ thống cho hoạt động của người dùng như biến môi trường, script khởi động, profile khi sử dụng X window ... Home directory của người dùng thường là `/home`; cho root là `/root`. Tuy nhiên chúng ta cũng có thể đặt vào vị trí khác thông qua lệnh `useradd` hoặc `usermod`

III. Quản lý người dùng

III.1. Tạo tài khoản người dùng

Để tạo một tài khoản, bạn có thể sử dụng lệnh `useradd`, cú pháp lệnh `useradd` như sau:

```
#useradd [-c lời_mô_tả_về_người_dùng] [-d thư_mục_cá_nhân] [-m] [-g nhóm_của_người_dùng] [tên_tài_khoản]
```

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

Ví dụ:

```
# useradd -c "Nguyen van B " nvb
```

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

```
# passwd nvb
```

```
Changing password for user nvb
```



New UNIX password: ****

Retype new UNIX password: ****

passwd: all authentication tokens updated successfully

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

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

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

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

nvb có user_ID 1012 và thuộc nhóm 1013.

Xem tập tin `/etc/group` ta thấy:

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

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

```
.....
```

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

```
.....
```

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

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

```
# useradd -D
```

```
GROUP=100
```

```
HOME=/home
```

```
INACTIVE=-1
```

```
EXPIRE=
```

```
SHELL=/bin/bash
```

```
SKEL=/etc/skel
```

III.2. Thay đổi thông tin của tài khoản

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



#usermod [-c mô_tả_thông_tin_người_dùng] [-d thư_mục_cá_nhân] [-m] [-g nhóm_của_người_dùng] [tên_tài_khoản].

Ví dụ: Cho tài khoản nvb vào nhóm admin

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

III.3. Tạm khóa tài khoản người dùng

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

Khóa (locking)	Mở khóa (unlock)
passwd -l <username>	passwd -u
usermod -L <username>	usermod -U

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

III.4. Hủy tài khoản

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

```
#userdel <option> [username]
```

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

```
#userdel -r nvb
```

IV. Nhóm người dùng

Thiết lập những người dùng có chung một số đặc điểm nào đó hay có chung quyền hạn trên tài nguyên vào chung một nhóm. Mỗi nhóm có một tên riêng và một định danh nhóm, một nhóm có thể có nhiều người dùng và người dùng có thể là thành viên của nhiều nhóm khác nhau. Tuy nhiên tại một thời điểm, một người dùng chỉ có thể là thành viên của một nhóm duy nhất.

Thông tin về nhóm lưu tại tập tin /etc/group. Mỗi dòng định nghĩa một nhóm, các trường trên dòng cách nhau bằng dấu :

```
<tên-nhóm>:<password-của-nhóm>:<định-danh-nhóm>:các-user-thuộc-nhóm>
```

IV.1. Tạo nhóm

Chúng ta có thể chỉnh sửa trực tiếp trong tập tin /etc/group hoặc dùng lệnh groupadd. Cú pháp của lệnh:

```
#groupadd [tên-nhóm]
```

IV.2. Thêm người dùng vào nhóm

Chúng ta có thể sửa từ tập tin /etc/group, các tên tài khoản người dùng cách nhau bằng dấu “;”. Một cách khác là cho từng người dùng vào nhóm bằng lệnh:

```
#usermod -g [tên-nhóm tên-tài-khoản]
```



Hay sửa thông tin tài khoản trực tiếp trong tập tin /etc/passwd thông qua việc chỉnh sửa lại định danh nhóm trong dòng khai báo tài khoản người dùng.

IV.3. Hủy nhóm

Ta có thể xóa trực tiếp nhóm trong tập tin /etc/group hay dùng lệnh:

```
#groupdel [ tên-nhóm]
```

IV.4. Xem thông tin về user và group

Ta có thể dùng lệnh groups hoặc id để xem thông tin về một tài khoản hay một nhóm nào đó trong hệ thống, cú pháp lệnh:

```
#id <option> <username>
```

Ví dụ: Ta muốn xem groupID của một user tdnhon ta dùng lệnh:

```
#id -g tdnhon
```

Ta có thể xem tên nhóm của một user nào đó ta dùng lệnh groups <username>

Ví dụ:

```
[root@server root]# groups root
```

```
root : root bin daemon sys adm disk wheel
```



BÀI 7

Quản Lý Tài Nguyên Đĩa Cứng

Tóm tắt

Lý thuyết: 3 tiết - Thực hành: 5 tiết.

Mục tiêu	Các mục chính	Bài tập bắt buộc	Bài tập làm thêm
Giới thiệu cơ chế thiết lập hạn ngạch để giới hạn tài nguyên đĩa cho người dùng.	I. Giới thiệu QUOTA II. Thiết lập QUOTA III. Kiểm tra và thống kê hạn ngạch IV. Thay đổi Grace Periods	Bài tập 7.1 (sách bài tập)	



I. Giới thiệu QUOTA

Một công cụ tốt nhất để quản lý tài nguyên đĩa cứng là quota. Quota được dùng để hiển thị việc sử dụng và giới hạn đĩa cứng đối với người dùng. Không phải áp dụng quota cho tất cả những hệ thống tập tin. Chỉ có những hệ thống tập tin nào cần thiết chúng ta mới dùng quota (ví dụ như /home - /home phải là một partition). Khi được gọi, quota sẽ đọc tập tin /etc/fstab và kiểm tra những tập tin hệ thống trong tập tin này. Để giúp cho việc giới hạn có hiệu quả, trước khi cấu hình bạn cần hiểu những khái niệm sau:

- **Giới hạn cứng(Hard Limit):** Định nghĩa dung lượng đĩa cứng tối đa mà người dùng có thể sử dụng. Nếu người dùng cố tình lưu những thông tin vào thì những thông tin trước đó có thể bị xóa và đẩy lên dần. Việc giới hạn này thật mạnh mẽ và cần thiết đối với một số người dùng.
- **Giới hạn mềm(Soft Limit):** Định nghĩa dung lượng đĩa cứng tối đa mà người dùng có thể sử dụng. Tuy nhiên, không giống như giới hạn cứng, giới hạn mềm cho phép người dùng sử dụng vượt quá dung lượng cho phép trong một khoảng thời gian nào đó. Thời gian này được xác định trước và gọi là thời gian gia hạn (grace period). Khi người dùng vượt quá dung lượng cho phép, họ sẽ nhận một lời cảnh báo trước. Một ý kiến hay là bạn cấu hình giới hạn mềm nhỏ hơn giới hạn cứng, và cấu hình khi người dùng vượt quá dung lượng cho phép hệ thống sẽ gửi một lời cảnh báo trước khi cho phép người dùng lưu dữ liệu.
- Thời gian gia hạn(Grace Period): Là thời gian cho phép người dùng vượt quá dung lượng đĩa cứng được cấp phép trong giới hạn mềm.

II. Thiết lập Quota

Quá trình thiết lập quota sẽ trải qua những bước sau:

- Chỉnh sửa tập tin /etc/fstab.
- Thực hiện quotacheck.
- Phân bổ quota.

II.1. Chỉnh sửa tập tin /etc/fstab

Mở tập tin /etc/fstab để thêm một số thông số giới hạn usrquota (cho người dùng), grpquota(cho nhóm). **Ví dụ** file /etc/fstab:

```

/dev/md0          /                ext3    defaults          1 1
LABEL=/boot      /boot           ext3    defaults          1 2
none             /dev/pts        devpts  gid=5,mode=620   0 0
LABEL=/home      /home           ext3    efaults,usrquota,grpquota 1 2
none             /proc           proc    defaults          0 0
none             /dev/shm        tmpfs   defaults          0 0
/dev/md1          swap            swap    defaults          0 0
    
```

Trong ví dụ trên, ta đặt cấu hình hạn ngạch trên hệ thống tập tin /home cho cả người dùng và nhóm bằng cách thêm các tùy chọn usrquota,grpquota (Trong đó usrquota để đặt hạn ngạch cho user và grpquota sử dụng cho nhóm).

Sau đó ta tạo các tập tin lưu trữ thông tin cấu hình cho user(aquota.user), cho nhóm(aquota.group) trong thư mục /home và đặt quyền hạn lên hai tập tin này.



```
#touch aquota.user
#chmod 600 aquota.user
#touch aquota.group
#chmod 600 aquota.group
```

Sau đó ta phải reboot lại hệ thống để remount lại file system /home thông qua lệnh init 6.

II.2. Thực hiện quotacheck

Sau khi đã cấp phép quota và gắn kết lại hệ thống tập tin, hệ thống bây giờ có khả năng làm việc quota. Tuy nhiên, những hệ thống tập tin này cũng chưa thực sự sẵn sàng, cho nên chúng ta cần dùng quotacheck. Lệnh quotacheck sẽ kiểm tra những hệ thống tập tin được cấu hình quota và xây dựng lại bảng sử dụng đĩa hiện hành.

```
#quotacheck -avug
```

Những tùy chọn:

- + -a : kiểm tra tất cả những hệ thống tập tin cấu hình quota.
- + -v : Hiển thị thông tin trạng thái khi kiểm tra.
- + -u : kiểm tra quota của người dùng.
- + -g : kiểm tra quota của nhóm.

II.3. Phân bổ quota

Người quản trị hệ thống sẽ thiết lập quota cho người dùng trong tập tin có tên aquota.user nằm trong hệ thống tập tin mà chúng ta muốn cấu hình quota. Tương tự, chúng ta cũng sẽ thiết lập quota cho nhóm trong tập tin aquota.group.

```
#edquota <option> <username>
```

Bạn có thể điều khiển lệnh quota một cách hiệu quả với những tùy chọn sau:

- + -g chỉnh sửa quota cho nhóm
- + -p sao chép quota của một người dùng cho một người dùng khác
- + -u chỉnh sửa quota cho người dùng(mặc định của lệnh)
- + -t chỉnh sửa thời gian của giới hạn mềm.

Ví dụ: #edquota -u hv

Disk quotas for user mp3user (uid 503):

```
Filesystem blocks soft hard inodes soft hard
```

```
/dev/hda3 24 0 0 7 0 0
```

- Blocks: Dung lượng(block) user đang sử dụng
- Inodes: Số lượng file user đang sử dụng.
- Soft Limit: Dung lượng giới hạn mềm (blocks/inodes), thông thường kích thước này phải <= kích thước giới hạn cứng. Nếu user sử dụng quá dung lượng này thì quota sẽ cấp một khoảng thời gian(grace periods). Khi Soft Limit bằng 0 có nghĩa giới hạn này không sử dụng.



- Hard Limit: Dung lượng giới hạn cứng (blocks/inodes)
- Sau đó ta chọn phím i để edit các thông số trên cho phù hợp, sau đó chọn phím Esc và chọn :x

Sau khi thiết lập quota, bạn phải khởi động quota lên bằng lệnh **quotaon /dev/hda3**

- Với tùy chọn -a của lệnh quotaon sẽ kiểm tra tất cả những hệ thống tập tin
- Lệnh quotaoff thì có tính năng ngược lại, tạm ngưng quota trên hệ thống tập tin.

III. Kiểm tra và thống kê hạn ngạch

Người dùng có thể dùng lệnh quota -v để xem hạn ngạch, cú pháp của lệnh:

```
#quota [tùy_chọn] [người_dùng] [nhóm]
```

Những tùy chọn của lệnh quota.

- + -g hiển thị quota của nhóm mà người dùng này là một thành viên.
- + -q chỉ hiển thị những hệ thống tập tin có thiết lập quota.
- + -u hiển thị quota của người dùng.

Ngoài ra ta có thể sử dụng quotastats, repquota để xem một số thông tin thống kê về hạn ngạch....

Ví dụ:

```
# repquota /home
*** Report for user quotas on device /dev/hda3
Block grace time: 7days; Inode grace time: 7days
Block limits      File limits
User used soft hard grace  used soft hard  grace
-----
root 52696    0    0      1015  0    0
...
...
mp3user 24    0    0         7  0    0
```

IV. Thay đổi Grace Periods

Ta có thể dùng lệnh edquota -t để thay đổi grace periods cho filesystem, đơn vị thời gian này có thể seconds, minutes, hours, days, weeks, and months. Để thay đổi thông số này sau khi ta dùng lệnh edquota -t ta dùng i để nhập giá trị (7days nếu ta muốn đặt 7 ngày) grace periods và dùng :x để lưu.

Ví dụ:

```
# edquota -t
Grace period before enforcing soft limits for users:
Time units may be: days, hours, minutes, or seconds
Filesystem Block grace period Inode grace period
/dev/hda3   7days          7days
```



BÀI 08 Cấu Hình Mạng

Tóm tắt

Lý thuyết: 5 tiết - Thực hành: 10 tiết.

Mục tiêu	Các mục chính	Bài tập bắt buộc	Bài tập làm thêm
Giới thiệu một số thao tác cấu hình mạng, các phương thức quản trị mạng từ xa, cấu hình DHCP để cấp phát địa chỉ IP động cho máy trạm.	<ol style="list-style-type: none"> I. Đặt tên máy II. Cấu hình địa chỉ IP cho NIC III. Lệnh netstat IV. Thay đổi default gateway VI. Dịch vụ Telnet. VII. Secure Remote Access – SSH (Secure Shell). VIII. Dynamic Host Configuration Protocol. 	Bài tập 8.1 (sách bài tập.)	



I. Đặt tên máy

Lệnh `hostname` dùng để xem và cấu hình tên máy tính. Khi ta dùng lệnh `hostname` không kèm theo tham số, điều này có nghĩa là ta muốn xem tên máy của hệ thống.

Tuy nhiên ta cũng có thể dùng lệnh `hostname <hostname>` để đặt tên máy cho hệ thống nội bộ, tên máy sẽ được thay đổi một khi user logoff và logon trở lại. lệnh `hostname` chỉ đặt tên máy tạm thời, khi hệ thống reboot lại thì tên máy sẽ trở về tên cũ trước đó. Thông tin về tên máy tính được lưu trong tập tin `/etc/hosts` bao gồm các thông tin sau:

```
Địa chỉ ip      <tên máy>
```

Nếu ta muốn thay đổi tên máy cố định và sẽ được lưu lại sau khi hệ thống reboot, ta sẽ thay đổi thông số `HOSTNAME=<hostname>` trong tập tin `/etc/sysconfig/network` mô tả thông tin về đường mạng:

```
NETWORKING=yes
```

```
HOSTNAME=Server
```

II. Cấu hình địa chỉ IP cho NIC

II.1. Xem địa chỉ IP

Xem thông tin địa chỉ IP của PC ta dùng lệnh `ifconfig`, lệnh này được sử dụng trên Unix/Linux. `eth0` là tên của card mạng trong, `lo` là tên của loopback interface. Ví dụ sau ta dùng lệnh `ifconfig -a` để xem thông tin cấu hình mạng trên card mạng.

```
# ifconfig -a
eth0   Link encap:Ethernet HWaddr 00:0C:29:6D:F0:3D
       inet addr:172.29.14.150 Bcast:172.29.14.159
       Mask:255.255.255.224
       inet6 addr: fe80::20c:29ff:fe6d:f03d/64 Scope:Link
       UP BROADCAST RUNNING MULTICAST MTU:1500
       Metric:1
       RX packets:6622 errors:0 dropped:0 overruns:0 frame:0
       TX packets:1425 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:1000
       RX bytes:793321 (774.7 Kb) TX bytes:240320 (234.6 Kb)
       Interrupt:10 Base address:0x1080
lo     Link encap:Local Loopback
       inet addr:127.0.0.1 Mask:255.0.0.0
       inet6 addr: ::1/128 Scope:Host
       UP LOOPBACK RUNNING MTU:16436 Metric:1
       RX packets:76 errors:0 dropped:0 overruns:0 frame:0
       TX packets:76 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:0
       RX bytes:8974 (8.7 Kb) TX bytes:8974 (8.7 Kb)
```

II.2. Thay đổi địa chỉ IP

Ta có nhiều cách thay đổi địa chỉ IP của PC trên Linux, sau đây là ba cách cơ bản nhất:

- Cách 1: Dùng lệnh `ifconfig <interface_name> <IP_address> netmask <netmask_address> up`



Ví dụ :

```
[root@bigboy tmp]# ifconfig eth0 10.0.0.1 netmask 255.255.255.0 up
```

Chú ý: Khi dùng lệnh này thay đổi địa chỉ IP thì hệ thống lưu trữ tạm thời thông tin cấu hình này trong bộ nhớ và sẽ bị mất khi hệ thống reboot lại, để cho thông tin này có thể được lưu giữ lại sau khi reboot hệ thống thì ta phải thêm lệnh trên vào tập tin /etc/rc.local.

- Cách 2: Ta có thể thay đổi thông tin cấu hình mạng trực tiếp trong file /etc/sysconfig/network-scripts/ifcfg-eth0(ta có thể dùng chương trình mc để edit file này)

```
Gán địa chỉ IP tĩnh(tham khảo file ifcfg-eth0 )
# Advanced Micro Devices [AMD]79c970 [PCnet32
LANCE]
DEVICE=eth0
BOOTPROTO=static
BROADCAST=172.29.14.159
HWADDR=00:0C:29:6D:F0:3D
IPADDR=172.29.14.150
NETMASK=255.255.255.224
NETWORK=172.29.14.128
ONBOOT=yes
TYPE=Ethernet
```

```
Gán địa chỉ IP động(tham khảo file ifcfg-eth0)
DEVICE=eth0
BOOTPROTO=dhcp
ONBOOT=yes
```

Sau đó ta dùng lệnh:

```
# ifdown eth0
```

```
# ifup eth0
```

Cách 3: Ta dùng trình tiện ích setup để cấu hình(tham khảo trình tiện ích setup trong bài học Trình Tiện Ích)

II.3. Tạo nhiều địa chỉ IP trên card mạng

Thông thường phương thức tạo nhiều địa chỉ IP trên card mạng được gọi là IP alias. Alias này phải có tên dạng: **parent-interface-name:X** , trong đó **X** là chỉ số của interface thứ cấp (subinterface number). Để tạo Alias IP ta dùng hai cách sau:

Cách 1:

- Bước 1: Đảm bảo rằng tên interface thật phải tồn tại, và kiểm tra các IP Alias trong hệ thống có tồn tại hay không.
- Bước 2:Tạo Virtual interface dùng lệnh ifconfig:

```
# ifconfig ifcfg-eth0:0 192.168.1.99 netmask 255.255.255.0 up
```

Hoặc tạo một tên file /etc/sysconfig/network-scripts/ifcfg-eth0:0 từ file /etc/sysconfig/network-scripts/ifcfg-eth0 sau đó ta thay đổi thông tin địa chỉ trong file này.

- Bước 3: Bật và tắt alias interface thông qua lệnh ifconfig

```
# ifup eth0:0
```



ifdown eth0:0

Hoặc dùng lệnh /etc/init.d/network restart

- Bước 4: Kiểm tra thông tin cấu hình alias interface dùng lệnh ifconfig:

```
# ifconfig
eth0  Link encap:Ethernet HWaddr 00:0C:29:6D:F0:3D
      inet addr:172.29.14.150 Bcast:172.29.14.159 Mask:255.255.255.224
      inet6 addr: fe80::20c:29ff:fe6d:f03d/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:7137 errors:0 dropped:0 overruns:0 frame:0
      TX packets:1641 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:848367 (828.4 Kb) TX bytes:265688 (259.4 Kb)
      Interrupt:10 Base address:0x1080
eth0:0 Link encap:Ethernet HWaddr 00:0C:29:6D:F0:3D
      inet addr:172.29.15.150 Bcast:172.29.15.159 Mask:255.255.255.224
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:7137 errors:0 dropped:0 overruns:0 frame:0
      TX packets:1641 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:848367 (828.4 Kb) TX bytes:265688 (259.4 Kb)
      Interrupt:10 Base address:0x1080
```

Cách 2:

- Tạo tập tin parent-interface-name:X bằng cách copy file /etc/sysconfig/network-scripts/ifcfg-eth0 thành file /etc/sysconfig/network-scripts/ifcfg-eth0:X (trong đó X là số thứ tự của subinterface).
- Thay đổi thông tin cấu hình mạng trong file ifcfg-eth0:X (các thông tin in đậm là thông tin bắt buộc ta phải thay đổi)

DEVICE=eth0:0

ONBOOT=yes

BOOTPROTO=static

IPADDR=172.29.14.151

NETMASK=255.255.255.224

GATEWAY=172.29.129

- Dùng lệnh **service network restart**

II.4. Lệnh netstat

Để kiểm tra trạng thái của tất cả các card mạng ta dùng lệnh:

#netstat -in

Ví dụ:

```
#netstat -in
Name Mtu Net/Dest Address Ipkts Ierrs Opkts Oerrs Collis
queue
le0 1500 172.16.0.0 172.16.12.2 1547 1 1127 0 135
0
lo0 1536 127.0.0.0 127.0.0.1 133 0 133 0 0
```



0

Ngoài ra ta còn có thể dùng lệnh `netstat -rn` để xem bảng routing table của router (nếu trong trường hợp hệ thống của ta đóng vai trò là router mềm)

Ví dụ:

```
# netstat -rn
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
172.29.15.128 0.0.0.0 255.255.255.224 U 0 0 0 eth0
172.29.14.128 0.0.0.0 255.255.255.224 U 0 0 0 eth0
169.254.0.0 0.0.0.0 255.255.0.0 U 0 0 0 eth0
1.0.0.0 0.0.0.0 255.0.0.0 U 0 0 0 eth0
0.0.0.0 172.29.14.129 0.0.0.0 UG 0 0 0 eth0
```

III. Thay đổi default gateway

Việc chỉ định địa chỉ default gateway cho hệ thống là công việc rất quan trọng vì default gateway chính là cầu nối quan trọng giúp cho hệ thống nội bộ có thể giao tiếp với hệ thống bên ngoài và ngược lại, việc đặt địa chỉ này tùy thuộc vào từng hệ thống cụ thể mà ta có địa chỉ default gateway thích hợp, để đặt địa chỉ default gateway trên Linux ta có thể dùng lệnh `route`. Thông qua lệnh này ta cũng có thể mô tả, cập nhật các con đường đi hỗ trợ việc xây dựng bảng định tuyến trên router. Ta chỉ định địa chỉ 172.29.14.150 là default gateway cho hệ thống nội bộ, ta có thể dùng lệnh sau:

```
# route add default gw 172.29.14.150
```

Ta có thể dùng lệnh `route add` để chỉ định nhiều default gateway:

```
# route add -net 10.0.0.0 netmask 255.0.0.0 gw 192.168.1.254 eth0
```

hoặc ta có thể dùng option `-host` để chỉ định cho host:

```
# route add -host 10.0.0.1 gw 192.168.1.254 eth0
```

III.1. Mô tả đường đi (route) thông qua script file

Thông thường khi ta mô tả các route cho bảng routing table cho hệ thống khi ta muốn triển khai hệ thống nội bộ như 1 router mềm thì ta dùng file `/etc/sysconfig/static-routes` hoặc có thể dùng lệnh các route add trong file `/etc/rc.d/rc.local`, tuy nhiên ta có thể làm cách khác bằng cách dùng tạo script file sau: `/etc/sysconfig/network-scripts/route-interface_name`, trong đó `interfacename` chính là tên outgoing interface. Cú pháp của file này như sau:

Destination/prefix_mask via gateway

Trong **ví dụ** sau ta thêm đường mạng 10.0.0.0 và bảng định tuyến.

```
[root@bigboy tmp]# netstat -nr
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
192.168.1.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
169.254.0.0 0.0.0.0 255.255.0.0 U 0 0 0 eth0
0.0.0.0 192.168.1.1 0.0.0.0 UG 0 0 0 eth0
[root@bigboy tmp]#
[root@bigboy tmp]# ./ifup-routes eth0 (->thực thi interface )
```



```
[root@bigboy tmp]# netstat -nr
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
192.168.1.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
169.254.0.0 0.0.0.0 255.255.0.0 U 0 0 0 eth0
10.0.0.0 192.168.1.254 255.0.0.0 UG 0 0 0 eth0
0.0.0.0 192.168.1.1 0.0.0.0 UG 0 0 0 eth0
[root@bigboy tmp]#
```

III.2. Xóa route trong bảng định tuyến

Để xóa đường đi(route) trong bảng định tuyến ta dùng lệnh **route del**

```
# route del -net 10.0.0.0 netmask 255.0.0.0 gw 192.168.1.254 eth0
```

IV. Truy cập từ xa

Khi cấu hình hệ thống kết nối vào mạng, máy chủ sẽ cung cấp một số dịch vụ Internet. Thông thường mỗi dịch vụ Internet gắn liền với một daemon và thực hiện trong chế độ background. Những daemon này hoạt động bằng cách liên kết đến một cổng nào đó và sau đó đợi những yêu cầu kết nối được gửi đến từ chương trình client. Khi một kết nối xảy ra nó sẽ tạo ra một tiến trình con đảm nhiệm kết nối này và tiếp tục lắng nghe những yêu cầu kết nối khác. Nếu như hệ thống có quá nhiều daemon sẽ làm tăng xử lý của CPU. Để khắc phục điều này, Linux tạo ra một super-server gọi là Xinetd.

IV.1. xinetd

Mỗi dịch vụ Internet đều gắn liền với một cổng chẳng hạn như: smtp – 25, pop3 – 110, dns – 53... Việc phân bổ này do một tổ chức qui định.

Xinetd là một Internet server daemon. Xinetd quản lý tập trung tất cả các dịch vụ Internet. Xinetd quản lý mỗi dịch vụ tương ứng với một cổng(port). Xinetd lắng nghe và khi nhận được một yêu cầu kết nối từ các chương trình client, nó sẽ đưa yêu cầu đến dịch vụ tương ứng xử lý. Và sau đó, Xinetd vẫn tiếp tục lắng nghe những yêu cầu kết nối khác. Khi hệ điều hành được khởi động, Xinetd được khởi tạo ngay lúc này bởi script /etc/rc.d/init.d/xinetd. Khi Xinetd được khởi tạo, nó sẽ đọc thông tin từ tập tin cấu hình /etc/xinetd.conf và sẽ dẫn đến thư mục /etc/xinetd - nơi lưu tất cả những dịch vụ mà Xinetd quản lý. Trong thư mục /etc/xinetd, thông tin cấu hình của mỗi dịch vụ được lưu trong một tập tin có tên trùng với tên dịch vụ đó. Nội dung tập tin của dịch vụ telnet cụ thể như sau:

```
service telnet
{
    disable = yes
    flags = REUSE
    socket_type = stream
    wait = no
    user = root
```



```
server = /usr/sbin/in.telnetd
log_on_failure += USERID
}
```

Những thuộc tính trong tập tin bao gồm :

Tên	Ý nghĩa
Disable	Tạm đình chỉ dịch vụ này. Có 2 giá trị: yes, no
Flags	
Socket_type	Loại socket. Trong trường hợp này là stream, stream là một loại socket cho những kết nối connection-oriented chẳng hạn như TCP
Wait	Thường chỉ liên quan đến những kết nối có loại socket là datagram. Giá trị của nó có thể là nowait, điều này có nghĩa là xinetd sẽ tiếp tục nhận và xử lý những yêu cầu khác trong lúc xử lý kết nối này. Hoặc có thể là wait nghĩa là tại một thời điểm xinetd chỉ có thể xử lý một kết nối tại một cổng chỉ định.
User	Chỉ ra user chạy dịch vụ này. Thông thường là root.
Server	Chỉ ra đường dẫn đầy đủ đến nơi quản lý dịch vụ

IV.2. Tập tin /etc/services

Khi xinetd được khởi tạo nó sẽ truy cập đến tập tin /etc/services để tìm cổng tương ứng với từng dịch vụ. Nội dung của tập tin này như sau:

```
echo          7/tcp
echo          7/udp
discard9/tcp  sink null
discard9/udp  sink null
systat 11/tcp users
systat 11/udp users
daytime      13/tcp
daytime      13/udp
qotd         17/tcp quote
qotd         17/udp quote
msp          18/tcp# message send protocol
msp          18/udp# message send protocol
chargen      19/tcp      ttytst source
chargen      19/udp      ttytst source
ftp-data     20/tcp
ftp-data     20/udp
```




```
# 21 is registered to ftp, but also used by fsp
ftp          21/tcp
ftp          21/udp      fsp fspd
ssh          22/tcp # SSH Remote Login Protocol
ssh          22/udp # SSH Remote Login Protocol
telnet       23/tcp
telnet       23/udp
```

24 - private mail system

```
smtp        25/tcp      mail
smtp        25/udp      mail
time        37/tcp      timserver
time        37/udp      timserver
rlp         39/tcp      resource # resource location
rlp         39/udp      resource # resource location
nameserver  42/tcp      name      # IEN 116
nameserver  42/udp      name      # IEN 116
```

Mỗi dòng trong tập tin mô tả cho một dịch vụ, bao gồm những cột sau:

- Cột 1: tên của dịch vụ.
- Cột 2: số cổng và giao thức mà dịch vụ này hoạt động.
- Cột 3: danh sách những tên gọi khác của dịch vụ này.

IV.3. Khởi động xinetd

Sau khi chỉnh sửa tập tin cấu hình của từng dịch vụ trong thư mục `/etc/xinetd`, ta thực hiện lệnh sau để đọc lại nội dung của tập tin cấu hình :

```
/etc/rc.d/init.d/xinetd restart
```

V. Telnet

V.1. Khái niệm telnet

Vì một lý do nào đó người dùng không thể ngồi trực tiếp trên máy Linux làm việc. Dịch vụ telnet hỗ trợ cho người dùng trong vấn đề làm việc từ xa, . Nhưng để đảm bảo tính bảo mật cho hệ thống, một điều cảnh báo là chúng ta không nên làm việc từ xa bằng telnet mà nên làm việc trực tiếp tại máy Linux.

V.2. Cài đặt

Thông thường khi cài đặt Linux, dịch vụ telnet đã được cài sẵn. Nếu chưa cài bạn có thể cài telnet server từ packet bằng dòng lệnh sau :



```
rpm -i telnet-server-0.17-20.i386.rpm
```

V.3. Cấu hình

Có nhiều cách cấu hình telnet server, sau đây là hai cách cấu hình cơ bản nhất:

- **Cách 1:** Dựa vào tập tin cấu hình, Khi cài đặt xong trong thư mục /etc/xinetd.d sẽ xuất hiện tập tin telnet. Tập tin này lưu những thông tin cấu hình về dịch vụ telnet.

service telnet

```
{
    disable= yes
    flags          = REUSE
    socket_type    = stream
    wait          = no
    user          = root
    server        = /usr/sbin/in.telnetd
    log_on_failure += USERID
}
```

Nếu disable là no thì TELNET server được khởi động, ngược lại nếu disable là yes thì TELNET server không được khởi động. Sau khi chỉnh sửa tập tin cấu hình trên ta start, stop bằng lệnh :

```
/etc/rc.d/init.d/xinetd restart
```

Hoặc dùng lệnh:

```
# service xinetd restart
```

- **Cách 2:** Cấu hình telnet Server bằng dòng lệnh: **chkconfig telnet on**

Kiểm tra telnet thông qua lệnh:

```
#netstat -algrep telnet
tcp    0    0    *:telnet    *:*    LISTEN
```

Kiểm tra telnet có được đặt như dịch vụ hệ thống:

```
# chkconfig --list | grep telnet
```

```
telnet: on
```

Dừng telnet server:

```
# chkconfig telnet off
```



V.4. Bảo mật dịch vụ telnet

1. Cho phép telnet server hoạt động trên tcp port khác

Như ta đã biết telnet traffic không được mã hóa do đó nếu ta cho telnet server hoạt động trên tcp port 23 thì không được an toàn vì thế ta có thể đặt telnet server hoạt động trên tcp port khác 23. để làm điều này ta thực hiện các bước sau:

- Bước 1. Mở tập tin /etc/services và thêm dòng.

```
# Local services
```

```
stelnet    7777/tcp          # "secure" telnet
```

- Bước 2. Chép file telnet thành file stelnet.

```
# cp /etc/xinetd.d/telnet /etc/xinetd.d/stelnet
```

- Bước 3. Thay đổi một số thông tin trong file file /etc/xinetd.d/stelnet

```
service stelnet
```

```
{
```

```
    flags      = REUSE
```

```
    socket_type = stream
```

```
    wait       = no
```

```
    user       = root
```

```
server      = /usr/sbin/in.telnetd
```

```
    log_on_failure += USERID
```

```
    disable    = no
```

```
    port       = 7777
```

```
}
```

- Bước 4. Kích hoạt stelnet thông qua lệnh chkconfig

```
# chkconfig stelnet on
```

- Bước 5. Kiểm tra hoạt động stelnet thông qua lệnh netstat.

```
# netstat -an | grep 777
```

```
tcp 0 0 0.0.0.0:7777 0.0.0.0:* LISTEN
```

Ta có thể logon vào stelnet Server thông qua lệnh:

```
# telnet 192.168.1.100 7777
```

2. Cho phép một số địa chỉ truy xuất telnet.

Ta hiệu chỉnh một số thông số sau::

```
service telnet
```

```
{
```



```

flags      = REUSE
socket_type = stream
wait       = no
user       = root
server     = /usr/sbin/in.telnetd
log_on_failure += USERID
disable    = no
only_from  = 192.168.1.100 127.0.0.1 192.168.1.200
}

```

VI. Secure Remote Access – SSH (Secure Shell)

Có rất nhiều người muốn biết mật khẩu của người dùng root để xâm nhập vào hệ thống nhằm mục đích phá hoại hệ thống hay tìm kiếm những thông tin nào đó. Chương trình telnet trong Linux cho phép người dùng đăng nhập vào hệ thống Linux từ xa, như nó có khuyết điểm của chương trình này là tên người dùng và mật khẩu gửi qua mạng không được mã hóa. Do đó, nó rất dễ bị những người khác nắm giữ và sẽ là mối nguy hiểm cho hệ thống. Phần mềm Secure Remote Access là một sự hỗ trợ mới của Linux nhằm khắc phục nhược điểm của telnet. Nó cho phép bạn đăng nhập vào hệ thống Linux từ xa và mật khẩu sẽ được mã hóa. Vì thế, SSH an toàn hơn nhiều so với telnet.

VI.1. Cài đặt SSH Server trên Server Linux

Dùng lệnh rpm để cài package openssh-server. *.rpm

```
rpm -ivh openssh-server.*.rpm
```

Tập tin cấu hình /etc/ssh/sshd_config và /etc/ssh/ssh_config. Để start hay stop server dùng lệnh sau:

```
/etc/init.d/sshd start/stop/restart
```

VI.2. Sử dụng SSH Client trên Linux

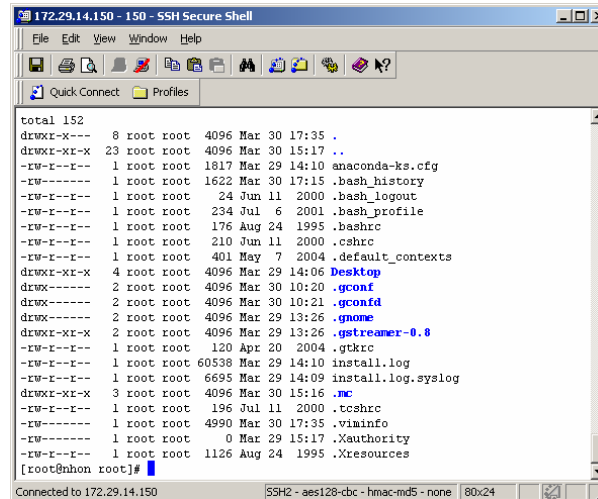
Trên client(Linux hoặc Unix) dùng lệnh ssh để login vào server. Cú pháp của lệnh:

```
$ssh [tùy_chọn] [tên/IP_máy] [tùy_chọn] [lệnh]
```

Ví dụ: \$ssh [-l] <tên_user> <ssh_address>

VI.3. Quản trị hệ thống Linux thông qua SSH client for Windows:

SSH client for Windows được thiết kế để cho phép người dùng có thể sử dụng/quản trị Unix/Linux từ hệ điều hành Windows. Ta có thể download phần mềm này từ site: <http://www.ssh.com/support/downloads/>. Phần mềm này hỗ trợ cho người dùng có thể làm việc từ xa, cung cấp dịch vụ sftp.



```

total 152
drwxr-x---  8 root root 4096 Mar 30 17:35 .
drwxr-xr-x 23 root root 4096 Mar 30 15:17 ..
-rw-r--r--  1 root root 1817 Mar 29 14:10 anaconda-ks.cfg
-rw-r--r--  1 root root 1622 Mar 30 17:15 .bash_history
-rw-r--r--  1 root root  24 Jun 11 2000 .bash_logout
-rw-r--r--  1 root root 234 Jul  6 2001 .bash_profile
-rw-r--r--  1 root root 176 Aug 24 1995 .bashrc
-rw-r--r--  1 root root 210 Jun 11 2000 .cshrc
-rw-r--r--  1 root root 401 May  7 2004 .default_contexts
drwxr-xr-x  4 root root 4096 Mar 29 14:06 Desktop
drwx----- 2 root root 4096 Mar 30 10:20 .gconf
drwx----- 2 root root 4096 Mar 30 10:21 .gconfd
drwx----- 2 root root 4096 Mar 29 13:26 .gnome
drwxr-xr-x  2 root root 4096 Mar 29 13:26 .gstreamer-0.8
-rw-r--r--  1 root root  120 Apr 20 2004 .gtkrc
-rw-r--r--  1 root root 60538 Mar 29 14:10 install.log
-rw-r--r--  1 root root 6695 Mar 29 14:09 install.log.syslog
drwxr-xr-x  3 root root 4096 Mar 30 15:16 .mc
-rw-r--r--  1 root root  196 Jul 11 2000 .tcshrc
-rw-r--r--  1 root root 4990 Mar 30 17:35 .viminfo
-rw-r--r--  1 root root  0 Mar 29 15:17 .Xauthority
-rw-r--r--  1 root root 1126 Aug 24 1995 .Xresources
    
```

Màn hình “SSH Client for Windows”

VII. Dynamic Host Configuration Protocol

DHCP là một dịch vụ hữu ích trong việc quản trị những mạng lớn hay mạng có những người dùng di động. DHCP Server là máy cấp phát địa chỉ IP cho những máy tính khác trong mạng, DHCP client là các máy nhận địa chỉ IP và những thông tin về mạng khác từ DHCP Server.

VII.1. Một số đặc điểm cần lưu ý trên DHCP Server

- Phải có một địa chỉ IP tĩnh.
- Không phải là một DHCP client.
- Cấp phát địa chỉ IP cho những máy tính trong một khoảng địa chỉ IP mà người quản trị đã định nghĩa.
- Có thể cung cấp địa chỉ default gateway, DNS server, tên domain và NetBIOS name server cho máy tính.
- Không có hai máy nhận cùng địa chỉ IP.
- Địa chỉ IP cấp cho DHCP client sẽ được làm mới khi máy tính khởi động lại.

VII.2. Ưu điểm của việc sử dụng DHCP

Người quản trị không cần đặt địa chỉ IP cho từng máy tính trong mạng

Người quản trị không cần cung cấp thông tin cho từng máy điều này tiết kiệm được thời gian và một số chi phí khác.

VII.3. Cấu hình DHCP Server

Để cấu hình DHCP server bạn cần phải cài package dhcpd.*.rpm này trong đĩa CD Linux.

Cài đặt DHCP bằng lệnh: `#rpm -ivh dhcpd.*.rpm`

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

```
ddns-update-style interim;
```



```

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

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

VII.4. Khởi động dịch vụ DHCP:

Sau khi thiết lập những tập tin cấu hình, ta cần khởi động dịch vụ bằng lệnh sau:

```
#!/etc/init.d/dhcpd start
```



BÀI 9 SAMBA

Tóm tắt

Lý thuyết: 4 tiết - Thực hành: 5 tiết.

Mục tiêu	Các mục chính	Bài tập bắt buộc	Bài tập làm thêm
Giới thiệu cơ chế chia sẻ tài nguyên trên hệ thống Linux thông qua dịch vụ SAMBA.	<ol style="list-style-type: none">I. Cài đặt SAMBAII. Khởi động dịch vụ SAMBAIII. Cấu hình Samba ServerIV. Sử dụng SAMBA SWATV. Khởi động Samba ServerVI. Sử dụng SMB clientVII. Mount thư mục chia sẻVIII. Mount tự động tài nguyên từ SMB ServerIX. Mã hoá mật khẩu	Bài tập 9.1 (sách bài tập)	



Samba là chương trình tiện ích hỗ trợ việc chia sẻ tài nguyên từ hệ thống Linux với các hệ thống khác(Linux, Windows), nó hỗ trợ tính năng gia nhập(join) Linux với Windows như gia nhập Linux vào PDC trên Windows, gia nhập vào Windows Workgroup,...

Bộ Samba gồm nhiều thành phần. Daemon mang tên smbd cung cấp dịch vụ in ấn và tập tin. Tập tin cấu hình của Daemon này là smb.conf, còn daemon nmbd thì hỗ trợ dịch vụ tên NETBIOS, cho phép các máy tính khác truy cập và sử dụng các tài nguyên được cấp bởi máy chủ Samba

Trình smbclient, một thành phần khác của bộ Samba, hoạt động như một client bình thường giống như ftp. Trình tiện ích này dùng khi bạn truy cập những tài nguyên trên các server tương thích khác.

I. Cài đặt SAMBA

Bạn có thể cài đặt Samba trong quá trình cài Linux hoặc cài sau bằng tiện ích RPM, các bộ này được tích hợp vào Fedora CD, các file này bao gồm:

- system-config-samba-1.2.15-0.fc2.1 ; hỗ trợ cấu hình trên giao diện Xwindows
- samba-3.0.7-2.FC2 ; package chính của SAMBA.
- samba-client-3.0.7-2.FC2 ; package cho SAMBA Client.
- samba-common-3.0.7-2.FC2 ; hỗ trợ các thư viện cho SAMBA.
- samba-swat-3.0.7-2.FC2 ; hỗ trợ cấu hình SAMBA qua Web.

II. Khởi động dịch vụ SAMBA

Bạn có thể khởi động dịch vụ samba tại thời điểm boot của hệ thống chkconfig.

```
# chkconfig smb on
```

Ta có thể start/stop/restart samba thông qua lệnh:

```
# service smb restart
```

Để kiểm tra samba có hoạt động trong hệ thống hay không

```
# pgrep smb
```

III. Cấu hình Samba Server

Tập tin cấu hình /etc/samba/smb.conf. Đây là một tập tin có dạng text. Các thành phần trong file cấu hình:

Thành phần	Giải thích
[global]	Chứa các tham số cấu hình chung của samba server.
[printers]	Chứa các tham số sử dụng cho việc cấu hình máy in.
[homes]	Chỉ định SMB chia sẻ thư mục home directory của user.
[netlogon]	Chia sẻ logon script.
[profile]	Chia sẻ profile.



III.1. Đoạn [global]

Đoạn này kiểm soát tất cả tham số cấu hình chung của server smb. Đoạn này cũng cung cấp giá trị mặc định cho những đoạn khác:

[global]

workgroup = LINUX ; chỉ ra nhóm mà máy này sẽ tham gia

server string = Samba Server ;

hosts allow = 192.168.1. 192.168.2. 127. ; host được phép truy xuất đến samba.

Guest account = pguest ; cung cấp username cho một account khách trên server của bạn. Account này để nhận diện những user nào được dùng các dịch vụ samba dành cho khách

Log file = /var/log/samba/smb.%m ; xác định vị trí tập tin log của từng client truy cập samba.

Max log size = 50 ; kích thước tối đa của một tập tin log (tính bằng kb)

encrypt passwords = yes ; cần hay không cần mã hoá password khi đăng nhập vào máy chủ Samba. Mọi password gửi từ Windows 9x đều mã hoá. Do đó, nếu ta chọn “no” thì máy chủ samba sẽ không chấp nhận sự đăng nhập của bất kỳ user nào. Nếu giá trị là “yes” thì chỉ có các user có password trong tập tin /etc/samba/password là có thể thấy máy chủ Samba.

smb passwd file = /etc/samba/smbpasswd ; tập tin lưu trữ những user được phép truy cập đến server smb. Một số biến cần tham khảo:

Tên biến	Mô tả giá trị
%S	Tên của dịch vụ hiện hành, nếu có
%P	Thư mục gốc của dịch vụ hiện hành, nếu có
%u	tên user của dịch vụ hiện hành
%g	tên của nhóm chính của %u
%U	tên phiên làm việc của user
%G	tên của nhóm chính của %U
%H	thư mục gốc của user
%v	phiên bản của Samba
%h	tên của host mà Samba đang chạy
%m	tên NETBIOS của máy khách
%L	tên NETBIOS của máy chủ
%M	tên Internet của máy khách
%I	Địa chỉ IP của máy khách
%T	ngày và giờ hiện hành
%a	kiến trúc của máy từ xa. Chỉ có một số máy được nhận diện là Win9x, WinNT, Win2k

III.2. Đoạn [homes]

Mặc định SMB chia sẻ home của từng người dùng trong hệ thống để cho phép các user có thể truy xuất vào home directory của mình từ máy trạm.

```
[homes]
comment = Home Directories ;
path = %H ;
read only = no ;
```



```
valid users = %S ; Chỉ định tên user được phép truy xuất,
nếu ta cho phép group ta dùng cú pháp @group_name.
browseable = no ;
writeable = yes ;
create mask = 0750 ;
```

III.3. Chia sẻ máy in dùng SMB

Để chia sẻ máy in, ta mô tả đoạn [printers] trong file /etc/smb.conf

```
[printers]
comment = All Printers
path = /var/spool/samba
browseable = no
public = yes
guest ok = no
writable = no
printable = yes ; cho phép in
create mask = 0700
```

III.4. Chia sẻ thư mục

Sau khi lập cấu hình mặc định cho server Samba, bạn có thể tạo ra nhiều thư mục dùng chung, và quyết định xem cá nhân nào, hoặc group nào được phép sử dụng chúng.

```
[dirshare]
comment = "chia sẻ thư mục"
path = /usr/local/share
valid users = hv1
browseable = yes
public = no
writable = yes
```

Đoạn trên đã tạo ra một thư mục chia sẻ mang tên dirshare. Đường dẫn đến thư mục này là /usr/local/share. Vì public là no nên chỉ có user hv1 được truy cập đến thư mục này.

IV. Sử dụng SAMBA SWAT

Swat là một công cụ cho phép bạn có thể cấu hình SAMBA qua giao diện Web. Nếu ta muốn sử dụng công cụ này thì ta phải cài thêm package samba-swat-3.0.7-2.FC2.rpm (trong Fedora Core).

IV.1. Tập tin cấu hình SAMBA SWAT

Trước khi cấu hình SAMBA-SWAT ta cần thiết lập một số thông số

disable = no

only_from = 172.29.14.149 localhost

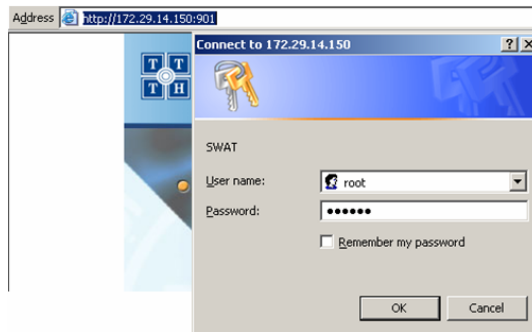
Trong file /etc/xinetd.d/swat để khởi động dịch vụ SWAT và cho phép các host nào có quyền truy xuất SAMBA SWAT qua Web.

```

service swat
{
    disable = no
    port      = 901
    socket_type = stream
    wait      = no
    only_from = 172.29.14.149 localhost
    user      = root
    server    = /usr/sbin/swat
    log_on_failure += USERID
}
    
```

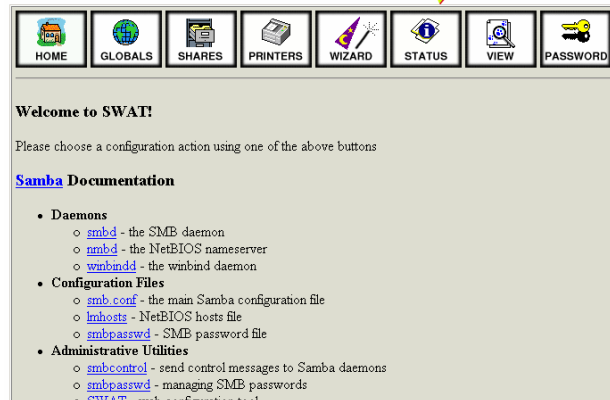
IV.2. Truy xuất SWAT từ Internet Explorer

Từ IE ta truy xuất SMB SWAT thông qua địa chỉ http://172.29.14.150:901, Sau đó ta chỉ định username(root nếu ta muốn quản lý SMB), và mật khẩu để đăng nhập:



Màn hình đăng nhập

Sau khi đăng nhập thành công



Giao diện Samba SWAT

IV.3. Cấu hình SAMBA SWAT

Thành phần	Giải thích
 HOME	Cung cấp các tài liệu tham khảo về samba.
 GLOBALS	Quản lý thông tin cấu hình.
 SHARES	Quản lý tài nguyên chia sẻ
 PRINTERS	Quản lý việc chia sẻ máy in
 WIZARD	Quản lý Server Type, Wins và một số tham số khác.
 STATUS	Quản lý trạng thái của SAMBA, theo dõi các connection...
 VIEW	Xem các thông tin cấu hình trong file smb.conf
 PASSWORD	Quản lý mật khẩu

V. Khởi động Samba Server

Server Samba gồm 2 daemon `smbd` và `nmbd`. Để khởi động samba server ta dùng script sau:
`/etc/init.d/smb {start | stop | restart | status}`

VI. Sử dụng SMB client

Từ dấu nhắc lệnh của shell ta sử dụng `smbclient` để truy xuất thư mục chia sẻ trên SMB Server theo cú pháp sau: `Smbclient <//SMB_ServerName/Sharename> <option> <username>`

Ví dụ:

```
[root@nhon xinetc.d]# smbclient //nhon/data -U hv
```

```
Password: ****
```

```
Domain=[NHON] OS=[Unix] Server=[Samba 3.0.7-2.FC2]
```

```
smb: \>
```

Từ dấu nhắc lệnh này, bạn có thể ra bất kỳ lệnh nào được liệt kê ở Bảng sau để thực thi cơ chế download/upload từ tài nguyên chia sẻ:

Lệnh	Tham số	Mô tả
? hoặc help	[Lệnh]	xem giúp đỡ của lệnh
!	[lệnh dạng shell]	thực thi lệnh shell hoặc đưa user về dấu nhắc shell
Cd	[Thư mục]	Chuyển về thư mục trên server



Lcd	[Thư mục]	Chuyển về thư mục máy cục bộ
Del	[Các tập tin]	Xóa tập tin
Dir hoặc ls	[Các tập tin]	Liệt kê các tập tin được chọn
Exit hoặc quit	Không có	Thoát khỏi chương trình smbclient
Get	[tập tin][tên cục bộ]	Sao chép tập tin trên máy server về máy cục bộ. Nếu tên cục bộ không chỉ ra sẽ lấy tên tập tin cũ trên máy server
Mget	[các tập tin]	Sao chép tất cả các tập tin được xác định vào máy cục bộ.
Md hoặc mkdir	[thư mục]	Tạo thư mục trên máy server
Rd hoặc rmdir	[thư mục]	Xóa thư mục trên máy server.
Put	[tập tin]	Sao chép tập tin từ máy cục bộ vào máy server
Mput	[các tập tin]	Sao chép tất cả tập tin từ máy cục bộ vào máy server
Print	[tập tin]	In tập tin trên máy server
Queue	Không có	Liệt kê tất cả các công việc in ấn đang xếp hàng chờ trên máy server

VII. Mount thư mục chia sẻ

Ta có thể ánh xạ một thư mục chia sẻ trên SAMBA Server vào ổ đĩa cục bộ thông qua lệnh smbmount. Cú pháp lệnh:

```
[root@bigboy tmp]# mount -t smbfs -o username=username,password=password
winclient/cdrom /mnt/cdrom
```

Ví dụ:

```
[root@nhon xinetc.d]# smbmount //nhon/data /mnt/smb -o username=hv,password=hv
```

VIII. Mount tự động tài nguyên từ SMB Server

Để tự động mount một tài nguyên chia sẻ ta thực hiện các bước sau:

- **Bước 1:** Tạo một thư mục mount point (ví dụ /mnt/smb)
- **Bước 2:** mô tả dòng sau đây vào file /etc/fstab

```
//SMB_Server/share_name /mnt/smb smbfs credentials=/etc/cred 0 0
```

- **Bước 3:** Tạo file /etc/cred để mô tả thông tin username và mật khẩu.

```
username = <username>
```

```
password = <password>
```

- **Bước 4:** Dùng lệnh mount -a để update file /etc/fstab và kiểm tra.



IX. Mã hoá mật khẩu

Mặc định giao thức SMB của Microsoft sử dụng password không mã hóa (plain text). Tuy nhiên, Windows 2K (SP 3 trở lên) yêu cầu password mã hóa. Do đó, hoặc là bạn chỉnh lại Registry của Windows để sử dụng password không mã hóa. Lúc này, bạn phải chỉnh Registry của tất cả các máy Windows. Điều này thật bất tiện và có nguy cơ đem lại một số xung đột và rất có thể sai sót hoặc là bạn cấu hình lại Samba chấp nhận password mã hoá. Bạn cần làm theo các bước sau:

- **Bước 1:** Tạo một tập tin mật khẩu riêng cho Samba. Từ tập tin `/etc/passwd` có sẵn, tạo một tập tin mới bằng cách dùng lệnh:

```
cat /etc/passwd | mksmbpasswd.sh > /etc/samba/smbpasswd
```

với `mksmbpasswd` là một script đã cài sẵn trong hệ thống

- **Bước 2:** Dùng lệnh:

```
chmod 600 /etc/samba/smbpasswd
```

để chỉ cấp quyền đọc và ghi cho root

- **Bước 3:** Người dùng chưa được sử dụng samba khi người dùng đó chưa được cấp password và ghi vào tập tin trên. Bạn dùng lệnh dưới đây để cấp password cho user

```
smbpasswd <username>
```

với `username` là định danh của người dùng đó

- **Bước 4:** Chỉnh lại tập tin `smb.conf` như sau:

```
encrypt password = yes
```

```
smb passwd file = /etc/samba/smbpasswd
```

Khởi động lại dịch vụ samba dùng lệnh **`/etc/init.d/smb restart`**.



BÀI 10

Network File System

Tóm tắt

Lý thuyết: 3 tiết - Thực hành: 5 tiết.

Mục tiêu	Các mục chính	Bài tập bắt buộc	Bài tập làm thêm
Giới thiệu dịch vụ NFS để thực thi cơ chế ánh xạ tài nguyên chia sẻ thành filesystem cục bộ.	<ol style="list-style-type: none">I. Tổng quan về quá trình hoạt động của NFSII. Cài đặt NFSIII. Cấu hình NFS	Bài tập 10.1 (sách bài tập)	



I. Tổng quan về quá trình hoạt động của NFS

NFS là dịch vụ hỗ trợ cơ chế chia sẻ tài nguyên giữa các máy chủ Linux. NFS được phát triển để cho phép hệ thống nội bộ có thể truy xuất một thư mục trên hệ thống máy khác bằng cách mount nó vào hệ thống tập tin cục bộ, người quản trị trên NFS Server chỉ cần xuất (export) các thư mục để cung cấp cho các NFS Client sử dụng.

I.1. Một số luật chung khi cấu hình NFS

- Export các thư mục con của thư mục “/”.
- Không xuất những thư mục con của những thư mục cha đã được export trước đó.
- Chỉ được export hệ thống tập tin cục bộ.

I.2. Một số khái niệm chính về NFS

- **Virtual filesystem (VFS) interface:** là một kỹ thuật tự động chuyển hướng tất cả các truy xuất đến NFS-mount file một cách thông suốt trên Remote Server. VFS giúp biến đổi yêu cầu định dạng file phù hợp trên NFS Server.
- **Stateless Operation:** là những chương trình đọc và ghi file trên hệ thống tập tin cục bộ dựa vào hệ thống để theo dõi và ghi nhận vị trí đọc dữ liệu thông qua con trỏ địa chỉ pointer. Khi NFS Server không còn hoạt động (hoặc bị lỗi) thì NFS Client sẽ thiết lập lại giá trị cho pointer là 0 và NFS Client có thể phát hiện (detect) khi NFS Server hoạt động trở lại.
- **Caching:** trên NFS Client để lưu lại một số dữ liệu cần thiết vào hệ thống cục bộ, điều này làm giảm lưu lượng truy xuất trên NFS Server.
- **NFS Background Mounting:** NFS Client sử dụng RPC để mount file trên remote server, nếu Remote Server không tồn tại thì ta có thể dùng lệnh mount đặt tùy chọn bg để chỉ định khoảng thời gian đợi trong 1 tuần.
- **Hard and Soft Mounts:** Hard mount có ý nghĩa rằng quá trình mount file sẽ luôn luôn được tiến hành trên foreground hoặc background để đảm bảo tính thống nhất dữ liệu. Soft mount là quá trình sử dụng RPC để mount remote file system, một khi RPC bị lỗi và lặp lại nhiều lần dẫn tới hoạt động của NFS bị fail dẫn tới sự thống nhất dữ liệu không được đảm bảo.
- **NFS Versions:** NFS hiện tại có 3 phiên bản 2, 3, and 4. Đối với Version 2 hỗ trợ kích thước tới 4GB, bị giới hạn 8 KB trong mỗi lần đọc và ghi dữ liệu. NFS Version 3 hỗ trợ kích thước file tới 264 – 1 bytes, có khả năng điều chỉnh kích thước việc đọc/ghi dữ liệu giữa NFS Client và NFS Server. NFS Version 4 tương tự như NFS Version 3 nhưng được tích hợp thêm một số tính năng như lock file và mount file được được tích hợp vào NFS Daemon và được thực hiện một cách độc lập.
- Các NFS Daemons quan trọng như: Portmap là Daemon quan trọng quản lý kết nối cho ứng dụng, Portmap listen trên TCP port 111, ngoài ra còn có NFS Daemon, NFSLOCK Daemon, NETFS Daemon.

II. Cài đặt NFS

NFS được cài đặt mặc định trên Redhat Linux, mặc định NFS được hoạt động khi hệ thống khởi động, ta có thể dùng một số lệnh sau đây để kiểm tra NFS được cài đặt trong hệ thống:



```
# rpm -qa | grep nfs
redhat-config-nfs-1.1.3-1      ; kết quả hiển thị
nfs-utils-1.0.1-3.9          ; kết quả hiển thị
# rpm -q portmap
portmap-4.0-57                ; kết quả hiển thị
```

III. Cấu hình NFS

III.1. Cấu hình NFS Server

Cả hai NFS Server và NFS Client đều phải cài NFS package. Trên NFS Server cần phải có các daemon portmap, nfs, and nfslock, sau đó ta tiến hành cấu hình NFS trong file /etc/exports

```
#/etc/exports
- Dòng 1: /data/files      *(ro,sync)
- Dòng 2: /home            192.168.1.0/24(rw,sync)
- Dòng 3: /data/test      *.my-site.com(rw,sync)
- Dòng 4: /data/database  192.168.1.203/32(rw,sync)
```

Giải thích một số ví dụ về cấu hình NFS trên file /etc/exports

- Dòng 1: Chỉ được đọc trên /data/files từ bất kỳ mạng nào.
- Dòng 2: Read/Write trên thư mục /home từ tất cả các máy trên mạng 192.168.1.0
- Dòng 3: Read/Write trên thư mục /data/test từ tất cả các máy trong miền my-site.com
- Dòng 4: Read/Write trên thư mục /data/database từ máy 192.168.1.203

Sau khi ta cấu hình xong ta phải reactive lại NFS server để cập nhật lại thông tin cấu hình.

III.1.1 Khởi động NFS Server:

Đặt tính năng hệ thống cho các dịch vụ:

```
# chkconfig --level 35 nfs on
# chkconfig --level 35 nfslock on
# chkconfig --level 35 portmap on
Khởi tạo các dịch vụ liên quan.
#service portmap start
#service nfs start
# service nfslock start
```

III.1.2 Kiểm tra hoạt động NFS

Ta có thể dùng lệnh rpcinfo để kiểm tra danh sách các portmapper đã được đăng ký trên host.

```
# rpcinfo -p localhost
```



```

program vers proto  port
100000  2  tcp  111 portmapper
100000  2  udp  111 portmapper
100003  2  udp  2049 nfs
100003  3  udp  2049 nfs
100021  1  udp  1024 nlockmgr
100021  3  udp  1024 nlockmgr
100021  4  udp  1024 nlockmgr
100005  1  udp  1042 mountd
100005  1  tcp  2342 mountd
100005  2  udp  1042 mountd
100005  2  tcp  2342 mountd
100005  3  udp  1042 mountd
100005  3  tcp  2342 mountd
    
```

III.2. Cấu hình NFS Client

Cấu hình mount NFS tự động thông qua file /etc/fstab ta thực hiện các bước sau:

- **Bước 1.** Khi cấu hình NFS Client ta phải khởi động NFS.

```

# chkconfig --level 35 netfs on
# chkconfig --level 35 nfslock on
# chkconfig --level 35 portmap on
#service portmap start
# service netfs start
# service nfslock start
Kiểm tra hoạt động của NFS
# rpcinfo -p
    program vers proto  port
    100000  2  tcp  111 portmapper
    100000  2  udp  111 portmapper
    100024  1  udp  32768 status
    100024  1  tcp  32768 status
    100021  1  udp  32769 nlockmgr
    100021  3  udp  32769 nlockmgr
    
```



```
100021 4 udp 32769 nlockmgr
100021 1 tcp 32769 nlockmgr
100021 3 tcp 32769 nlockmgr
100021 4 tcp 32769 nlockmgr
391002 2 tcp 32770 sgi_fam
```

- **Bước 2.** Mount một tài nguyên từ NFS Server bằng cách dùng /etc/fstab

```
MountPoint          Type Options Dump FSCK
```

```
192.168.1.100:/data/files /mnt/nfs nfs soft,nfsvers=2 0 0
```

- **Bước 3.** Thực hiện lệnh mount -a để thực thi file /etc/fstab

```
# mkdir /mnt/nfs
```

```
# mount -a          ; cập nhật lại file exports
```

```
# ls /mnt/nfs
```

```
ISO ISO-RedHat kickstart RedHat
```

Mount NFS file thông qua lệnh:

```
# mount -t nfs 192.168.1.100:/data/files /mnt/nfs
```

```
# ls /mnt/nfs
```

```
ISO ISO-RedHat kickstart RedHat
```

III.3. Kích hoạt file /etc/exports

Khi ta thay đổi cấu hình trong file /etc/exports thì ta phải restart lại NFS.

```
# exportfs -a
```

Export chỉ có entry mới trong file /etc/exports dùng lệnh

```
# exportfs -r
```

Xóa hay thay đổi một thư mục đã chia sẻ qua NFS ta phải umount thư mục đó bằng lệnh umount sau đó sửa đổi lại tập tin /etc/fstab, sau đó tiến hành reload lại NFS bằng lệnh exportfs -ua.

```
# umount /mnt/nfs
```

```
# exportfs -ua
```

```
# exportfs -a
```

III.4. Troubleshooting NFS Server

Để theo dõi và xử lý các sự cố trên NFS ta thực hiện một số lệnh sau:

- Liệt kê các export directory:

```
#showmount -a
```

- Liệt kê các mounting file system



#df -F nfs

- Thống kê lỗi trên NFS

#nfsstat -s



BÀI 11

LẬP TRÌNH SHELL TRÊN LINUX

Tóm tắt

Lý thuyết: 5 tiết - Thực hành: 5 tiết.

Mục tiêu	Các mục chính	Bài tập bắt buộc	Bài tập làm thêm
Giới thiệu các chương trình SHELL phổ biến trên Linux, đặc điểm của các chương trình SHELL, lập trình shell script để tự động hóa thao tác quản trị.	<ol style="list-style-type: none">I. Giới thiệu về SHELL Và Lập Trình SHELLII. Mục đích và ý nghĩa của việc lập trình ShellIII. Điều khiển Shell từ dòng lệnhIV. Điều khiển tập tin lệnhV. Cú pháp ngôn ngữ Shell	Bài tập 11.1 (Sách bài tập)	



I. Giới thiệu về SHELL Và Lập Trình SHELL

I.1. Giới thiệu về Shell

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

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

I.1.1 Một số đặc điểm của shell

- Xử lý tương tác (Interactive processing) : Người dùng tương tác với shell dưới dạng đối thoại trực quan.
- Chạy nền : Các chương trình trên shell có thời gian thực thi lâu và chiếm ít tài nguyên có thể cho phép chạy nền bên dưới trong khi đó người dùng có thể thực hiện các công việc khác. Điều này tăng hiệu quả sử dụng hệ thống.
- Chuyển hướng (Redirection): Có thể linh hoạt chuyển đổi các dữ liệu ra vào chuẩn và lỗi.
- Ống dẫn (pipe): Cho phép thực hiện nhiều lệnh liên tiếp trong đó dữ liệu ra của lệnh này được sử dụng như dữ liệu vào của lệnh kia.
- Tập tin lệnh (shell script): Tạo các tập tin chứa các lệnh làm việc theo trình tự. Cấp quyền và thực thi tập tin này.
- Biến shell: shell hỗ trợ sử dụng các biến lưu trữ các thông tin để điều khiển hoạt động.
- Sử dụng lại các lệnh đã thực hiện (history command). Đây là tính năng rất có ích cho người dùng. Để thực hiện lại các lệnh mình đã thực hiện trước đó thay vì phải gõ lại.
- Cấu trúc lệnh như ngôn ngữ lập trình: Shell cho phép sử dụng lệnh như ngôn ngữ lập trình, bởi nó có thể kết hợp xử lý các tác vụ phức tạp.
- Tự động hoàn tất tên tập tin, hoặc lệnh : Chúng ta có thể gõ phần đầu của lệnh hoặc tập tin sau đó dùng <Tab> để hoàn tất phần còn lại.
- Bí danh cho lệnh (command alias). Bạn có thể dùng một tên mới cho một lệnh. Sau đó sử dụng tên này thay thế lệnh : \$alias dir='ls -l'. Lúc này ta sử dụng lệnh dir dùng như ls -l

I.1.2 Các shell trong Linux.

Tên shell	Lịch sử ra đời
sh (Bourne)	Shell nguyên thủy trong Unix
Csh, tcsh và zsh	Shell sử dụng cấu trúc lệnh của ngôn ngữ C làm ngôn ngữ script. Shell này được tạo bởi Bill Joy, đây là shell thông dụng thứ 2 sau bash
Bash	Bash(bourne Again shell)là shell sử dụng chính trong Linux, ra đời từ dự án GNU. Bash có ưu điểm là mã nguồn mở, có thể download



	từ địa chỉ http://www.gnu.org
Rc	Là shell mở rộng của c shell với nhiều tương thích với ngôn ngữ C, ra đời từ dự án GNU

Shell bash là shell mặc định trên Linux, ta có thể dùng lệnh `#echo` để xem tên shell sử dụng hiện tại của hệ thống.

```
#echo $SHELL
```

1.2. Lập cấu hình môi trường đăng nhập

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

- Thành phần thứ nhất gọi là môi trường terminal để điều khiển terminal (chính là màn hình và bàn phím) của bạn.
- Thành phần thứ hai gọi là môi trường shell để điều khiển nhiều khía cạnh khác nhau của shell, cùng với mọi chương trình bạn thực hiện.

1.2.1 Thiết lập môi trường terminal

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

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



1.2.2 Thiết lập môi trường Shell

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

Biến	Mô tả
HOME=/home/đăng-nhập	HOME lập home directory của bạn. Đăng-nhập là ID đăng nhập. Ví dụ, nếu ID đăng nhập của bạn là jack, thì HOME sẽ là /home/jack
LOGNAME=đăng-nhập	Máy sẽ tự động lập LOGNAME bằng ID đăng nhập của bạn
PATH=đường-dẫn	Tùy chọn đường-dẫn trỏ đến danh sách các thư mục mà shell sẽ duyệt qua để tìm lệnh. Ví dụ, bạn có thể lập đường dẫn như sau: PATH=/usr:/bin:/usr/local/bin
PS1=dấu-nhắc	PS1 là dấu nhắc shell đầu tiên để yêu cầu bạn xác định hình dáng của dấu nhắc riêng theo ý của mình. Nếu bạn không có thay đổi gì dấu nhắc mặc định sẽ là dấu \$(cho người dùng không phải là root). Bạn có thể thay đổi, chẳng hạn như: PS1=Enter Command >
PWD=thư-mục	Xác định vị trí của bạn trong hệ thống tập tin
SHELL=shell	SHELL xác định shell mà bạn đang sử dụng.
TERM=loại-terminal	Kiểu terminal bạn dùng

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

1.2.3 Sử dụng các biến Shell đặc biệt

Biến HOME: luôn xác định home directory của bạn. Khi vừa đăng nhập thành công, bạn ở ngay trong home directory.

- Muốn trở về home directory của mình, bạn chỉ cần gõ lệnh cd.
- Bạn có thể dùng biến HOME khi biên soạn shell script để xác định những tập tin trong home directory.
- \$HOME luôn đại diện cho home directory của bất kỳ ai sử dụng lệnh. Nếu bạn gõ lệnh bằng \$HOME thì những người khác cũng có thể dùng chung lệnh.



Biến PATH: Liệt kê các thư mục mà shell sẽ đến tìm những câu lệnh. Shell tìm các thư mục theo thứ tự đã liệt kê.

Ví dụ: Nếu PATH=/bin:/usr/bin: Mỗi khi thông dịch một câu lệnh, shell sẽ tìm trước tiên trong thư mục /bin. Nếu chưa phát hiện ra lệnh cần tìm, shell tiếp tục duyệt sang thư mục /usr/bin. Nếu vẫn chưa có kết quả, shell lại dò sang thư mục (thư mục hiện hành). Chúng ta nên xếp tất cả các shell script của mình vào một thư mục và ghi vào biến PATH. Như thế, sau này cho dù bạn đang ở thư mục nào thì cũng thực thi được những shell script đó.

Biến MAIL: Chứa tên tập tin lưu trữ email của bạn. Mỗi khi nhận email, hệ thống sẽ đưa vào tập tin do biến MAIL xác định. Nếu bạn có chương trình thông báo mỗi khi có mail đến, chương trình này sẽ liên hệ với tập tin kết hợp với biến MAIL. Biến PS1: chứa những chuỗi kí tự mà bạn nhìn thấy tại dấu nhắc sơ khởi.

Biến TERM: Dùng để nhận dạng loại terminal. Những chương trình nào chạy ở chế độ toàn màn hình, ví dụ như vi, sẽ tham khảo biến TERM

Biến LOGNAME: Chứa chuỗi kí tự mà hệ thống dùng để nhận dạng ra user đăng nhập. Biến này còn giúp hệ thống biết bạn là chủ sở hữu các tập tin và thư mục, là người ra lệnh chạy một số chương trình, và là tác giả của email gửi bằng lệnh write.

II. Mục đích và ý nghĩa của việc lập trình Shell

Shell là lớp vỏ bên ngoài hạt nhân, là phần trung gian cho người dùng thao tác với hạt nhân. Bạn đã rất quen thuộc với các shell trong DOS như command.com sẽ dịch các lệnh như del, copy, ... thành những ngắt cấp thấp của hệ điều hành DOS để thực hiện. Ngoài ra DOS còn cho chúng ta tạo các tập tin .bat gồm nhiều lệnh thực hiện trình tự. Shell trong DOS nói chung còn rất đơn giản và không sử dụng nhiều các tác vụ hệ thống. Linux cung cấp các shell phong phú, uyển chuyển hơn. Nó cho phép bạn tạo những tập tin dạng bat với cấu trúc lặp như C, hay có thể sử dụng phối hợp nhiều lệnh shell với nhau.

Ví dụ: bạn có thể kết hợp lệnh ls và more để xem danh sách các tập tin thư mục theo từng trang.

```
ls -l | more
```

Linux cho phép kết hợp dữ liệu vào ra giữa các lệnh với nhau thông qua cơ chế chuyển tiếp (redirect) và ống dẫn (pipe). Ngoài ra, Linux cho phép sử dụng các lệnh có cấu trúc giống C như if, case, for ... Đây là điểm mạnh của shell trong Linux. Với các cấu trúc điều khiển như vậy chúng ta xử lý được nhiều trường hợp bằng cách kết hợp các lệnh shell với các điều kiện xử lý. Ngoài ra shell còn hỗ trợ chế độ ra vào dữ liệu, tương tác các biến môi trường.

Những chương trình shell sẽ giúp người dùng sử dụng và quản lý hệ thống và dịch vụ trên Linux. Ví dụ như khởi động hay ngưng một ứng dụng, bạn có thể viết một đoạn chương trình shell thực hiện tác vụ này. Chính sự đa dạng trong shell cho phép người dùng tạo ra chương trình shell quản lý dịch vụ hệ thống một cách hiệu quả.

III. Điều khiển Shell từ dòng lệnh

Người dùng có thể sử dụng các lệnh shell từ dòng lệnh. Khi người dùng chưa hoàn tất lệnh thì shell hiển thị dấu > để chúng ta thêm vào.

Ví dụ:



```
$ if [ $file -d ] ;
    > echo ls $file
> else echo "$file is not file"
> fi
```

Chúng ta sử dụng nhiều lệnh trên một dòng cách nhau bằng dấu chấm phẩy (;)

Ví dụ:

```
cd /etc ; ls -l
```

Bạn chỉ cần gõ Enter thì sẽ thực hiện các lệnh trên dòng đó. Điều bất tiện nhất khi sử dụng trên dòng lệnh là khả năng sửa chữa lỗi khi chúng ta nhầm lẫn. Do vậy người ta thường ghi các lệnh vào trong tập tin, rồi cho nó thực hiện tuần tự. Tập tin chứa các lệnh này được gọi là tập tin lệnh hay các shell script.

IV. Điều khiển tập tin lệnh

Tập tin lệnh có thể được thực thi theo 2 cách.

Cách 1: Bạn gọi shell và dùng tập tin là tham số :

```
$ /bin/sh tên-tập-tin.
```

Ví dụ: \$/bin/sh hello.

Cách 2: Bạn sẽ gọi tập tin lệnh từ dấu nhắc của shell như thực hiện các lệnh thông thường. Theo cách này, trước hết bạn phải cấp quyền thực thi (execute) trên tập tin này. Tùy theo nhu cầu sử dụng tập tin lệnh bạn có thể cấp quyền cho người sở hữu, cho nhóm sở hữu hay cho mọi người. Lệnh cấp quyền như chúng ta đã học là chmod. Lệnh cấp cho mọi người có quyền thực thi :

```
chmod +x <tên-tập-tin>
```

Chỉ cho người sở hữu thực thi :

```
chmod o+x tên-tập-tin
```

Chạy tập tin lệnh: Bạn gõ lệnh trong console **./đường-dẫn/tên-tập-tin** hoặc xác định biến môi trường PATH sử dụng thư mục chứa tập tin và gõ tên-tập-tin trong cửa sổ console. Nếu bạn đang làm việc tại thư mục chứa tập tin, bạn có thể chạy bằng lệnh:

```
./tên-tập-tin
```

Ví dụ: Cấp quyền và thực thi chương trình hello :

```
$cd /home/hv/baitap
```

```
$chmod +x hello
```

```
$./ hello
```



Bạn muốn tập tin này có thể thực thi được từ bất cứ nơi đâu chỉ mà chỉ cần gõ hello thì bạn sẽ đặt lại biến môi trường PATH trong tập tin `.bash_profile` trong thư mục `home`: `PATH=$PATH:/home/hv/baitap`. Nếu bạn muốn tập tin này cho những người dùng khác sử dụng thì bạn chép nó vào thư mục `/usr/local/bin`. Bạn nên nhớ cấp quyền lại cho tập tin này nếu bạn không muốn nó bị xóa hay bị sửa chữa. Đoạn lệnh sau có ý nghĩa: Chép tập tin hello vào thư mục `/usr/local/bin` và chuyển quyền sở hữu tập tin cho root, cấp cho root toàn quyền trên tập tin này, những người khác chỉ có quyền đọc và thực thi.

```
$cp hello /usr/local/bin
```

```
$chown root /usr/local/bin/hello
```

```
$chgrp root /usr/local/bin/hello
```

```
$chmod u=rwx go=rx /usr/local/bin/hello
```

V. Cú pháp ngôn ngữ Shell

Ngôn ngữ Shell là dạng ngôn ngữ script, không có độ uyển chuyển hay phức tạp như các ngôn ngữ lập trình chuyên nghiệp C, Pascal hay Java... Chương trình Shell được soạn thảo dưới dạng văn bản (text) và không được biên dịch thành tập tin binary như các ngôn ngữ khác. Khi chạy chương trình shell, shell sẽ biên dịch và thực thi. Trong Linux chúng ta gặp rất nhiều các chương trình shell xử lý những công việc rất hữu hiệu. Là nhà quản trị bạn cần phải nắm vững cú pháp ngôn ngữ shell để không chỉ viết những đoạn chương trình mà ít ra cũng hiểu được các script có sẵn điều khiển hệ thống của mình. Các thành phần trong ngôn ngữ shell:

- Biến: kiểu chuỗi, tham số và biến môi trường.
- Điều kiện: kiểm tra luận lý.
- Các lệnh điều khiển: if, for, while, until, case.
- Hàm.
- Các lệnh nội trú của shell.

V.1. Ghi chú, định shell thực thi, thoát chương trình

Dòng chú thích sử dụng trong các source chương trình dùng để giải thích ý nghĩa các lệnh hoặc chức năng của một biến hay một đoạn chương trình. Những dòng này không được biên dịch đối với các ngôn ngữ lập trình, và nó không được thực thi đối với chương trình shell. Bắt đầu một dòng chú thích là dấu `#`.

Ví dụ: một đoạn chương trình sử dụng dòng ghi chú.

```
# Kiểm tra có tồn tại tham số đầu tiên
if test $1 -z ; then
    echo "Không có tham số"
fi # kết thúc if
```



Trường hợp đặc biệt sau dấu # là dấu chỉ thị ! (#!) dùng để giải thích đây chính là dòng lệnh gọi shell để thông dịch các lệnh trong tập tin này. Bạn thường thấy dòng đầu tiên trong các chương trình shell là #!/bin/bash. Điều này có nghĩa là bạn sẽ dùng shell bash để thông dịch lệnh. Shell chúng ta chạy có thể xem là shell phụ và chúng có thể thực thi các lệnh mà không làm biến đổi các biến môi trường của shell chính. Cú pháp chung của chỉ thị này là :

```
#!shell-thực-thi
```

Nếu chúng ta không khai báo thì shell mặc nhiên trong Linux là bash. Các hệ Unix khác thì shell mặc nhiên là sh. Chỉ thị #! Còn dùng để chạy các chương trình khác trước khi thực thi các lệnh tiếp theo.

V.2. Sử dụng biến

Biến dùng trong chương trình shell không cần phải khai báo trước như các ngôn ngữ C, Pascal, ... Nó sẽ tự động khai báo khi người dùng sử dụng lần đầu tiên. Biến chỉ có thể lưu trữ dữ liệu dưới dạng chuỗi dù nó có thể chứa số. Trong trường hợp muốn sử dụng giá trị biến như là số thì phải có các phép biến đổi mà bạn sẽ tìm hiểu trong phần sau. Một vấn đề mà bạn phải lưu ý là shell phân biệt chữ hoa và chữ thường. Ví dụ hai biến **tong** và **Tong** là khác nhau.

V.2.1 Phép gán giá trị cho biến

Để đặt giá trị mới cho biến chúng ta sử dụng phép gán.

Cú pháp:

```
Ten-bien=giatri
```

Ví dụ:

```
Ten=Hung
```

```
So=200
```

Giá trị được gán có thể là hằng, biến hoặc biểu thức.

Lưu ý: Là bạn không được dùng dấu khoảng trắng giữa tên-biến=giá-trị

Ví dụ: ten =Hung là không hợp lệ

V.2.2 Lấy giá trị của biến

Muốn lấy giá trị của biến chúng ta thêm dấu \$ vào phía trước tên biến:

```
$tên-biến
```

Ví dụ:

```
tp=HaNoi
```

```
echo $tp
```

\$tp sẽ mang giá trị "HaNoi."

V.2.3 Hiện thị giá trị của biến ra màn hình

Lệnh echo dùng để hiển thị biến ra màn hình. Ta có thể dùng một trong 3:

```
echo "Dòng hiển thị"
```



echo “dòng hiển thị”

echo ‘dòng hiển thị’

Những kí tự nằm trong dấu ‘ ‘ được xem như là hàng chuỗi. Tất cả các kí tự sẽ hiển thị hết ra màn hình, kể cả các kí tự đặc biệt.

Ví dụ:

echo ‘ Gia tri cua bien la \$bien ‘

Kết quả hiện thị : Gia tri cua bien la \$bien

Khác với ý nghĩa của dấu ‘, dấu “ ” dùng để xác định chuỗi bao gồm cả các ký tự hiển thị và các giá trị biến. Muốn hiển thị các ký tự đặc biệt chúng ta phải thêm dấu \ vào trước ví dụ:

echo ten=Dung

echo “Su dung dau nhay kep”

echo “Gia tri bien la \$ten ”

echo “Ky hieu tien la \\$”

Kết quả hiện thị :

Su dung dau nhay kep

Gia tri bien la Dung

Ky hieu tien la \$

V.2.4 Nhập giá trị cho biến từ bàn phím

Cú pháp: **read** <tên-biến>

Gặp lệnh này chương trình sẽ đợi người dùng nhập giá trị vào, khi dữ liệu đã xong thì ấn Enter. Giá trị sẽ được gán vào biến tên-biến.

Ví dụ:

echo “Nhap vao ten cua ban “

read ten

echo “Ten vua nhap la \$ten”

Trong ví dụ trên khi xuất hiện dòng thông báo “Nhap vao ten cua ban “, người dùng nhập vào tên “ Nguyen Hung Dung” thì kết quả hiển thị là “Ten vua nhap la Nguyen Hung Dung “

V.2.5 Biến môi trường

Biến môi trường là biến được định nghĩa trước và mang giá trị mặc định khi shell khởi động. Nó giúp các chương trình cũng như hệ thống trong việc xử lý các công việc. Tên của biến môi trường thường là chữ hoa để phân biệt với các tên biến do người dùng đặt trong chương trình. Một số biến môi trường thông dụng:

Biến môi trường	Ý nghĩa



HOME	Chứa thư mục home của người dùng, là thư mục sử dụng sau khi đăng nhập hệ thống
PATH	Danh sách các thư mục tìm kiếm khi thực hiện các lệnh
PS1	Dấu nhắc hiển thị lệnh, dấu # đối với người dùng root, dấu \$ đối với người dùng thường.
PS2	Dấu nhắc thứ cấp thường là >
IFS	Dấu phân cách các trường trong danh sách chuỗi. Thường sử dụng dấu khoảng trắng, tab và xuống hàng
PPID	Số ID của tiến trình cha trong SHELL
RANDOM	Số ngẫu nhiên
SECONDS	Thời gian làm việc tính theo giây

V.2.6 Biến tham số

Khi gọi các lệnh chúng ta thường thêm vào sau lệnh các tham số, các tham số đó sẽ là giá trị của các biến tham số của chương trình.

Ví dụ cp sourc.txt dest.txt

Trong ví dụ sourc.txt và dest.txt là hai tham số của chương trình cp. Thao tác với các biến tham số từ trong chương trình chúng ta sử dụng các ký hiệu sau

Ký hiệu biến	Ý nghĩa
\$1, \$2, \$3	Giá trị các biến tham số thứ nhất, thứ 2.. tương ứng với các tham số từ trái sang phải trong dòng tham số.
\$0	Tên tập tin lệnh gọi
\$*	Danh sách tham số đầy đủ
\$#	Tổng số tham số.
\$\$	Số tiến trình mà chương trình đang hoạt động

V.3. Lệnh kiểm tra

Lệnh test hoặc dấu [] dùng để kiểm tra giá trị đúng sai của biểu thức. Lệnh test cho phép kiểm tra 3 kiểu dưới đây.



- Kiểm tra chuỗi:

Phép so sánh	Kết quả
Chuoi1 = chuoi2	Đúng (true) nếu 2 chuỗi bằng nhau
Chuoi1 != chuoi2	Đúng nếu 2 chuỗi khác nhau
-n chuoi	Đúng nếu chuỗi “chuoi” không rỗng
-z chuoi	Đúng nếu chuỗi “chuoi” rỗng

- So sánh toán học:

Phép so sánh	Kết quả
bieuthuc1 –eq biethuc2	Đúng nếu bieuthuc1 bằng biethuc2
bieuthuc1 –ne biethuc2	bieuthuc1 không bằng biethuc2
bieuthuc1 –gt biethuc2	bieuthuc1 lớn hơn biethuc2
bieuthuc1 –ge biethuc2	bieuthuc1 lớn hơn hoặc bằng biethuc2
bieuthuc1 –lt biethuc2	bieuthuc1 nhỏ hơn biethuc2
bieuthuc1 –le biethuc2	bieuthuc1 nhỏ hơn hoặc bằng biethuc2

- Kiểm tra tập tin

Phép kiểm tra	Kết quả
-d file	Đúng nếu tập tin là thư mục
-e file	tồn tại trên đĩa
-f file	là tập tin thông thường
-g file	có xác lập set-group-id trên file
-s file	có kích thước >0
-u file	có xác lập set-user-id
-r file	cho phép đọc
-w file	có phép ghi
-x file	cho phép thực thi

V.4. Biểu thức tính toán expr

Biểu thức **expr** được sử dụng cho việc tính toán. Các giá trị trong biểu thức được hiểu là số nguyên thay vì là chuỗi. Nó cũng dùng để đổi chuỗi thành số. Biểu thức expr được bao bọc bởi 2 dấu ` (Không phải dấu nháy đơn, là dấu ở phím bên trái phím số 1-!). Trong biểu thức tính toán các toán tử và toán hạng cách nhau bằng khoảng trắng. Các phép toán và phép so sánh expr cho phép:

	hoặc	=	bằng nhau
&	và	+	cộng
>	lớn hơn	-	trừ



<	nhỏ hơn	*	nhân
>=	lớn hơn hoặc bằng	/	chia
<=	nhỏ hơn hoặc bằng	%	chia lấy phần dư
!=	khác nhau		

V.5. Kết nối lệnh, khối lệnh và lấy giá trị của lệnh

Shell cho phép sử dụng phép hoặc (OR) và phép và (AND) để kết nối các lệnh.

V.5.1 Phép và (AND)

Cú pháp của phép toán logic AND:

```
lệnh_1 && lệnh_2 && lệnh_3 ...
```

Các lệnh thực hiện từ trái sang phải cho đến khi một lệnh có kết quả lỗi. Kết quả cuối cùng của dãy lệnh này là đúng (true) nếu tất cả các lệnh đều đúng, ngược lại là sai.

V.5.2 Phép hoặc (OR)

Cú pháp của phép toán logic OR:

```
lệnh_1 || lệnh_2 || lệnh_3 ...
```

Các lệnh thực hiện từ trái sang phải cho đến khi một lệnh có kết quả đúng. Kết quả cuối cùng của dãy lệnh này là đúng (true) nếu có ít nhất một lệnh là đúng, ngược lại là sai.

V.5.3 Khối lệnh

Khi chúng ta cần thực thi nhiều lệnh liên tiếp nhau, có thể dùng khối lệnh. Khối lệnh nằm giữa 2 dấu { }

V.5.4 Lấy giá trị của một lệnh

Khi viết chương trình nhiều khi chúng ta lấy kết quả của lệnh này làm đối số hay giá trị xử lý của lệnh kia. Ta có thể làm được điều này bằng cách sử dụng cú pháp \$(command). Khi dùng \$(command), kết quả của việc thực hiện lệnh command được trả về.

V.6. Cấu trúc rẽ nhánh if

Cú pháp của cấu trúc rẽ nhánh if:

```
if <btdk > ; then
    lệnh1
else
    lệnh2
fi
```




Nếu biểu thức điều kiện btdk là đúng thì các lệnh trong lenh1 sẽ thực hiện, ngược lại (btdk không đúng) thì các lệnh trong lenh2 sẽ được thực hiện với điều kiện mệnh đề else tồn tại. Trong lenh1, lenh2 có thể một hoặc nhiều lệnh.

Ví dụ: Nhập vào điểm của môn học, cho biết kết quả.

```
echo "chuong trinh ket qua mon hoc"
echo "Nhap vao diem"
read diem
if [ $diem -ge 5 ] ; then
    echo "Dat"
else
    echo "Hong"
fi
```

Cú pháp của if còn cho phép bạn sử dụng nhiều mệnh đề so sánh liên tiếp qua từ khóa elif như sau:

```
if <btdk1> ; then
    lenh1
elif <btdk2> ; then
    lenh2
...
elif <btdkn> ; then
    lenh n
else
    lenh_n+1
fi
```

Ví dụ: Nhập vào điểm cho biết xếp loại :

```
echo "Xep loai"
echo "Nhap vao diem"
read diem
if test $diem -ge 8 ; then
    echo "Loai Gioi"
elif test $diem -ge 7 ; then
    echo "Loai Kha"
elif test $diem -ge 5 ; then
    echo "Loai TB"
```



```
else
    echo "Loai Yeu"
fi
```

V.7. Cấu trúc lựa chọn Case

Dùng case khi chúng ta sử dụng giá trị của một biểu thức để rẽ các nhánh khác nhau. Cú pháp của cấu trúc lựa chọn như sau::

```
case <biên-bt> in
    giatri1 [ |giatri12 ... ] ) lenh-th1 ;;
    giatri21 [ |giatri22 ... ] ) lenh-th3 ;;
    giatri31 [ |giatri32 ... ] ) lenh-th3 ;;
    ...
    giatrin1 [ |giatrinn2 ... ] ) lenh-thn ;;
    *          ) lenh-thnn ;;
esac
```

Lệnh case sẽ kiểm tra biên-bt với các dạng hay giá trị bên dưới, nếu đúng thì thực hiện các lệnh trong mệnh đề đó.

Ví dụ: ta sẽ tạo menu lựa chọn và cho phép người dùng chọn chức năng thực hiện. Nếu biến chọn là 1 thì liệt kê thư mục hiện hành, 2 thì cho biết đường dẫn thư mục hiện hành, các số khác là không hợp lệ.

```
clear
echo
echo " Menu "
echo " 1. Liet ke thu muc hien hanh"
echo " 2. Cho biet duong dan thu muc hien hanh"
read chon
case $chon in
    1) ls -l ;;
    2) pwd ;;
    *) echo "Khong hop le" ;;
esac
```

V.8. Cấu trúc lặp

V.8.1 Vòng lặp For

Vòng lặp for sử dụng trong trường hợp xác định trước số lần lặp. Cú pháp của vòng lặp for:



```
for <variable> in giá-trị-1 giá-trị-2 giá-trị-3 ...
```

```
do
```

```
    các-lệnh ;
```

```
done
```

Chương trình có số lần sẽ lặp bằng số giá trị phía sau từ khoá in, trong quá trình lặp biến variable mang lần lượt các giá trị phía sau in

Ví dụ:

```
for gt in apple banana 34
```

```
do
```

```
    echo $gt
```

```
done
```

Kết quả sau khi thực hiện là :

```
apple
```

```
banana
```

```
34
```

V.8.2 Vòng lặp While

Lệnh while sử dụng khi số lần lặp không xác định trước. Cú pháp của vòng lặp while:

```
while <điều-kiện>
```

```
do
```

```
    các-lệnh;
```

```
done
```

Vòng lặp được thực hiện khi điều-kiện còn đúng.

Ví dụ:

```
echo "An phim Y/y de tiep tuc"
```

```
while [ $chon = 'y' || $chon = 'Y' ]
```

```
do
```

```
    echo "chao ban"
```

```
    read chon
```

```
done
```

V.8.3 Vòng lặp Until

Sử dụng tương tự như while nhưng điều kiện lặp ngược lại, until sẽ được lặp ít nhất một lần, điều kiện đúng sẽ thoát ra khỏi vòng lặp.

Cú pháp :



```
until <điều-kiện>
do
    Lệnh 1;
    Lệnh 2;
    ...
    Lệnh n
done
```

Ví dụ: Chương trình sẽ lặp cho đến khi $n \leq 10$

```
echo Nhập vào số n
read n
until [ $n -lt 10 ]
do
    echo "n lớn hơn 10"
    n=`expr $n -1`
done
```

V.9. Lệnh break, continue, exit

Lệnh **break** cho phép bạn thoát ra khỏi vòng lặp mà không cần kiểm tra điều kiện lặp. Lệnh **exit** thì làm chương trình thoát ra và trở về dấu nhắc lệnh \$

Ví dụ:

Nhập số n từ đối số dòng lệnh, tính tổng $S = 1+2+ \dots +n$

```
echo "chương trình tính tổng"
if [ -z $1 ]
echo " tổng <n> "
exit 0
```

```
fi
```

```
s=0
```

```
i=1
```

```
while true
```

```
do
```

```
    s=`expr $i + $s`
```

```
    i=`expr $i + 1`
```

```
    if [ i -gt n ]; then
```

```
        break;
```



fi

done

echo \$s

Lệnh continue dùng để quay lại vòng lặp kế mà không cần thực hiện các lệnh còn lại.

V.10. Các lệnh khác

Lệnh . dùng thực thi một script trong thư mục hiện hành và giữ nguyên các thay đổi môi trường mà chương trình đã tác động sau khi thoát khỏi chương trình. Cách sử dụng:

./tên-script

Lệnh exec. Dùng thực thi một chương trình như chạy từ dòng lệnh, sử dụng shell phụ khác.

Ví dụ:

exec mc

Lệnh export : dùng chuyển giá trị biến sang các shell khác sử dụng.

V.11. Hàm(function)

Cũng như các ngôn ngữ lập trình khác, shell cho phép bạn sử dụng hàm. Hàm là một đoạn chương trình con nằm trong script chính. Nó có thể được gọi lại nhiều lần trong script chính. Cú pháp định nghĩa hàm:

```
tên-hàm() {
    các-lệnh-của-hàm.
}
```

Ví dụ:

```
chao()
{
    echo "hello"
}
```

V.11.1 Gọi hàm và truyền tham số cho hàm

Để gọi hàm thực hiện ta sử dụng tên hàm hoặc có thêm tham số đi kèm:

```
tên-hàm
tên-hàm thamso-1 thamso-2 ...
```

V.11.2 Lấy giá trị của hàm

Để lấy giá trị của hàm trong shell ta thực hiện theo cú pháp sau:

```
$( tên_hàm )
```

Ví dụ:



```
Conn_value=$( netstat -an|grep :80|wc -l )
```



BÀI 12

Quản Lý Tiến Trình

Tóm tắt

Lý thuyết: 5 tiết - Thực hành: 5 tiết.

Mục tiêu	Các mục chính	Bài tập bắt buộc	Bài tập làm thêm
Giới thiệu cơ chế quản lý và điều phối tiến trình, thiết lập lịch biểu hoạt động cho các chương trình trong hệ thống.	<ol style="list-style-type: none">I. Định nghĩaII. Xem thông tin tiến trìnhIII. Tiến trình tiền cảnh (foreground process)IV. Tiến trình hậu cảnh (background process)V. Tạm dừng và đánh thức tiến trình.VI. Hủy một tiến trìnhVII. Chương trình lập lịch atVIII. Chương trình lập lịch batchIX. Chương trình lập lịch crontab	Bài tập 1.1 (quản lý tiến trình)	



I. Định nghĩa

Bạn có thể kích hoạt một chương trình bằng tên của chương trình ấy, hoặc từ các tập tin có chứa lệnh shell. Trong khi thực hiện, chương trình có thể tương tác với nhiều thành phần khác của hệ thống. Chương trình có thể đọc và ghi vào tập tin, quản lý thông tin trong RAM, hoặc gửi thông tin đến máy in, modem hay những thiết bị khác.

Tiến trình là một chương trình đơn chạy trên không gian địa chỉ ảo của nó, ở một khía cạnh nào đó, tiến trình hơn chương trình ở chỗ là biết sử dụng tài nguyên của một hệ thống đang chạy, trong khi chương trình chỉ đơn thuần là một loạt các câu lệnh. Một chương trình hay lệnh có thể phát sinh ra nhiều tiến trình khác. Khảo sát lệnh **nroff –man ps.1 | grep kill | more** sẽ sinh ra 3 tiến trình khác nhau. Có 3 loại tiến trình chính trên Linux:

- Tiến trình tương tác (Interactive processes) : là tiến trình khởi động và quản lý bởi shell, kể cả tiến trình foreground hoặc background.
- Tiến trình thực hiện theo lô (Batch processes) : tiến trình không gắn liền đến bàn điều khiển (terminal) và được nằm trong hàng đợi để lần lượt thực hiện.
- Tiến trình ẩn trên bộ nhớ (Daemon processes) : là các tiến trình chạy ẩn bên dưới hệ thống (background). Các tiến trình này thường được khởi tạo - một cách tự động - sau khi hệ thống khởi động. Đa số các chương trình server cho các dịch vụ chạy theo phương thức này. Đây là các chương trình sau khi được gọi lên bộ nhớ, đợi (thụ động) các yêu cầu từ các chương trình khách (client) để trả lời sau các cổng xác định (cổng là khái niệm gắn liền với giao thức TCP/IP BSD socket). Hầu hết các dịch vụ trên Internet như Mail, Web, Domain Name Service ... đều được thi hành theo nguyên tắc này. Các chương trình loại này được gọi là các chương trình daemon và tên của nó thường kết thúc bằng ký tự "d" như named, inerd ...

Một tiến trình khi thực hiện nếu sinh ra nhiều tiến trình con được gọi là tiến trình cha (Parent Process). Khi tiến trình cha bị dừng thì các tiến trình con của nó cũng bị dừng theo.

Mỗi tiến trình mang một định danh gọi là PID (Process IDentification). Process Id là một con số lớn hơn 0 và là duy nhất. Hệ thống dựa vào các PID này để quản lý các tiến trình. Khi khởi động, Linux sẽ thực hiện một tiến trình sẵn có trong hệ thống mang tên Init (Vì là tiến trình đầu tiên được thực hiện nên PID=1). Sau đó tiến trình này mới sinh ra các tiến trình khác; các tiến trình khác có thể sinh ra các tiến trình khác nữa và cứ tiếp tục như thế tạo thành cây phân cấp các tiến trình (xem hình cây tiến trình bên dưới). Như vậy, dừng tiến trình Init nghĩa là dừng toàn bộ hệ thống.

Ví dụ: Xem tiến trình trong hệ thống.

```
$pstree -n -p
```

```
init(1)-+-keventd(2)
          |-kadm-idled(3)
          |-mdrecoveryd(9)
          |-syslogd(629)
          |-klogd(634)
          |-rpc.statd(683)
```




|apmd(795)

|sshd(851)---sshd(1064)---bash(1065)---pstree(1492)

|xinetd(884)

|sendmail(924)

|crond(961)

Số trong dấu () là chỉ số PID của tiến trình.

II. Xem thông tin tiến trình

Cách đơn giản nhất để kiểm tra những tiến trình đang chạy trong hệ thống là sử dụng lệnh ps (process status). Lệnh ps có nhiều tùy chọn và phụ thuộc một cách mặc định vào người đăng nhập vào hệ thống. Cú pháp lệnh #ps <option>

Một số tùy chọn của lệnh ps cần tham khảo:

Tên lệnh và tùy chọn	Mục đích
ps -ux	Xem tất cả các tiến trình mà user kích hoạt
ps -T	Xem những tiến trình được chạy tại terminal hiện tại của user.
ps -aux	Xem tất cả các tiến trình trong hệ thống
ps -u username	Xem tất cả các tiến trình của user nào đó (được chỉ định thông qua tham số username)

Ví dụ: Lệnh ps kết quả hiển thị như sau:

```
PID TTY STAT TIME COMMAND
41   v01  S    0:00  -bash
134  v01  R    0:00  ps
```

Để hiển thị tất cả các tiến trình, ta có thể sử dụng lệnh ps -a. Bất cứ người dùng nào trong hệ thống đều có thể thấy tất cả các tiến trình, nhưng chỉ có thể điều khiển được các tiến trình do mình tạo ra. Tuy nhiên, đối với super-user thì có quyền điều khiển tất cả các tiến trình trong hệ thống. Lệnh ps -ax cho phép hiển thị tất cả các tiến trình, kể cả những tiến trình không gắn với thiết bị đầu cuối (tty). Chúng ta có thể coi các tiến trình đang thực hiện cùng với đầy đủ dòng lệnh đã khởi tạo nó bằng lệnh ps -axl. Lệnh man ps cho phép coi các tham số tự chọn khác của lệnh ps.



III. Tiến trình tiền cảnh(foreground process)

Khi thực hiện một chương trình từ dấu nhắc shell (\$ hoặc #), chương trình sẽ thực hiện và không xuất hiện dấu nhắc cho đến khi thực hiện xong chương trình. Do đó, chúng ta không thể thực hiện các công việc khác trong khi chương trình này đang thực hiện. Chương trình hoạt động như vậy gọi là chương trình tiền cảnh. Chúng ta thử chạy 1 chương trình có thời gian thực hiện lâu để kiểm tra, ví dụ liệt kê tất cả các thư mục của hệ thống bằng lệnh **find / -name pro -print**. Thực hiện lệnh `find / -name pro -print > results.txt`. Vì kết quả rất lớn nên chúng ta có thể cho vào tập tin : **find / -name pro -print > results.txt**. Khi chương trình chạy bạn phải chờ rất lâu cho đến khi dấu nhắc xuất hiện trở lại.

IV. Tiến trình hậu cảnh(background process)

Tiến trình hậu cảnh là tiến trình sinh ra độc lập với tiến trình cha. Khi chạy một chương trình chiếm thời gian lâu chúng ta có thể cho phép chúng chạy ngầm định bên dưới và tiếp tục thực hiện các công việc khác. Để tiến trình chạy dưới chế độ hậu cảnh chúng ta thêm dấu & vào sau lệnh thực hiện chương trình

Ví dụ: `$ find / -name pro -print > results.txt &`

[1] 2489

Khi chạy chương trình hệ thống sẽ xuất hiện dấu \$ ngay lập tức, chương trình này thực đang thực hiện với mã số tiến trình là 2489 và đặt ở hậu cảnh [1], chúng ta có thể kiểm tra chương trình này có hoạt động không bằng lệnh: **ps -aux | grep find**. Đơn giản hơn chúng ta dùng lệnh `jobs` để xem các tiến trình đang có ở hậu cảnh:

`$jobs`

[1] + Running find / -name pro -print > results.txt &

Dòng trên cho biết có 1 tiến trình đang chạy ở hậu cảnh. Khi thực hiện xong chương trình thì màn hình xuất hiện câu thông báo:

[1] Done find / -name pro -print.

Việc sử dụng các tiến trình chạy hậu cảnh giúp cho chúng đưa vào hoạt động nhiều tiến trình đồng thời nó thích hợp với chương trình hoạt động liên tục như daemon.

V. Tạm dừng và đánh thức tiến trình

Trong một số trường hợp khi đang chạy chương trình nhưng thời gian thực hiện quá lâu và muốn đưa nó vào hậu cảnh. Linux cho phép chúng ta đưa nó tạm dừng và vào hậu cảnh bằng phím Ctrl-Z. Khi tiến trình đang chạy nhận được tín hiệu Ctrl-Z thì nó tạm dừng và chuyển vào hậu cảnh, trả dấu nhắc lệnh lại cho người dùng. Chúng ta có thể xem tiến trình có trong hậu cảnh:

`$ jobs`

[1] + Stopped find / -name pro -print > results.txt

Dòng kết quả của `jobs` cho thấy tiến trình này đã có trong hậu cảnh nhưng không được thực hiện vì chúng ta đã tạm dừng trước đó. Để cho tiến trình đang dừng tại hậu cảnh hoạt động trở lại ta dùng lệnh `bg`. Lệnh này yêu cầu tham số là số thứ tự của tiến trình ở hậu cảnh. Với ví dụ trên ta cho chương trình hoạt động bằng lệnh: **\$bg 1**



```
find / -name pro -print > results.txt&
```

```
$ jobs
```

```
[1] + Running find / -name pro -print > results.txt &
```

Khi đưa vào chạy tại hậu cảnh chúng ta thấy dòng lệnh thực hiện được thêm dấu & vào cuối. Ngược lại khi muốn một tiến trình đang chạy ở hậu cảnh chuyển sang chạy tiền cảnh chúng ta dùng lệnh: **fg <số-tt-tiến-trình>**.

```
$ fg 1
```

```
find / -name pro -print > results.txt
```

VI. Hủy một tiến trình

Trong nhiều trường hợp, một tiến trình có thể bị treo, chẳng hạn như: Một bàn phím điều khiển không trả lời các lệnh từ bàn phím, một chương trình server cần nhận cấu hình mới, card mạng cần thay đổi địa chỉ IP ..., khi đó chúng ta phải dừng (kill) tiến trình đang có vấn đề. Linux có lệnh kill để thực hiện công việc này. Trước tiên, bạn cần phải biết PID của tiến trình cần dừng thông qua lệnh ps. Sau đó, ta sử dụng lệnh:

```
#kill -9 PID-của-tiến-trình
```

Tham số -9 là tín hiệu dừng tiến trình không điều kiện. Không nên dừng các tiến trình mà mình không biết vì có thể làm treo máy hoặc những dịch vụ khác. Một tiến trình có thể sinh ra các tiến trình con trong quá trình hoạt động của mình. Nếu tiến trình cha bị dừng, các tiến trình con cũng sẽ dừng theo, nhưng không tức thì. Vì vậy, phải đợi một khoảng thời gian và sau đó kiểm tra lại xem tất cả các tiến trình con có dừng đúng hay không. Trong một số trường hợp hạn hữu, tiến trình có lỗi nặng không dừng được, biện pháp cuối cùng là khởi động lại máy.

Lưu ý: Chỉ có người dùng root mới có quyền dừng tất cả các tiến trình, còn những người dùng khác chỉ được dừng các tiến trình do mình tạo ra.

VII. Chương trình lập lịch at

Linux có các lệnh cho phép thực hiện các tiến trình ở thời điểm định trước thông qua lệnh at. Thời điểm thực hiện tiến trình được nhập vào thông qua tham số của lệnh at. Cú pháp của lệnh at như sau:

```
$ at [time]
```

```
<các lệnh thực hiện>
```

```
...
```

```
<Ctrl+D>
```

Sau khi bạn kết thúc lệnh at, dòng thông báo giống như sau sẽ hiện ra màn hình: **job 756001.a at Sat Dec 21 01:23:00 2000**. Trong đó số 756001.a là số nhận dạng công việc (job number) cho phép tham chiếu tới lịch thực hiện đó. Sau khi lập lịch, nếu muốn hủy bỏ, ta có thể sử dụng lệnh.

```
at -r [job-number]
```



Lệnh này có thể khác với các phiên bản khác nhau. Ví dụ như đối với RedHat 6.2, lệnh xóa một job là `atrm job_number`. Trong mọi trường hợp, xem manpage để biết các lệnh và tham số cụ thể. Bạn có thể dùng quy tắc chuyển hướng (redirect) để lập trình cho nhiều lệnh cùng một lúc

```
at 10:59 < tập_lệnh
```

Trong đó, `tập_lệnh` là một tập tin dạng text có các lệnh. Để kiểm tra các tiến trình mà bạn đã nhập vào, dùng lệnh `at -l`

VIII. Chương trình lập lịch batch

Khác với lệnh `at` là tiến trình được thực hiện vào các thời điểm do người sử dụng qui định, lệnh `batch` cho phép hệ thống tự quyết định khi nào tiến trình được thực hiện dựa trên mức tải của hệ thống. Thông thường, các tiến trình `batch` được thi hành khi mức tải của hệ thống dưới 20%. Những chương trình như in ấn, cập nhật dữ liệu lớn rất thích hợp với kiểu lệnh này. Cú pháp của lệnh `batch` như sau :

```
$ batch<Return>
```

```
lp/usr/sales/reports/*<Return>
```

```
<^D>
```

IX. Chương trình lập lịch crontab

Các lệnh `at` và `batch` cho phép lập kế hoạch thực hiện tiến trình một lần. Linux còn cho phép lập kế hoạch có tính chất chu kỳ thông qua lệnh `cron` (viết tắt của *chronograph*) và các tập tin `crontabs`. Chương trình daemon `cron` (`crond`) được kích hoạt ngay từ đầu với khởi động của hệ thống. Khi khởi động, `cron` xem có các tiến trình trong hàng đợi nhập vào bởi lệnh `at`, sau đó xem xét các tập tin `crontabs` xem có các tiến trình cần phải thực hiện hay không rồi “đi ngủ”. `Cron` sẽ “thức dậy” mỗi phút để kiểm tra xem có phải thực hiện tiến trình nào không. Mọi người dùng trong hệ thống đều có thể lập lịch các tiến trình sẽ được thực hiện bởi `cron`. Để làm điều này, bạn cần tạo một tập tin văn bản theo cú pháp của `cron` như sau:

Phút	giờ	ngày_trong_tháng	tháng_trong_năm	ngày_trong_tuần	lệnh
0	8	*	*	1	/u/ sartin/bin/status_report

Cho phép `/u/sartin/bin/status_report` được thực hiện vào 8 giờ 00 phút các thứ hai. Mỗi dòng chứa thời gian và lệnh. Lệnh sẽ được `cron` thực hiện tại thời điểm ghi ở trước trên cùng dòng đó. Năm cột đầu liên quan tới thời gian có thể thay thế bằng dấu “*” có nghĩa là “với mọi”. Các giá trị có thể cho các trường là:



- + Phút (0 – 59)
- + Giờ (0 – 23)
- + Ngày trong tháng (1 – 31)
- + Tháng trong năm (1-12)
- + Ngày trong tuần (0 – 6, 0 is Sunday)
- + Lệnh (rest of line)

Sau đó dùng lệnh `crontab` để cài đặt tập tin lệnh vào thư mục `/usr/spool/cron/crontabs`. Mỗi người dùng sẽ có một tập tin `crontab` trùng tên mình (user name) để lưu tất cả các lệnh cần thực hiện theo chu kỳ trong thư mục này. Cú pháp sử dụng **crontab**:

`Crontab <tên_tập_tin_ệnh>`



BÀI 13

Domain Name System

Tóm tắt

Lý thuyết: 5 tiết - Thực hành: 10 tiết.

Mục tiêu	Các mục chính	Bài tập bắt buộc	Bài tập làm thêm
Giới thiệu cơ chế tổ chức và quản lý dịch vụ DNS trên môi trường Linux	I. Giới thiệu về DNS. II. Cách phân bổ dữ liệu quản lý domain name III. Cơ chế phân giải tên IV. Sự khác nhau giữa domain name và zone V. Fully Qualified Domain Name (FQDN) VI. Phân loại Domain Name Server VII. Sự ủy quyền(Delegating Subdomains) VIII. Resource Record (RR) IX. Hoạt động của Name Server trong Linux X. Cài đặt BIND XI. Kiểm tra hoạt động của DNS XII. Cấu hình Secondary Name Server XIII. Một số quy ước XIV. Cấu hình sự ủy quyền cho các miền con	Bài tập 02.1 (Dịch vụ DNS)	Bài tập 02.2 (Dịch vụ DNS)



I. Giới thiệu về DNS

Mỗi máy tính trong mạng muốn liên lạc hay trao đổi thông tin, dữ liệu cho nhau cần phải biết rõ địa chỉ IP của nhau. Nếu số lượng máy tính nhiều thì việc nhớ những địa chỉ IP này rất là khó khăn. Mỗi máy tính ngoài địa chỉ IP ra còn có một cái tên (computer name). Đối với con người việc nhớ những cái tên này dù sao cũng dễ dàng hơn vì chúng có tính trực quan và gợi nhớ hơn địa chỉ IP. Vì thế, người ta nghĩ ra cách làm sao ánh xạ địa chỉ IP thành tên máy tính.

Ban đầu do quy mô mạng ARPAnet (tiền thân của mạng Internet) còn nhỏ chỉ vài trăm máy, nên chỉ có một tập tin đơn HOSTS.TXT lưu thông tin về ánh xạ tên máy thành địa chỉ IP. Trong đó tên máy chỉ là 1 chuỗi văn bản không phân cấp (flat name). Tập tin này được duy trì tại 1 máy chủ và các máy chủ khác lưu giữ bản sao của nó. Tuy nhiên khi quy mô mạng lớn hơn, việc sử dụng tập tin HOSTS.TXT có các nhược điểm như sau:

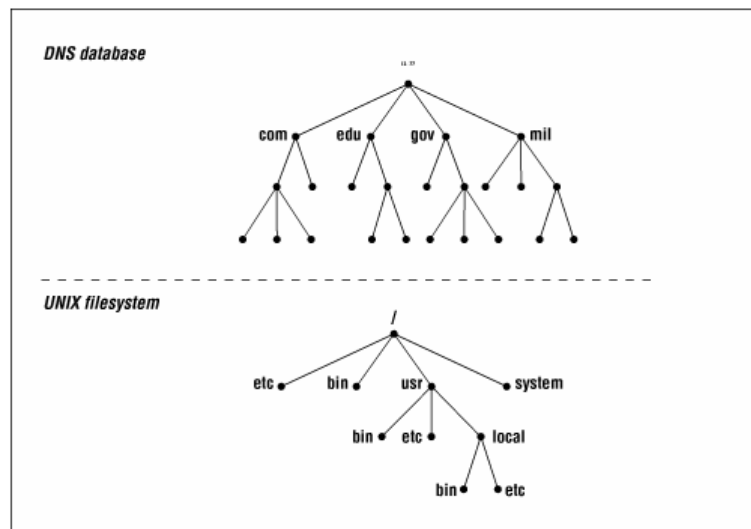
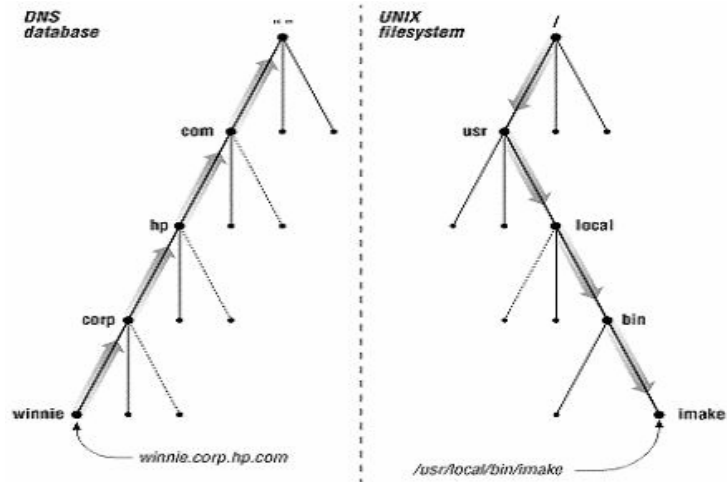
- Lưu lượng mạng và máy chủ duy trì tập tin HOSTS.TXT bị quá tải do hiệu ứng “cổ chai”.
- Xung đột tên: Không thể có 2 máy tính có cùng tên trong tập tin HOSTS.TXT. Tuy nhiên do tên máy không phân cấp và không có gì đảm bảo để ngăn chặn việc tạo 2 tên trùng nhau vì không có cơ chế uỷ quyền quản lý tập tin nên có nguy cơ bị xung đột tên.
- Không đảm bảo sự toàn vẹn: việc duy trì 1 tập tin trên mạng lớn rất khó khăn. Ví dụ như khi tập tin HOSTS.TXT vừa cập nhật chưa kịp chuyển đến máy chủ ở xa thì đã có sự thay đổi địa chỉ trên mạng rồi.

Tóm lại việc dùng tập tin HOSTS.TXT không phù hợp cho mạng lớn vì thiếu cơ chế phân tán và mở rộng. Do đó, dịch vụ DNS ra đời nhằm khắc phục các nhược điểm này. Người thiết kế cấu trúc của dịch vụ DNS là Paul Mockapetris - USC's Information Sciences Institute, và các khuyến nghị RFC của DNS là RFC 882 và 883, sau đó là RFC 1034 và 1035 cùng với 1 số RFC bổ sung như bảo mật trên hệ thống DNS, cập nhật động các bản ghi DNS ...

Lưu ý: Hiện tại trên các máy chủ vẫn sử dụng được tập tin hosts.txt để phân giải tên máy tính thành địa chỉ IP (Trong Linux là /etc/hosts)

Dịch vụ DNS hoạt động theo mô hình Client - Server: phần Server gọi là máy chủ phục vụ tên nameserver, còn phần Client là trình phân giải tên resolver. Nameserver chứa các thông tin CSDL của DNS, còn resolver đơn giản chỉ là các hàm thư viện dùng để tạo các truy vấn (query) và gửi chúng qua đến name server. DNS được thi hành như một giao thức tầng Application trong mạng TCP/IP.

DNS là 1 CSDL phân tán. Điều này cho phép người quản trị cục bộ quản lý phần dữ liệu nội bộ thuộc phạm vi của họ, đồng thời dữ liệu này cũng dễ dàng truy cập được trên toàn bộ hệ thống mạng theo mô hình Client - Server. Hiệu suất sử dụng dịch vụ được tăng cường thông qua cơ chế nhân bản (replication) và lưu tạm (caching). Một hostname trong domain là sự kết hợp giữa những từ phân cách nhau bởi dấu chấm. Ví dụ hostname là server.t3h.com, trong đó server là hostname và t3h.com là domain name. Domain name phân bổ theo cơ chế phân cấp tương tự như sự phân cấp của hệ thống tập tin Unix/Linux.



Cơ sở dữ liệu(CSDL) của DNS là một cây đảo ngược. Mỗi nút trên cây cũng lại là gốc của 1 cây con. Mỗi cây con là 1 phân vùng con trong toàn bộ CSDL DNS gọi là 1 miền (domain). Mỗi domain có thể phân chia thành các phân vùng con nhỏ hơn gọi là các miền con (subdomain). Mỗi domain có 1 tên (domain name). Tên domain chỉ ra vị trí của nó trong CSDL DNS. Trong DNS tên miền là chuỗi tuân tự các tên nhãn tại nút đó đi ngược lên nút gốc của cây và phân cách nhau bởi dấu chấm. Tên nhãn bên phải trong mỗi domain name được gọi là top-level domain. Trong ví dụ trước server.t3h.com, vậy com là top-level domain. Bảng sau đây liệt kê top-level domain.

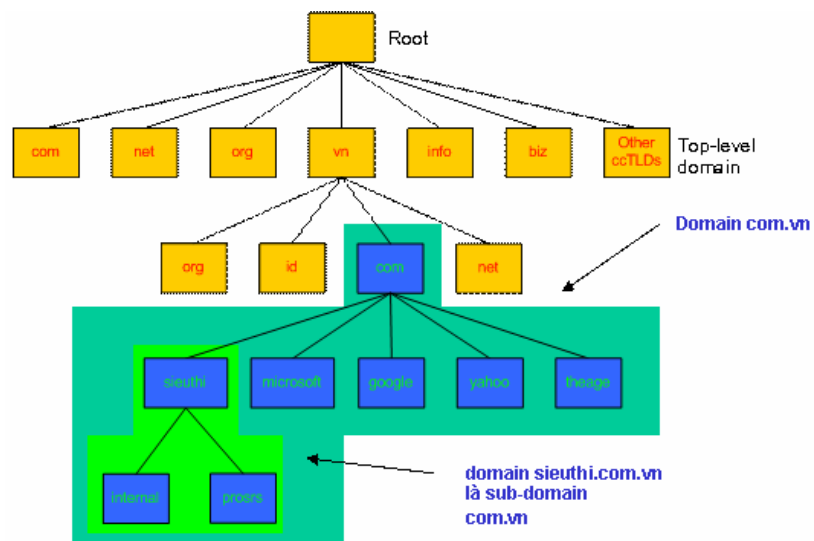
Tên miền	Mô tả
.com	Các tổ chức, công ty thương mại
.org	Các tổ chức phi lợi nhuận
.net	Các trung tâm hỗ trợ về mạng
.edu	Các tổ chức giáo dục

.gov	Các tổ chức thuộc chính phủ
.mil	Các tổ chức quân sự
.int	Các tổ chức được thành lập bởi các hiệp ước quốc tế

Vì sự quá tải của những domain name đã tồn tại, do đó đã làm phát sinh những top-level domain mới. Bảng sau đây liệt kê những top-level domain mới.

Tên miền	Mô tả
.arts	Những tổ chức liên quan đến nghệ thuật và kiến trúc
.nom	Những địa chỉ cá nhân và gia đình
.rec	Những tổ chức có tính chất giải trí, thể thao
.firm	Những tổ chức kinh doanh, thương mại.
.info	Những dịch vụ liên quan đến thông tin.

Bên cạnh đó, mỗi nước cũng có một top-level domain. Ví dụ top-level domain của Việt Nam là vn, Mỹ là us... Mỗi nước khác nhau có cơ chế tổ chức phân cấp domain khác nhau tùy thuộc vào mỗi nước. Ví dụ về tổ chức domain của Việt Nam:





II. Cách phân bổ dữ liệu quản lý domain name

Những root name server (.) quản lý những top-level domain trên Internet. Tên máy và địa chỉ IP của những name server này được công bố cho mọi người biết và chúng được liệt kê trong bảng sau. Những name server này cũng có thể đặt khắp nơi trên thế giới.

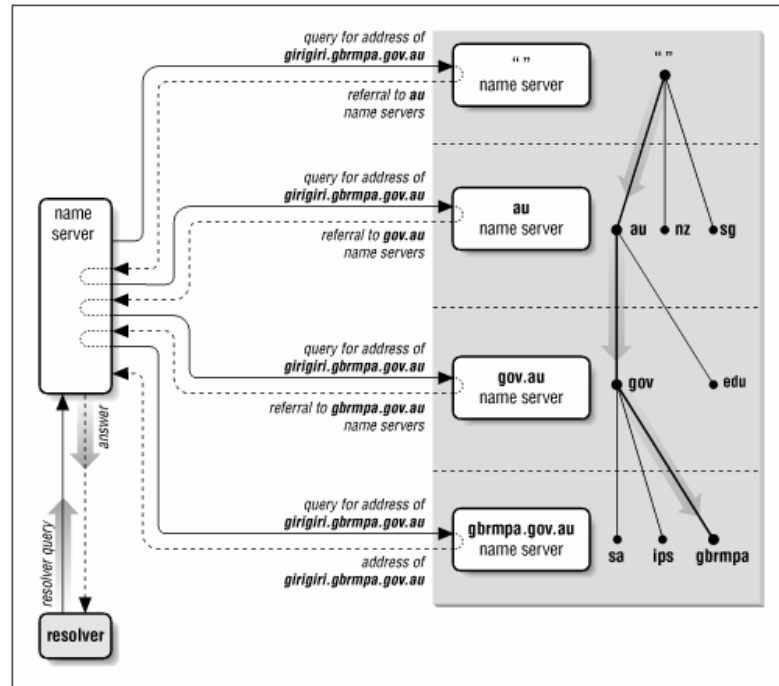
Tên máy tính	Địa chỉ IP
H.ROOT-SERVERS.NET	128.63.2.53
B.ROOT-SERVERS.NET	128.9.0.107
C.ROOT-SERVERS.NET	192.33.4.12
D.ROOT-SERVERS.NET	128.8.10.90
E.ROOT-SERVERS.NET	192.203.230.10
I.ROOT-SERVERS.NET	192.36.148.17
F.ROOT-SERVERS.NET	192.5.5.241
F.ROOT-SERVERS.NET	39.13.229.241
G.ROOT-SERVERS.NET	192.112.88.4
A.ROOT-SERVERS.NET	198.41.0.4

Thông thường một tổ chức được đăng ký một hay nhiều domain name. Sau đó, mỗi tổ chức sẽ cài đặt một hay nhiều name server và duy trì cơ sở dữ liệu cho tất cả những máy tính trong domain. Những name server của tổ chức được đăng ký trên Internet. Một trong những name server này được biết như là Primary Name Server. Nhiều Secondary Name Server được dùng để làm backup cho Primary Name Server. Trong trường hợp Primary bị lỗi, Secondary được sử dụng để phân giải tên. Primary Name Server có thể tạo ra những subdomain và ủy quyền những subdomain này cho những Name Server khác.

III. Cơ chế phân giải tên

III.1. Phân giải tên thành IP

Root name server : Là máy chủ quản lý các nameserver ở mức top-level domain. Khi có truy vấn về một tên miền nào đó thì Root Name Server phải cung cấp tên và địa chỉ IP của name server quản lý top-level domain (Thực tế là hầu hết các root server cũng chính là máy chủ quản lý top-level domain) và đến lượt các name server của top-level domain cung cấp danh sách các name server có quyền trên các second-level domain mà tên miền này thuộc vào. Cứ như thế đến khi nào tìm được máy quản lý tên miền cần truy vấn. Qua trên cho thấy vai trò rất quan trọng của root name server trong quá trình phân giải tên miền. Nếu mọi root name server trên mạng Internet không liên lạc được thì mọi yêu cầu phân giải đều không thực hiện được. Hình vẽ dưới mô tả quá trình phân giải grigiri.gbrmpa.gov.au trên mạng Internet.



Client sẽ gửi yêu cầu cần phân giải địa chỉ IP của máy tính có tên girigiri.gbrmpa.gov.au đến name server cục bộ. Khi nhận yêu cầu từ resolver, Nameserver cục bộ sẽ phân tích tên này và xét xem tên miền này có do mình quản lý hay không. Nếu như tên miền do server cục bộ quản lý, nó sẽ trả lời địa chỉ IP của tên máy đó ngay cho resolver. Ngược lại, server cục bộ sẽ truy vấn đến một Root Name Server gần nhất mà nó biết được. Root Name Server sẽ trả lời địa chỉ IP của Name Server quản lý miền au. Máy chủ name server cục bộ lại hỏi tiếp name server quản lý miền au và được tham chiếu đến máy chủ quản lý miền gov.au. Máy chủ quản lý gov.au chỉ dẫn máy name server cục bộ tham chiếu đến máy chủ quản lý miền gbrmpa.gov.au. Cuối cùng máy name server cục bộ truy vấn máy chủ quản lý miền gbrmpa.gov.au và nhận được câu trả lời. Các loại truy vấn : truy vấn có thể ở 2 dạng :

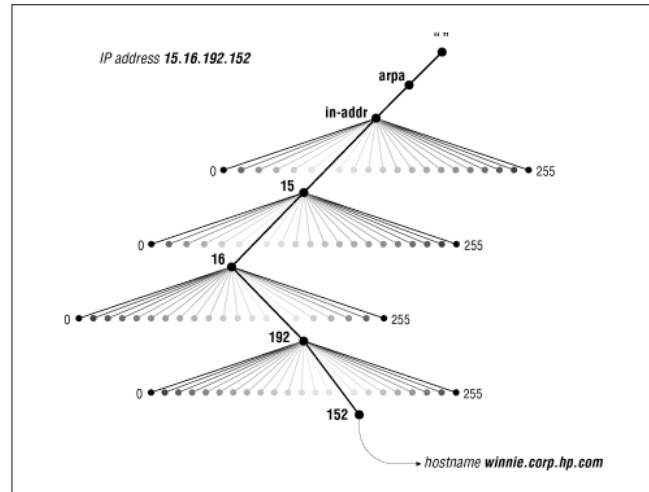
- Truy vấn đệ quy (recursive query) : Khi nameserver nhận được truy vấn dạng này, nó bắt buộc phải trả về kết quả tìm được hoặc thông báo lỗi nếu như truy vấn này không phân giải được. Nameserver không thể tham chiếu truy vấn đến một name server khác. Nameserver có thể gửi truy vấn dạng đệ quy hoặc tương tác đến nameserver khác nhưng nó phải thực hiện cho đến khi nào có kết quả mới thôi.
- Truy vấn tương tác: khi nameserver nhận được truy vấn dạng này, nó trả lời cho resolver với thông tin tốt nhất mà nó có được vào thời điểm lúc đó. Bản thân nameserver không thực hiện bất cứ một truy vấn nào thêm. Thông tin tốt nhất trả về có thể lấy từ dữ liệu cục bộ (kể cả cache). Trong trường hợp nameserver không tìm thấy trong dữ liệu cục bộ nó sẽ trả về tên miền và địa chỉ IP của nameserver gần nhất mà nó biết.

III.2. Phân giải IP thành tên máy tính

Ánh xạ địa chỉ IP thành tên máy tính được dùng để diễn dịch các tập tin log cho dễ đọc hơn. Nó còn dùng trong một số trường hợp chứng thực trên hệ thống UNIX (kiểm tra các tập tin .rhost hay host.equiv). Trong không gian tên miền đã nói ở trên dữ liệu -bao gồm cả địa chỉ IP- được lập chỉ mục theo tên miền. Do đó với một tên miền đã cho việc tìm ra địa chỉ IP khá dễ dàng.

Để có thể phân giải tên máy tính của một địa chỉ IP, trong không gian tên miền người ta bổ sung thêm một nhánh tên miền mà được lập chỉ mục theo địa chỉ IP. Phần không gian này có tên miền là in-addr.arpa.

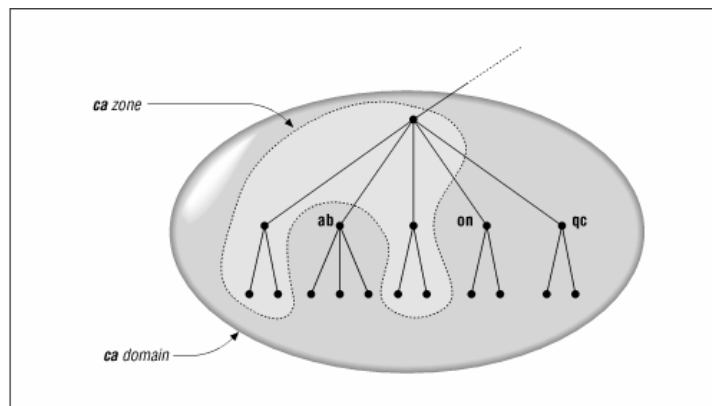
Mỗi nút trong miền in-addr.arpa có một tên nhãn là chỉ số thập phân của địa chỉ IP. **Ví dụ** miền in-addr.arpa có thể có 256 subdomain, tương ứng với 256 giá trị từ 0 đến 255 của byte đầu tiên trong địa chỉ IP. Trong mỗi subdomain lại có 256 subdomain con nữa ứng với byte thứ hai. Cứ như thế và đến byte thứ tư có các bản ghi cho biết tên miền đầy đủ của các máy tính hoặc các mạng có địa chỉ IP tương ứng.



Lưu ý khi đọc tên miền địa chỉ IP sẽ xuất hiện theo thứ tự ngược. Ví dụ nếu địa chỉ IP của máy winnie.corp.hp.com là 15.16.192.152, khi ánh xạ vào miền in-addr.arpa sẽ là 152.192.16.15.in-addr.arpa.

IV. Sự khác nhau giữa domain name và zone

Một miền gồm nhiều thực thể nhỏ hơn gọi là miền con (subdomain). Ví dụ: miền ca bao gồm nhiều miền con như ab.ca, on.ca, qc.ca,...(như hình vẽ dưới). Bạn có thể ủy quyền một số miền con cho những DNS Server khác quản lý. Những miền và miền con mà DNS Server được quyền quản lý gọi là zone. Như vậy, một Zone có thể gồm một miền, một hay nhiều miền con. Hình sau mô tả sự khác nhau giữa zone và domain.





V. Fully Qualified Domain Name (FQDN)

Mỗi nút trên cây có một tên gọi (không chứa dấu chấm) dài tối đa 63 ký tự. Tên riêng dành riêng cho gốc (root) cao nhất và biểu diễn bởi dấu chấm. Một tên miền đầy đủ của một nút chính là chuỗi tuần tự các tên gọi của nút hiện tại đi ngược lên nút gốc, mỗi tên gọi cách nhau bởi dấu chấm. Tên miền có xuất hiện dấu chấm sau cùng được gọi là tên tuyệt đối (absolute) khác với tên tương đối là tên không kết thúc bằng dấu chấm. Tên tuyệt đối cũng được xem là tên miền đầy đủ đã được chứng nhận (fully qualified domain name – FQDN).

VI. Phân loại Domain Name Server

Có nhiều loại Domain Name Server được tổ chức trên Internet. Sự phân loại này tùy thuộc vào nhiệm vụ mà chúng sẽ đảm nhận. Tiếp theo sau đây mô tả những loại Domain Name Server

VI.1. Primary Name Server

Mỗi miền phải có một Primary Name Server. Server này được đăng kí trên Internet để quản lý miền. Mọi người trên Internet đều biết tên máy tính và địa chỉ IP của server này. Người quản trị DNS sẽ tổ chức những tập tin CSDL trên Primary Name Server. Server này có nhiệm vụ phân giải tất cả các máy trong miền hay zone.

VI.2. Secondary Name Server

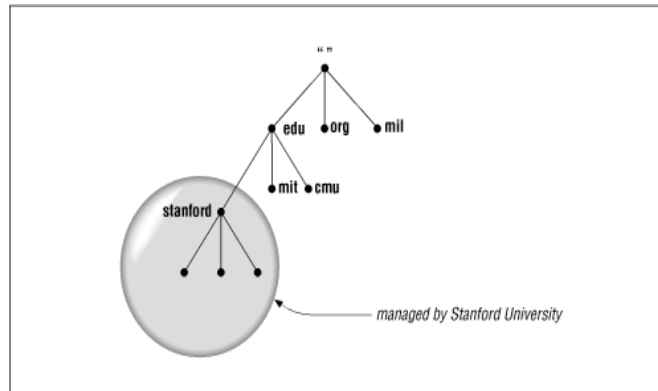
Mỗi miền có một Primary Name Server để quản lý CSDL của miền. Nếu như server này tạm ngưng hoạt động vì một lý do nào đó thì việc phân giải tên máy tính thành địa chỉ IP và ngược lại xem như bị gián đoạn. Việc gián đoạn này làm ảnh hưởng rất lớn đến những tổ chức có nhu cầu trao đổi thông tin ra ngoài Internet cao. Nhằm khắc phục nhược điểm này, những nhà thiết kế đã đưa ra một Server dự phòng gọi là Secondary (hay Slave) Name Server. Server này có nhiệm vụ sao lưu tất cả những dữ liệu trên Primary Name Server và khi Primary Name Server bị gián đoạn thì nó sẽ đảm nhận việc phân giải tên máy tính thành địa chỉ IP và ngược lại. Trong một miền có thể có một hay nhiều Secondary Name Server. Theo một chu kỳ, Secondary sẽ sao chép và cập nhật CSDL từ Primary Name Server. Tên và địa chỉ IP của Secondary Name Server cũng được mọi người trên Internet biết đến.

VI.3. Caching Name Server

Caching Name Server có chức năng phân giải tên máy trên những mạng ở xa thông qua những Name Server khác. Nó lưu giữ lại những tên máy đã được phân giải trước đó và được sử dụng lại những thông tin này nhằm mục đích:

- Làm tăng tốc độ phân giải bằng cách sử dụng cache.
- Giảm bớt gánh nặng phân giải tên máy cho các Name Server.
- Giảm việc lưu thông trên những mạng lớn.

VII. Sự ủy quyền(Delegating Subdomains)



Một trong các mục tiêu khi thiết kế hệ thống DNS là khả năng quản lý phân tán thông qua cơ chế ủy quyền (delegation). Trong một miền có thể tổ chức thành nhiều miền con, mỗi miền con có thể được ủy quyền cho một tổ chức khác và tổ chức đó chịu trách nhiệm duy trì thông tin trong miền con này. Khi đó, miền cha chỉ cần một con trỏ trỏ đến miền con này để tham chiếu khi có các truy vấn. Không phải một miền luôn luôn tổ chức miền con và ủy quyền toàn bộ cho các miền con này, có thể chỉ có vài miền con được ủy quyền. Ví dụ miền hcmuns.edu.vn của Trường ĐHKHTN chia một số miền con như csc.hcmuns.edu.vn (Trung Tâm Tin Học), fit.hcmuns.edu.vn (Khoa CNTT) hay math.hcmuns.edu.vn (Khoa Toán), nhưng các máy chủ phục vụ cho toàn trường thì vẫn thuộc vào miền hcmuns.edu.vn.

VIII. Resource Record (RR)

VIII.1.SOA(Start of Authority)

Trong mỗi tập tin CSDL phải có một và chỉ một record SOA (start of authority). Record **SOA** chỉ ra rằng máy chủ Name Server là nơi cung cấp thông tin tin cậy từ dữ liệu có trong zone. Cú pháp của record SOA:

```
[tên-miền] IN SOA [tên-server-dns] [địa-chỉ-email] (
serial number;
refresh number;
retry number;
experi number;
Time-to-live number)
```

Ví dụ: Khai báo record SOA:

```
t3h.com. IN SOA dnserver.t3h.com. root.t3h.com. (
2005040401 ; Serial
10800 ; Refresh after 3 hours
3600 ; Retry after 1 hour
604800 ; Expire after 1 week
```



86400) ; Minimum TTL of 1 day

Tên miền t3h.com. nằm ở cột đầu tiên. Từ khoá IN chỉ ra lớp (class) dữ liệu là Internet. Có một số lớp dữ liệu khác ngoài Internet nhưng mặc định là IN. Tên xuất hiện sau từ khoá SOA (dnserver.t3h.com.) là tên của primary name server (primary master name server) cho zone này. Tên thứ hai (root.t3h.com.) là địa chỉ e-mail của người có trách nhiệm quản lý dữ liệu trong zone (dấu "." đầu tiên được thay thế cho dấu "@"). Dấu ngoặc cho phép record SOA trải rộng trên nhiều dòng. Các dữ liệu trong phần này chủ yếu dùng cho các máy Secondary Name Server.

- **Serial** : Áp dụng cho mọi dữ liệu trong zone và là 1 số nguyên. Trong ví dụ, giá trị này bắt đầu từ 1 nhưng thông thường người ta sử dụng theo định dạng thời gian như 1997102301. Định dạng này theo kiểu YYYYMMDDNN, trong đó YYYY là năm, MM là tháng, DD là ngày và NN số lần sửa đổi dữ liệu zone trong ngày. Bất kể là theo định dạng nào, luôn luôn phải tăng số này lên mỗi lần sửa đổi dữ liệu zone. Khi máy chủ Secondary liên lạc với máy chủ Primary, trước tiên nó sẽ hỏi số serial. Nếu số serial của máy Secondary nhỏ hơn số serial của máy Primary tức là dữ liệu zone trên Secondary đã cũ và sau đó máy Secondary sẽ sao chép dữ liệu mới từ máy Primary thay cho dữ liệu đang có hiện hành.
- **Refresh**: Chỉ ra khoảng thời gian máy chủ Secondary kiểm tra dữ liệu zone trên máy Primary để cập nhật nếu cần. Trong ví dụ trên thì cứ mỗi 3 giờ máy chủ Secondary sẽ liên lạc với máy chủ Primary để cập nhật dữ liệu nếu có. Giá trị này thay đổi tùy theo tần suất thay đổi dữ liệu trong zone.
- **Retry**: nếu máy chủ Secondary không kết nối được với máy chủ Primary theo thời hạn mô tả trong refresh (ví dụ máy chủ Primary bị shutdown vào lúc đó thì máy chủ Secondary phải tìm cách kết nối lại với máy chủ Primary theo một chu kỳ thời gian mô tả trong retry. Thông thường giá trị này nhỏ hơn giá trị refresh.
- **Expire**: Nếu sau khoảng thời gian này mà máy chủ Secondary không kết nối được với máy chủ Primary thì dữ liệu zone trên máy Secondary sẽ bị quá hạn. Một khi dữ liệu trên Secondary bị quá hạn thì máy chủ này sẽ không trả lời mọi truy vấn về zone này nữa. Giá trị expire này phải lớn hơn giá trị refresh và giá trị retry.
- **TTL**: Viết tắt của time to live. Giá trị này áp dụng cho mọi record trong zone và được đính kèm trong thông tin trả lời một truy vấn. Mục đích của nó là chỉ ra thời gian mà các máy chủ name server khác cache lại thông tin trả lời. Việc cache thông tin trả lời giúp giảm lưu lượng truy vấn DNS trên mạng.

VIII.2.NS (Name Server)

Record tiếp theo cần có trong zone là NS (name server) record. Mỗi name server cho zone sẽ có một NS record. Cú pháp khai báo:

```
[tên-domain] IN NS [DNS-Server_name]
```

Ví dụ: Record NS sau:

```
t3h.com. IN NS dnserver.t3h.com.
```

```
t3h.com. IN NS server.t3h.com.
```

Ví dụ trên chỉ ra 2 nameserver quản lý cơ sở dữ liệu cho miền t3h.com



VIII.3.A (Address) và CNAME (Canonical Name)

Record A (Address) ánh xạ tên máy(hostname) vào địa chỉ IP. Record CNAME (canonical name) tạo tên bí danh alias trỏ vào một tên canonical. Tên canonical là tên host trong record A hoặc lại trỏ vào 1 tên canonical khác.

Cú pháp record A:

[tên-máy-tính] IN A [địa-chỉ-IP]

Ví dụ: Số record A trong tập tin db.t3h

// Hostname ánh xạ vào địa chỉ IP tương ứng

localhost.t3h.com. IN A 127.0.0.1

dnsserver.t3h.com. IN A 172.29.14.2

//Một hostname ánh xạ cho nhiều địa chỉ IP

server.t3h.com. IN A 172.29.14.1

server.t3h.com. IN A 192.253.253.1

// Chỉ định server.t3h.com. ánh xạ về www.t3h.com.

server.t3h.com. IN CNAME www.t3h.com.

VIII.4.MX (Mail Exchange)

DNS dùng record MX trong việc chuyển mail trên mạng Internet. Ban đầu chức năng chuyển mail dựa trên 2 record: record MD (mail destination) và record MF (mail forwarder) records. MD chỉ ra đích cuối cùng của một thông điệp mail có tên miền cụ thể. MF chỉ ra máy chủ trung gian sẽ chuyển tiếp mail đến được máy chủ đích cuối cùng. Tuy nhiên, việc tổ chức này hoạt động không tốt. Do đó, chúng được tích hợp lại thành một record là MX. Khi nhận được mail, trình chuyển mail (mailer) sẽ dựa vào record MX để quyết định đường đi của mail. Record MX chỉ ra một mail exchanger cho một miền - mail exchanger là một máy chủ xử lý (chuyển mail đến mailbox cục bộ hay làm gateway chuyển sang một giao thức chuyển mail khác như UUCP) hoặc chuyển tiếp mail đến một mail exchanger khác (trung gian) gần với mình nhất để đến tới máy chủ đích cuối cùng hơn dùng giao thức SMTP (Simple Mail Transfer Protocol). Để tránh việc gửi mail bị lặp lại, record MX có thêm 1 giá trị bổ sung ngoài tên miền của mail exchanger là 1 số thứ tự tham chiếu. Đây là giá trị nguyên không dấu 16-bit (0-65535) chỉ ra thứ tự ưu tiên của các mail exchanger. Cú pháp record MX:

[tên-domain] IN MX [độ-ưu-tiên] [tên-Mail-Server]

Ví dụ:

t3h.com. IN MX 10 mailserver.t3h.com.

Chỉ ra máy chủ mailserver.t3h.com là một mail exchanger cho miền t3h.com với số thứ tự tham chiếu 10.

Chú ý: Các giá trị này chỉ có ý nghĩa so sánh với nhau:

Ví dụ: khai báo miền t3h.com có hai mail server quản lý là listo.t3h.com và hep.t3h.com quản lý.

- t3h.com. IN MX 1 listo.t3h.com.



- t3h.com. IN MX 2 hep.t3h.com.

Trình chuyển thư mailer sẽ thử phân phát thư đến mail exchanger có số thứ tự tham chiếu nhỏ nhất trước. Nếu không chuyển thư được thì mail exchanger với giá trị kế sau sẽ được chọn. Trong trường hợp có nhiều mail exchanger có cùng số tham chiếu thì mailer sẽ chọn ngẫu nhiên giữa chúng.

VIII.5.PTR (Pointer)

Record PTR (pointer) dùng để ánh xạ địa chỉ IP thành hostname. Cú pháp khai báo:

[địa-chỉ-IP] IN PTR [tên-máy-tính]

Ví dụ các record PTR cho các host trong mạng 192.249.249:

- 1.14.29.172.in-addr.arpa. IN PTR server.t3h.com.
- 2.14.29.172.in-addr.arpa. IN PTR dnsserver.t3h.com.
- 3.14.29.172.in-addr.arpa. IN PTR mailserver.t3h.com.
- 4.14.29.172.in-addr.arpa. IN PTR diehard.t3h.com.

IX. Hoạt động của Name Server trong Linux

DNS name server khi hoạt động sẽ phát sinh ra một daemon có tên là **named**. Trong quá trình khởi động, named đọc các tập tin dữ liệu rồi chờ các yêu cầu phân giải qua cổng xác định trong tập tin /etc/services. Khi nhận được một yêu cầu từ resolver, đầu tiên Named dùng giao thức UDP để truy vấn. Nếu dùng giao thức UDP phân giải không có kết quả, sau đó named sẽ dùng giao thức TCP. Một số đặt điểm cần ghi nhớ trong quá trình truy vấn giữa Client và Server.

- Truy vấn từ Client đến Server sử dụng cổng nguồn là 1023, cổng đích là 53.
- Server trả lời truy vấn về cho sử dụng cổng nguồn là 53, cổng đích là lớn hơn 1023.
- Truy vấn và trả lời giữa các server sử dụng giao thức UDP cổng nguồn và đích đều là 53, với TCP truy vấn của server sẽ sử dụng cổng > 1023.

X. Cài đặt BIND

Hầu hết các phiên bản của RedHat và Fedora Linux cung cấp package BIND.*.rpm(đối với FC là bind-9.2.3-13.*.rpm)...Một số package của BIND(trong Fedora):

- bind-9.2.3-13.i386.rpm : Là package chính của DNS Server.
- bind-libs-9.2.3-13.386.rpm : Cung cấp các thư viện trợ giúp cho DNS Server.
- bind-utils-9.2.3-13.386.rpm : Cung cấp các tiện ích tích hợp cho DNS Server.
- system-config-bind-2.0.2-5.386.rpm : Cung cấp giao diện cấu hình DNS Server trên môi trường XWindows.
- caching-nameserver-7.2-12.386.rpm : Là package cung cấp các file mẫu hỗ trợ cấu hình Caching nameserver và cấu hình dịch vụ DNS.
- caching-nameserver-ltsp-7.2-k12ltsp.5.3.0.386.rpm : Là package cung cấp các file cấu hình mẫu cho zone ltsp.
- bind-chroot-9.2.3-13.i386.rpm : là package cung cấp một số tính năng bảo mật mới để giới hạn truy xuất file cấu hình của dịch vụ DNS.



Ta dùng lệnh `rpm -ivh` để cài đặt các package trên.

X.1. Một số file cấu hình quan trọng

RedHat/Fedora BIND hoạt động trong hệ thống dưới dạng tiến trình `named` do user có tên `named` làm chủ sở hữu. Để tăng tính năng bảo mật trong hệ thống Fedora, BIND cung cấp thêm package `bind-chroot-9.2.3-13.i386.rpm` để giới hạn việc truy xuất vào các file cấu hình của `named`, khi ta cài `chroot` package thì `named` xem thư mục `/var/named/chroot` là thư mục gốc, các file `/var/named/chroot/etc/named.conf` là tập tin khai báo zone, `/var/named/chroot/var/named/` là thư mục lưu trữ file cơ sở dữ liệu. Khi ta dùng `chroot` thì tất cả các file cấu hình `named` đều được đưa vào thư mục `/var/named/chroot`. Nếu ta không sử dụng package này nghĩa là ta loại bỏ `bind-chroot-9.2.3-13.i386.rpm` thì các file mô tả thông tin cấu hình DNS được lưu tại:

- `/etc/named.conf`
- `/var/named/`
- `/etc/rndc.key`, `/etc/rndc.conf` là các file hỗ trợ cho vấn đề chứng thực trong `named`.

X.2. Cấu hình

Trước khi cấu hình những Name Server chúng ta cần phải trải qua những bước sau:

- Tạo hoặc mở tập tin `/etc/named.conf`
- Cấu hình zone file (forward zone file, reverse zone file)
- Cấu hình DNS client

X.2.1 Cấu hình tập tin `/etc/named.conf`

Tập tin này chứa những thông tin quan trọng được sử dụng bởi daemon `named` khi daemon này khởi động. Nội dung của tập tin này như sau:

```
options {
    ; Chỉ định các tùy chọn
    directory "/var/named";      Thư mục lưu trữ file cơ sở dữ liệu của zone
    forwarders {172.29.2.2};     Chỉ định truy vấn đệ quy lên server khi truy vấn ra ngoài
};

// Khai báo caching zone name
zone "." {
    type hint;
    file "named.ca";
};

// Khai báo zone thuận cục bộ localhost
zone "localhost" {
    type master;
    file "localhost.zone";
    allow-update { none; };
};
```



```
};
// Khai báo zone nghịch cục bộ localhost
zone "0.0.127.in-addr.arpa" {
    type master;
    file "named.local";
    allow-update { none; };
};
//khai báo zone thuận cục bộ t3h.com
zone "t3h.com" {
    type master;
    file "named.hosts";
allow-query { any; };
};
//khai báo zone nghịch cho zone t3h.com
zone "14.29.172.in-addr.arpa" {
    type master;
    file "named.rev";
allow-query { any; };
};
```

- Directory: Thư mục làm việc của Server. Bất kỳ những đường dẫn không tuyệt đối nào đều ánh xạ đến thư mục này. Cú pháp: directory “[tên-thư-mục]”
- **forwarders**: chỉ ra những địa chỉ IP của các name server mà nó sẽ gửi yêu cầu truy vấn khi có nhu cầu. Cú pháp: forwarders {[địa-chỉ-IP];...}; Nameserver cục bộ sẽ truy vấn Name Server có địa chỉ IP 172.29.2.2 khi có một yêu cầu không phân giải được.

Ví dụ:

```
options {
    directory "/var/named";
    forwarders {172.29.2.2;};
};
```

- **zone**: định nghĩa một zone để quản lý CSDL cho miền hay miền con. Cú pháp khai báo:

```
zone [ten-mien] IN {
    type master/slave/hint/stub;
    [ masters [ port ip_port ] { ip_addr [key key_id]; [ ... ] }; ]
    file path_name;
};
```



- **type:** chỉ ra loại name server:
- **master:** server có bản copy chính cơ sở dữ liệu.
- **Slave:** server lưu một bản sao CSDL từ master. Nếu một tập tin được chỉ ra nó sẽ sao chép toàn bộ zone master về.
- **Stub:** tương tự như slave nhưng chỉ sao chép record NS từ Master chứ không phải toàn bộ dữ liệu.
- **Hint:** zone chỉ ra những root name server
- **masters:** chỉ ra địa chỉ IP của master name server (sử dụng trong khai báo secondary zone)
- **file:** tập tin định nghĩa CSDL.

X.2.2 Cấu hình zone file

- Tạo tập tin CSDL phân giải tên máy tính thành địa chỉ IP
- Tạo tập tin CSDL phân giải tên địa chỉ IP thành tên máy tính

Sau đây là tuần tự những bước:

Bước 1: Tạo tập tin cơ sở dữ liệu chuyển đổi tên máy tính thành địa chỉ IP, Tập tin này lưu danh sách tất cả những máy tính trong miền. Nó được dùng để phân giải tên máy (hostname) thành địa chỉ IP. Những record khác như: CNAME, MX cũng được định nghĩa trong tập tin này.

Ví dụ:

```
@           IN SOA dnsserver.t3h.com. root.t3h.com. (
2001112800;
10800;
1800;
36000;
86400)
IN  NS     dnsserver.t3h.com.
IN  MX     0 mailserver.t3h.com.
dnsserver  IN  A      172.29.14.2
server    IN  A      172.29.14.1
mailserver IN  A      172.29.14.3
www       IN  CNAME  server.t3h.com.
```

Bước 2: Tạo tập tin cơ sở dữ liệu chuyển đổi địa chỉ IP thành tên máy tính. Tập tin này được sử dụng để phân giải địa chỉ IP thành tên máy.

Ví dụ:

```
@           IN  SOA  dnsserver.t3h.com. root.t3h.com. (
2001112800;
10800;
1800;
3600000;
86400)
IN  NS     dnsserver.t3h.com.
IN  MX     0 mailserver.t3h.com.
2   IN  PTR  dnsserver.t3h.com.
```



1 *IN PTR* *server.t3h.com.*

X.2.3 Cấu hình DNS Client

Cấu hình DNS Client nhằm sử dụng công cụ nslookup kiểm tra những Name Server vừa cấu hình. Trong Linux, những thông số cấu hình DNS client được lưu trong tập tin /etc/resolv.conf. Tập tin /etc/resolv.conf dùng để quyết định DNS Server cụ thể cần phải truy vấn và cách bổ sung phần tên miền cho phần tên của máy. Nội dung của tập tin có dạng sau:

```
nameserver [địa-chỉ-IP-của-Name-Server]
domain [tên-miền]
```

Trong đó:

- **nameserver:** dùng để định nghĩa máy chủ DNS mà resolver sẽ gửi yêu cầu phân giải tên hoặc địa chỉ IP khi có nhu cầu. Sau từ khoá Name Server là địa chỉ IP của Name Server.
- **domain:** sẽ được nối thêm vào sau tên máy tính khi resolver gửi yêu cầu đến server.

Ví dụ: Về nội dung tập tin /etc/resolver

```
nameserver 172.29.14.2
domain t3h.com
```

XI. Kiểm tra hoạt động của DNS

Khi đã hoàn thành các thao tác cần thiết cấu hình cho máy chủ DNS, ta nên kiểm tra lại để khẳng định những cấu hình này đã đúng hay còn sai sót những điểm nào. Một công cụ đặc lực giúp kiểm tra cấu hình dns là nslookup, hoặc lệnh host.

Lệnh nslookup:

```
#nslookup
```

```
Default Server: dnsserver.t3h.com
```

```
Address: 172.29.14.2
```

```
>www.t3h.com
```

```
Server: dnsserver.t3h.com
```

```
Address: 172.29.14.2
```

```
Name: WebServer.t3h.com
```

```
Address: 172.29.14.41
```

```
Aliases: www.t3h.com
```

Kiểm tra các record SOA, NS, MX của miền bằng lệnh:

```
>Set type=any
```

```
>domain_name
```

Ví dụ:

```
>set type=mx
```



```
>t3h.com
server: dnsserver.t3h.com
address: 172.29.14.2
dnsserver.t3h.com preference=0, mail exchanger=mailserver.t3h.com
t3h.com nameserver=dnsserver.t3h.com
dnsserver.t3h.com internet address=172.29.14.2
Lệnh host
# host www.linuxhomenetworking.com
www.linuxhomenetworking.com has address 65.115.71.34
# host 65.115.71.34
34.71.115.65.in-addr.arpa domain name pointer 65-115-71-34.myisp.net.
```

XII. Cấu hình Secondary Name Server

Cấu hình Secondary Name Server tương tự như cấu hình Primary Name Server nhưng có một số điểm khác sau:

- Không tạo các tập tin CSDL cho zone. Những tập tin này sẽ tự động được sao chép từ Primary Name Server về lưu tại máy một bản.
- Trong tập tin /etc/named.conf thay thế thuộc tính type là master thành slave.
- Cung cấp địa chỉ IP của Primary Name Server.
- Ví dụ:

```
zone "netlab.vnedu.net"{
    type slave;
    file "sec/netlab.vnedu.net";
    masters{
        172.29.9.199;
    };
zone "29.29.192.in-addr.arpa"{
    type slave;
    file "named.rev";
    masters {192.29.29.1;};
};
```

XIII. Một số quy ước

Cột thứ 2 trong khai báo zone của tập tin /etc/named (zone "t3h.com" hay zone "14.29.172.in-addr.arpa") có thể giúp ta một số khai báo nhanh chóng trong tập tin cơ sở dữ liệu như sau :



dnsserver.t3h.com. IN A 172.29.14.2

Có thể viết là :

dnsserver IN A 172.29.14.2

Khai báo

2.14.29.172.in-addr.arpa. IN PTR dnsserver.t3h.com.

Có thể viết là :

2 IN PTR dnsserver.t3h.com.

Khai báo

@ IN SOA dnsserver.t3h.com. root.t3h.com. (

1; serial

10800 ; refresh after 3 hours

36000; retry after 1 hour

604800; expire after 1 week

86400) ; minimum TTL of 1 day

tương đương với :

net.hcmuns.edu.vn. IN SOA ...

...

Nếu cột đầu tiên của một entry trong tập tin cơ sở dữ liệu là các khoảng trắng hay spacebar thì nó sẽ lấy giá trị cột tương ứng của resource record ngay dòng trên của nó.

Ví dụ :

webserver IN A 172.29.14.41

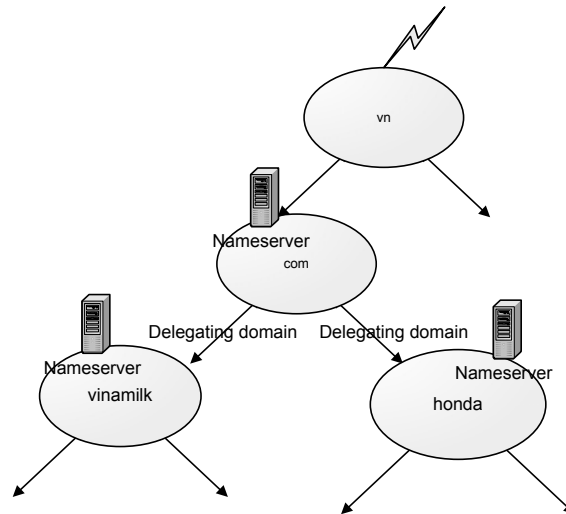
webserver IN A 172.29.14.42

Có thể viết là :

webserver IN A 172.29.14.41

IN A 172.29.14.42

XIV. Cấu hình sự ủy quyền cho các miền con



Thông thường miền cha cung cấp các domain con cho miền con dưới hình thức ủy quyền cho miền con tự quản lý và tổ chức cơ sở dữ liệu cho miền con (thuật ngữ này thường được gọi là delegation domain), hoặc miền cha tạo hosting domain cho miền con (theo cách này thì miền cha phải tổ chức và quản lý cơ sở dữ liệu cho miền con). Dựa vào sơ đồ trên ta thực hiện các thao tác cơ bản sau để thực hiện công việc cung cấp subdomain qua cơ chế ủy quyền cho các nameserver quản lý cơ sở dữ liệu của miền con. Tại nameserver quản lý cơ sở dữ liệu cho miền com.vn ta mô các thông tin sau để thực hiện cơ chế ủy quyền cho hai miền con vinamilk.com.vn và honda.com.vn cho hai server vinamilkserv, hondaserv quản lý:

```
vinamilkserv IN A <ipaddress1>
vinamilk      IN NS  vinamilkserv.com.vn.
```

Trong đó ipaddress1 là địa chỉ IP của nameserver quản lý cơ sở dữ liệu cho miền vinamilk.com.vn. Sau đó ta cần mô tả RR PTR cho vinamilkserv trong file mô tả cơ sở dữ liệu cho zone nghịch.

```
<host_id> IN PTR  vinamilkserv.com.vn
```

Tương tự ta có thể ủy quyền miền honda.com.vn cho hondaserv.

```
hondaserv IN A <ipaddress2>
honda     IN NS  hondaserv.com.vn.
```

Trong đó ipaddress2 là địa chỉ IP của nameserver quản lý cơ sở dữ liệu cho miền honda.com.vn. Sau đó ta cần mô tả RR PTR cho hondaserv trong file mô tả cơ sở dữ liệu cho zone nghịch.

```
<host_id> IN PTR  hondaserv.com.vn
```

Lưu ý: Ở miền con ta cần mô tả forwarders{ipaddress;...} lên miền cha để miền con nhờ nameserver của miền cha phân giải tên miền bên ngoài cho miền con.



BÀI 13

File Transfer Protocol

Tóm tắt

Lý thuyết: 5 tiết - Thực hành: 5 tiết.

Mục tiêu	Các mục chính	Bài tập bắt buộc	Bài tập làm thêm
Bài học giới thiệu cơ chế cấu hình và tổ chức quản trị dịch vụ FTP	I. Giới thiệu về FTP II. Chương trình FTP Server III. Chương trình FTP client IV. Giới thiệu VsFTP V. Cấu hình Virtual FTP Server	Bài tập 3.1 (Dịch vụ FTP)	



I. Giới thiệu về FTP

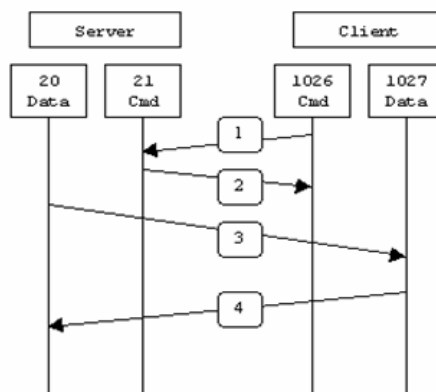
I.1. Giao thức FTP

FTP là từ viết tắt của File Transfer Protocol. Giao thức này được xây dựng dựa trên chuẩn TCP, FTP cung cấp cơ chế truyền tin dưới dạng file thông qua mạng TCP/IP, FTP là 1 dịch vụ đặc biệt vì nó dùng đến 2 cổng: cổng 20 dùng để truyền dữ liệu (data port) và cổng 21 dùng để truyền lệnh (command port).

I.1.1 Active FTP

Ở chế độ chủ động (active), máy khách FTP (FTP client) dùng 1 cổng ngẫu nhiên không dành riêng (cổng $N > 1024$) kết nối vào cổng 21 của FTP server. Sau đó, máy khách lắng nghe trên cổng $N+1$ và gửi lệnh PORT $N+1$ đến FTP server. Tiếp theo, từ cổng dữ liệu của mình, FTP server sẽ kết nối ngược lại vào cổng dữ liệu của client đã khai báo trước đó (tức là $N+1$), Ở khía cạnh firewall, để FTP Server hỗ trợ chế độ active các kênh truyền sau phải mở:

- Cổng 21 phải được mở cho bất cứ nguồn gửi nào (để client khởi tạo kết nối)
- FTP server's port 21 to ports > 1024 (Server trả lời về cổng điều khiển của client)
- Cho kết nối từ cổng 20 của FTP server đến các cổng > 1024 (Server khởi tạo kết nối vào cổng dữ liệu của client)
- Nhận kết nối hướng đến cổng 20 của FTP server từ các cổng > 1024 (Client gửi xác nhận ACKs đến cổng data của server)



Sơ đồ kết nối

- Bước 1, client khởi tạo kết nối vào cổng 21 của server và gửi lệnh PORT 1027.
- Bước 2 server gửi xác nhận ACK về cổng lệnh của client.
- Bước 3 server khởi tạo kết nối từ cổng 20 của mình đến cổng dữ liệu mà client đã khai báo trước đó.
- Bước 4 client gửi ACK phản hồi cho server.



Khi FTP Server hoạt động ở chế độ chủ động, Client không tạo kết nối thật sự vào cổng dữ liệu của FTP server, mà chỉ đơn giản là thông báo cho server biết rằng nó đang lắng nghe trên cổng nào và server phải kết nối ngược về client vào cổng đó. Trên quan điểm firewall đối với máy client điều này giống như 1 hệ thống bên ngoài khởi tạo kết nối vào hệ thống bên trong và điều này thường bị ngăn chặn trên hầu hết các hệ thống Firewall.

Ví dụ: Phiên làm việc active FTP:

Trong ví dụ này phiên làm việc FTP khởi tạo từ máy `testbox1.slacksite.com` (192.168.150.80), dùng chương trình FTP client dạng dòng lệnh, đến máy chủ FTP `testbox2.slacksite.com` (192.168.150.90). Các dòng có dấu `-->` chỉ ra các lệnh FTP gửi đến server và thông tin phản hồi từ các lệnh này. Các thông tin người dùng nhập vào dưới dạng chữ đậm.

Lưu ý Khi lệnh `PORT` được phát ra trên client được thể hiện ở 6 byte. 4 byte đầu là địa chỉ IP của máy client còn 2 byte sau là số cổng. Giá trị cổng được tính bằng $(\text{byte}_5 * 256) + \text{byte}_6$, ví dụ $((14 * 256) + 178)$ là 3762.

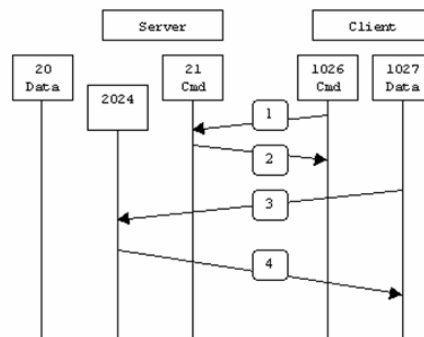
```
testbox1: {/home/p-t/slacker/public_html} %ftp -d testbox2
Connected to testbox2.slacksite.com.
220 testbox2.slacksite.com FTP server ready.
Name (testbox2:slacker): slacker
---> USER slacker
331 Password required for slacker.
Password: TmpPass
---> PASS XXXX
230 User slacker logged in.
---> SYST
215 UNIX Type: L8
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
ftp: setsockopt (ignored): Permission denied
---> PORT 192,168,150,80,14,178
200 PORT command successful.
---> LIST
150 Opening ASCII mode data connection for file list.
drwx----- 3 slacker users    104 Jul 27 01:45 public_html
226 Transfer complete.
ftp> quit
---> QUIT
221 Goodbye.
```

1.1.2 Passive FTP

Để giải quyết vấn đề là server phải tạo kết nối đến client, một phương thức kết nối FTP khác đã được phát triển. Phương thức này gọi là FTP thụ động (passive) hoặc PASV (là lệnh mà client gửi cho server để báo cho biết là nó đang ở chế độ passive).

Ở chế độ thụ động, FTP client tạo kết nối đến server, tránh vấn đề Firewall lọc kết nối đến cổng của máy bên trong từ server. Khi kết nối FTP được mở, client sẽ mở 2 cổng không dành riêng N, N+1 ($N > 1024$). Cổng thứ nhất dùng để liên lạc với cổng 21 của server, nhưng thay vì gửi lệnh PORT và sau đó là server kết nối ngược về client, thì lệnh PASV được phát ra. Kết quả là server sẽ mở 1 cổng không dành riêng bất kỳ P ($P > 1024$) và gửi lệnh PORT P ngược về cho client.. Sau đó client sẽ khởi tạo kết nối từ cổng N+1 vào cổng P trên server để truyền dữ liệu. theo quan điểm Firewall trên server FTP, để hỗ trợ FTP chế độ passive, các kênh truyền sau phải được mở:

- Cổng FTP 21 của server nhận kết nối từ bất kỳ nguồn nào (cho client khởi tạo kết nối)
- Cho phép trả lời từ cổng 21 FTP server đến cổng bất kỳ trên 1024 (Server trả lời cho cổng control của client)
- Nhận kết nối trên cổng FTP server > 1024 từ bất cứ nguồn nào (Client tạo kết nối để truyền dữ liệu đến cổng ngẫu nhiên mà server đã chỉ ra)
- Cho phép trả lời từ cổng FTP server > 1024 đến các cổng > 1024 (Server gửi xác nhận ACKs đến cổng dữ liệu của client)



Sơ đồ kết nối Passive FTP

- + Bước 1, client kết nối vào cổng lệnh của server và phát lệnh PASV.
- + Bước 2 server trả lời bằng lệnh PORT 2024, cho client biết cổng 2024 đang mở để nhận kết nối dữ liệu.
- + Bước 3 client tạo kết nối truyền dữ liệu từ cổng dữ liệu của nó đến cổng dữ liệu 2024 của server.
- + Bước 4 là server trả lời bằng xác nhận ACK về cho cổng dữ liệu của client.

Trong khi FTP ở chế độ thụ động giải quyết được vấn đề phía client thì nó lại gây ra nhiều vấn đề khác ở phía server. Thứ nhất là cho phép máy ở xa kết nối vào cổng bất kỳ > 1024 của server. Điều này khá nguy hiểm trừ khi FTP cho phép mô tả dãy các cổng ≥ 1024 mà FTP server sẽ dùng (ví dụ WU-FTP Daemon). Vấn đề thứ hai là một số FTP client lại không hỗ trợ chế độ thụ động. Ví dụ tiện ích FTP client mà Solaris cung cấp không hỗ trợ FTP thụ động. Khi đó cần phải có thêm trình FTP client. Một lưu ý là hầu hết các trình duyệt Web chỉ hỗ trợ FTP thụ động khi truy cập FTP server theo đường dẫn URL ftp://.

Ví dụ phiên làm việc passive FTP:



Trong ví dụ này phiên làm việc FTP khởi tạo từ máy `testbox1.slacksite.com` (192.168.150.80), dùng chương trình FTP client dạng dòng lệnh, đến máy chủ FTP `testbox2.slacksite.com` (192.168.150.90), máy chủ Linux chạy ProFTPd 1.2.2RC2. Các dòng có dấu `-->` chỉ ra các lệnh FTP gửi đến server và thông tin phản hồi từ các lệnh này. Các thông tin người nhập vào dưới dạng chữ đậm.

Lưu ý: Đối với FTP thụ động, cổng mà lệnh `PORT` mô tả chính là cổng sẽ được mở trên server. Còn đối với FTP chủ động cổng này sẽ được mở ở client.

```
testbox1: {/home/p-t/slacker/public_html} %ftp -d testbox2
Connected to testbox2.slacksite.com.
220 testbox2.slacksite.com FTP server ready.
Name (testbox2:slacker): slacker
---> USER slacker
331 Password required for slacker.
Password: TmpPass
---> PASS XXXX
230 User slacker logged in.
---> SYST
215 UNIX Type: L8
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> passive
Passive mode on.
ftp> ls
ftp: setsockopt (ignored): Permission denied
---> PASV
227 Entering Passive Mode (192,168,150,90,195,149).
---> LIST
150 Opening ASCII mode data connection for file list
drwx----- 3 slacker users 104 Jul 27 01:45 public_html
226 Transfer complete.
ftp> quit
---> QUIT
221 Goodbye.
```

II. Chương trình FTP Server

FTP Server là máy chủ lưu giữ những tài nguyên và hỗ trợ giao thức FTP để giao tiếp với những máy tính khác cho phép truyền dữ liệu trên Internet. Một số chương trình ftp server sử dụng trên Linux:

- Vsftpd
- Wu-ftp
- PureFTPd
- ProFTPD



III. Chương trình FTP client

Là chương trình giao tiếp với FTP Server, hầu hết các hệ điều hành đều hỗ trợ ftp client, trên linux hoặc Windows để mở kết nối tới FTP Server ta dùng lệnh #ftp <ftp_address>. Để thiết lập một phiên giao dịch, ta cần phải có địa chỉ IP (hoặc tên máy tính), một tài khoản (username, password). Username mà FTP hỗ trợ sẵn cho người dùng để mở một giao dịch FTP có tên là anonymous với password rỗng. Sau đây là một ví dụ về mở một phiên giao dịch đến FTP Server:

```
[markr@amber markr]$ ftp ftp.redhat.com
Connected to ftp.redhat.com.
220 FTP server ready:
Name (ftp.redhat.com:markr): anonymous
331 Guest login ok, send your complete e-mail address as password.
Password:
230- THE SOFTWARE AVAILABLE FROM THIS SITE IS PROVIDED AND LICENSED
230- "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR
230- IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
230- OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.
230-
230- Priority FTP access (via priority.redhat.com) is for the use
230- of registered users only. Red Hat reserves the right to cancel
230- access at its discretion.
230-
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files
ftp>
```

Một số tập lệnh của ftp client:

Tên lệnh	Cú pháp	Ý nghĩa
? hoặc lệnh help	? [command]	Hiển thị giúp đỡ về [command]
append	append local-file [remote-file]	Ghép một file cục bộ với 1 file trên Server
ascii	ASCII	Chỉ định kiểu truyền file là ascii (đây là kiểu truyền mặc định).
binary	binary	Chỉ định kiểu truyền file là binary (đây là kiểu truyền mặc định).
bye	bye	Kết thúc ftp session
cd	cd remote-directory	Thay đổi đường dẫn thư mục trên FTP Server
delete	delete remote-file	Xóa file trên FTP Server
dir	dir remote-directory	Liệt kê danh sách file



get	get remote-file [local-file]	Download file từ FTP Server về máy cục bộ
lcd	lcd [directory]	Thay đổi thư mục trên máy cục bộ
ls	ls [remote-directory] [local-file]	Liệt kê các tập tin và thư mục
mdelete	mdelete remote-files [...]	Xóa nhiều file
mget	mget remote-files [...]	Download nhiều file
mkdir	mkdir directory	Tạo thư mục
put	put local-file [remote-file]	Upload tập tin
mput	mput local-files [...]	Upload nhiều tập tin
open	open computer [port]	Kết nối tới ftp server
prompt	prompt	Tắt cơ chế confirm sau mỗi lần download file
disconne ct	disconnect	Hủy kết nối FTP
Pwd	pwd	Xem thư mục hiện tại
quit	quit	Thoát khỏi ftp session
recv	recv remote-file [local-file]	Copy file từ remote về local
Rename	rename filename newfilename	Thay đổi tên file
rmdir	rmdir directory	Xóa thư mục
Send	send local-file [remote-file]	Copy file từ local đến remote
User	user user-name [password] [account]	Chuyển đổi user khác



IV. Giới thiệu VsFTP

Vsftpd là một package mới giúp cấu hình ftp server trong RedHat Linux 9. Vsftpd (Very Secure FTP Daemon) được phát triển xoay quanh tính năng nhanh, ổn định và an toàn. VsFTP có khả năng quản lý số lượng kết nối lớn một cách hiệu quả và an toàn.

IV.1. Những tập tin được cài đặt liên quan đến vsftpd

Sau đây liệt kê những tập tin và thư mục thường được quan tâm khi cấu hình vsftpd server:

- `/etc/pam.d/vsftpd`: Tập tin cấu hình PAM cho vsftpd. Tập tin này định nghĩa những yêu cầu mà người dùng phải cung cấp khi đăng nhập vào ftp server.
- `/etc/vsftpd/vsftpd.conf`: tập tin cấu hình vsftpd server.
- `/etc/vsftpd.ftpusers`: liệt kê những người dùng không được login vào vsftpd. Mặc định danh sách những người dùng này gồm root, bin, daemon và những người dùng khác
- `/etc/vsftpd.user_list`: tập tin này được cấu hình để cấm hay cho phép những người dùng được liệt kê truy cập ftp server. điều này phụ thuộc vào tùy chọn `userlist_deny` được xét YES hay NO trong tập tin `vsftpd.conf`. Nếu những người dùng đã liệt kê trong tập tin này thì không được xuất hiện trong `vsftpd.ftpusers`
- `/var/ftp/`: thư mục chứa những tập tin đáp ứng cho vsftpd. Nó cũng chứa thư mục `pub` cho người dùng anonymous. Thư mục này chỉ có thể đọc, chỉ có root mới có khả năng ghi.

IV.2. Khởi động và dừng vsftpd

Sau khi cài đặt phần mềm VSFTPD hoặc sau khi ta thay đổi cấu hình, ta phải tiến hành kích hoạt dịch vụ FTP. Quá trình khởi động lại sẽ giúp cho Daemon VSFTPD cập lại các thông số mà ta đã thay đổi, sử dụng lệnh `chkconfig vsftpd on` để đặt dịch vụ FTP là system services. Một số lệnh cần sử dụng khi ta muốn khởi động lại dịch vụ FTP:

```
#service vsftpd start/stop/restart
```

Hoặc sử dụng lệnh

```
#/etc/init.d/vsftpd start/stop/restart
```

IV.3. Một số thông số cấu hình mặc định

Mặc định dịch vụ FTP sử dụng phần mềm VSFTPD cho phép người dùng anonymous, người dùng cục bộ trong hệ thống được quyền login vào FTP Server, chỉ có user root và những user khác có `UID<100` không được login.

- Đối với anonymous được login vào FTP server và có thư mục gốc `/var/ftp` với quyền truy xuất read (đọc và truy xuất tài liệu).
- Đối với người dùng cục bộ (localuser) được quyền login vào dịch vụ FTP và có thư mục FTP root là `/home/username` (username là tên user login) với quyền read, write.



IV.4. Những tùy chọn cấu hình vsftpd

Tất cả những cấu hình của vsftpd được lưu giữ trong tập tin cấu hình `/etc/vsftpd/vsftpd.conf`. Mỗi tùy chọn trong tập tin có định dạng sau: `<tùy chọn>=<value>`, những dòng chú thích được đánh dấu #

Daemon:

- Listen: Khi nó có giá trị YES thì VSFTPD chạy trong chế độ standalone. Thuộc tính này không được xét với `listen_ipv6`, giá trị mặc định là YES.
- Session_support: nếu tùy chọn này có giá trị là YES thì vsftpd cố gắng quản lý giao dịch login của người dùng ngang qua PAM (Pluggable Authentication Modules), giá trị mặc định là YES.

Đăng nhập và điều khiển truy cập:

- anonymous_enable: nếu tùy chọn này có giá trị là YES thì người dùng anonymous được phép login vào, giá trị mặc định YES
- banned_email_file: Nếu tùy chọn `deny_email_enable` được xét là YES, tùy chọn này chỉ ra tập tin chứa danh sách những password email của anonymous không cho phép truy cập đến server, giá trị mặc định: `/etc/vsftpd.banned_emails`
- banner_file: chỉ ra tập tin text sẽ được hiển thị khi kết nối đến server được thiết lập.
- cmds_allowed: chỉ ra danh sách những lệnh ftp (phân cách nhau bởi dấu phẩy) được cho phép bởi ftp server. Tất cả những lệnh khác sẽ bị từ chối.
- deny_email_enable: nếu tùy chọn này có giá trị là YES thì người dùng anonymous sử dụng password được chỉ ra trong tập tin `/etc/vsftpd.banned_emails` bị cấm truy cập đến server, giá trị mặc định là NO
- ftpd_banner: nếu tùy chọn này có giá trị là YES thì chuỗi được chỉ ra trong tùy chọn này sẽ hiển thị dòng thông tin mô tả khi người dùng thiết lập kết nối với server. Tùy chọn này sẽ ghi đè lên `banner_file`. Mặc định vsftpd hiển thị banner chuẩn.
- local_enable: nếu tùy chọn này có giá trị là YES thì những người dùng cục bộ được login vào hệ thống.
- userlist_deny: Được sử dụng khi tùy chọn `userlist_enable` được đặt là NO, tất cả những người dùng cục bộ bị cấm truy cập trừ những người dùng được chỉ ra trong `userlist_file`. Bởi vì những truy cập bị cấm trước khi client được yêu cầu nhập vào password, đặt tùy chọn này là NO để ngăn chặn những người dùng cục bộ gửi password không mã hóa trên mạng, giá trị mặc định là YES.
- userlist_enable: nếu tùy chọn này có giá trị là YES thì những người dùng được chỉ ra trong tập tin trong `userlist_file` bị cấm truy cập. Bởi vì client bị cấm trước khi client nhập password, người dùng bị ngăn chặn gửi password không mã hóa trên mạng, mặc định là YES.
- userlist_file: chỉ ra tập tin liệt kê danh sách các người dùng, giá trị mặc định `/etc/vsftpd.user_list`.

Người dùng Anonymous:

- anon_mkdir_write_enable: nếu tùy chọn này có giá trị là YES và kết hợp với `write_enable=YES` thì người dùng anonymous được phép tạo thư mục mới trong thư mục cha có quyền write, giá trị mặc định là NO



- anon_root: chỉ ra thư mục vsftpd trao đổi khi người dùng anonymous login vào
- anon_upload_enable: nếu tùy chọn này có giá trị là YES và cùng với write_enable=YES thì người dùng anonymous được phép upload tập tin trong thư mục cha với quyền ghi, giá trị mặc định là NO
- anon_world_readable_only: nếu tùy chọn này có giá trị là YES thì người dùng anonymous chỉ được phép download những tập tin có quyền đọc, giá trị mặc định là YES
- ftp_username: chỉ ra người dùng cục bộ được sử dụng cho anonymous ftp server. Home directory được chỉ ra trong tập tin /etc/passwd cho người dùng là thư mục gốc của anonymous ftp server, giá trị mặc định là ftp
- no_anon_password: nếu tùy chọn này có giá trị là YES thì người dùng anonymous sẽ không yêu cầu nhập password, giá trị mặc định là NO

Người dùng cục bộ:

- Những tùy chọn liệt kê sau đây sẽ ảnh hưởng đến cách truy cập của người dùng cục bộ đến server. Để sử dụng những tùy chọn này, tùy chọn local_enable=YES
- local_enable: cho phép người dùng cục bộ truy cập đến ftp server
- chmod_enable: cho phép người dùng được phép thay đổi quyền hạn trên tập tin, giá trị mặc định là YES
- chroot_local_user: nếu tùy chọn này có giá trị là YES thì người dùng có thể di chuyển đến home directory của họ sau khi login vào, giá trị mặc định là NO
- guest_enable: nếu tùy chọn này có giá trị là YES thì người dùng anonymous login vào như guest, mà được chỉ ra trong guest_username, giá trị mặc định là NO
- guest_username: chỉ ra username của người dùng guest, giá trị mặc định là ftp
- local_root: Chỉ ra thư mục vsftpd sau khi người dùng cục bộ login vào

Thư mục:

- dirlist_enable: Nếu tùy chọn này có giá trị là YES thì các người dùng được phép xem nội dung của thư mục, giá trị mặc định là YES
- dirmessage_enable: Nếu tùy chọn này có giá trị là YES thì mỗi khi người dùng di chuyển vào thư mục sẽ hiển thị ra một thông điệp được lưu trong tập tin chỉ định sẵn. Tập tin này được chỉ ra trong tùy chọn message_file và tên mặc định là .message. Nó được lưu trong thư mục di chuyển vào.
- Message_file: chỉ ra tên của tập tin message, Giá trị mặc định là .message

Truyền tập tin:

- Download_enable: nếu tùy chọn này có giá trị là YES thì download được cho phép, giá trị mặc định là YES
- Chown_uploads: nếu tùy chọn này có giá trị là YES thì tất cả những tập tin được upload bởi người dùng anonymous được sở hữu bởi người dùng được chỉ ra trong chown_username, giá trị mặc định là YES
- chown_username: chỉ ra người sở hữu những tập tin được upload bởi người dùng anonymous, giá trị mặc định là root
- write_enable: Cung cấp quyền ghi cho người dùng



Ngăn chặn host truy xuất vào ftp server: FTP Server kết hợp với tcp_wrappers để thực thi cơ chế giới hạn host truy xuất vào FTP Server:

- Bước 1: Đặt tcp_wrappers=YES trong file vsftpd.conf
- Bước 2: Mô tả file thông tin cấm host <x.y.z.t> trong file /etc/hosts.deny
vsftpd:<host_address>

V. Cấu hình Virtual FTP Server

Tạo thêm 1 Virtual IP address (ví dụ địa chỉ 1.2.3.4), chép tập tin /etc/vsftpd/vsftpd.conf /etc/vsftpd/*.conf và thay đổi các thông tin sau:

- listen=YES
- listen_address=1.2.3.4
- connect_from_port_20=YES
- anonymous_enable=YES
- anon_root=/srv/ftp/knuser
- ftpd_banner=Welcome to FTP at knuser.wiremonkeys.org. Behave!

Chỉnh sửa file /etc/vsftpd/vsftpd.conf và thêm chỉ dẫn listen_address=<địa chỉ IP ban đầu>

Sau đó restart lại dịch vụ VSFTPD bằng lệnh /etc/init.d/vsftpd restart.

V.1. Logging

- dual_log_enable: nếu tùy chọn này có giá trị là YES và cùng với xferlog_enable=YES thì vsftpd sẽ viết 2 tập tin đồng thời là: một log tương thích với wu-ftp được chỉ ra trong xferlog_file và một tập tin log chuẩn vsftpd được chỉ ra trong vsftpd_log_file, giá trị mặc định là NO
- xferlog_enable: nếu tùy chọn này có giá trị là YES thì vsftpd ghi lại những kết nối và thông tin truyền tập tin vào tập tin log được chỉ ra trong tùy chọn vsftpd_log_file, giá trị mặc định là NO
- xferlog_file: chỉ ra tập tin log tương thích với wu-ftp, giá trị mặc định là /var/log/xferlog
- vsftpd_log_file: chỉ ra tập tin log vsftpd, giá trị mặc định là /var/log/vsftpd.log

V.2. Network

Những tùy chọn sau đây phản ảnh cách vsftpd tương tác trong mạng:

- accept_timeout: Chỉ ra lượng thời gian một client sử dụng chế độ passive để thiết lập kết nối, giá trị mặc định là 60
- anon_max_rate: Chỉ ra tốc độ truyền dữ liệu tối đa cho người dùng anonymous. Tính bằng byte/second, giá trị mặc định là 0 (không giới hạn tốc độ truyền)
- connect_timeout: Chỉ ra lượng thời gian một client sử dụng chế độ active để trả lời cho quá trình kết nối dữ liệu. Tính bằng giây, giá trị mặc định là 60
- data_connect_timeout: Chỉ ra khối lượng thời gian truyền dữ liệu tối đa. Tính bằng giây. Khi hết thời gian cho phép kết nối từ client sẽ bị đóng, giá trị mặc định là 300
- max_clients: Chỉ ra số client tối đa có thể đồng thời truy cập đến server



BÀI 14 WEB SERVER

Tóm tắt

Lý thuyết: 5 tiết - Thực hành: 5 tiết.

Mục tiêu	Các mục chính	Bài tập bắt buộc	Bài tập làm thêm
Bài học giới thiệu cơ chế tổ chức cấu hình Web server sử dụng phần mềm Apache	<ul style="list-style-type: none"> I. Giới thiệu về Web Server. II. Giới thiệu Apache. III. Cấu hình Web Server. III. Cấu hình Webhosting. 	Bài tập 4.1 (Dịch vụ Web)	



I. Giới thiệu về Web Server

I.1. Giao thức HTTP

HTTP là một giao thức cho phép trình duyệt Web Browser và servers có thể giao tiếp với nhau. Nó chuẩn hoá các thao tác cơ bản mà một Web Server phải làm được.

HTTP bắt đầu là 1 giao thức đơn giản giống như với các giao thức chuẩn khác trên Internet, thông tin điều khiển được truyền dưới dạng văn bản thô thông qua kết nối TCP. Do đó, kết nối HTTP có thể thay thế bằng cách dùng lệnh "telnet" chuẩn.

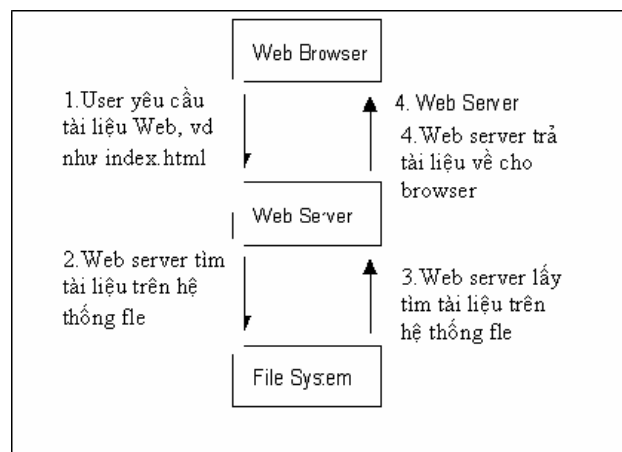
Ví dụ:

```
> telnet www.extropia 80
GET /index.html HTTP/1.0
```

Cổng 80 là cổng mặc định dành cho Web server "lắng nghe" các kết nối được gửi đến. Để đáp ứng lệnh HTTP GET , Web server trả về cho client trang "index.html" thông qua phiên làm việc telnet này, và sau đó đóng kết nối. thông tin trả về dưới dạng code HTML:

```
<HTML>
<HEAD>
<TITLE>eXtropia Homepage</TITLE>
</HEAD>
...
</HTML>
```

Giao thức chỉ thực thi đơn giản hai thao thác yêu-cầu/đáp-ứng (request/response). Một trong các thay đổi lớn nhất trong HTTP/1.1 là nó hỗ trợ kết nối lâu dài (persistent connection).





Trong HTTP/1.0, một kết nối phải được thiết lập đến server cho mỗi đối tượng mà Browser muốn download. Nhiều trang Web có rất nhiều hình ảnh, ngoài việc tải trang HTML cơ bản, browser phải lấy về một số lượng hình ảnh. Nhiều cái trong chúng thường là nhỏ hoặc chỉ đơn thuần là để trang trí cho phần còn lại của trang HTML. Thiết lập một kết nối cho mỗi hình ảnh thật là phí phạm, vì sẽ có nhiều gói thông tin mạng sẽ được luân chuyển giữa Web browser và Web server trước khi dữ liệu ảnh được truyền về. Ngược lại, mở một kết nối TCP truyền tài liệu HTML và sau đó mỗi hình ảnh sẽ truyền nối tiếp theo như thế sẽ thuận tiện hơn và quá trình thiết lập các kết nối TCP sẽ được giảm xuống.

1.2. Web Server và cách hoạt động

Ban đầu Web Server chỉ phục vụ các tài liệu HTML và hình ảnh đơn giản. Tuy nhiên, đến thời điểm hiện tại nó có thể làm nhiều hơn thế. Đầu tiên xét Web server ở mức độ cơ bản, nó chỉ phục vụ các nội dung tĩnh. Nghĩa là khi Web server nhận 1 yêu cầu từ Web browser :<http://www.hcmuns.edu.vn/index.html>, nó sẽ ánh xạ đường dẫn này (Uniform Resource Locator - URL) thành một tập tin cục bộ trên máy Web server. Máy chủ sau đó sẽ nạp tập tin này từ đĩa và đưa nó thông qua mạng đến Web browser của người dùng. Web browser và web server sử dụng giao thức HTTP trong quá trình trao đổi dữ liệu. Các trang tài liệu HTML là một văn bản thô (raw text). Chúng chứa các thẻ định dạng (HTML tag).

Ví dụ:

```
<html>
<head> <title> WWW </title>
</head>
<body>
<p align=center>
<a href="http://www.hcmuns.edu.vn/"><b>Trường Đại Học Khoa Học Tự Nhiên TP.HCM
</b></a>
</p>
</body>
</html>
```

Trên cơ sở phục vụ những trang web tĩnh đơn giản này, ngày nay Web Server đã được phát triển với nhiều thông tin phức tạp hơn được chuyển giữa Web Server và Web Browser, trong đó quan trọng nhất có lẽ là nội dung động (dynamic content). Với phiên bản đầu tiên, Web server hoạt động theo mô hình sau:

- Tiếp nhận các yêu cầu từ browsers.
- Trích nội dung từ đĩa .
- Chạy các chương trình CGI .
- Truyền dữ liệu ngược lại cho client
- Chạy càng nhanh càng tốt.

Điều này sẽ thực hiện tốt đối với các Web sites đơn giản, nhưng server sẽ bắt đầu gặp phải vấn đề khi có nhiều người truy cập hoặc có quá nhiều trang web động phải tốn thời gian để tính toán cho ra kết quả.



Ví dụ:

Nếu một chương trình CGI tốn 30 giây để sinh ra nội dung, trong thời gian này Web server có thể sẽ không phục vụ các trang khác nữa. Do vậy, mặc dù mô hình này hoạt động được, nhưng nó vẫn cần phải thiết kế lại để phục vụ được nhiều người trong cùng 1 lúc. Web server có xu hướng tận dụng ưu điểm của 2 phương pháp khác nhau để giải quyết vấn đề này là: đa tiến trình (multi-threading) hoặc đa tiến trình (multi-processing) hoặc các hệ lai giữa multi-processing và multi-threading.

1.3. Web client

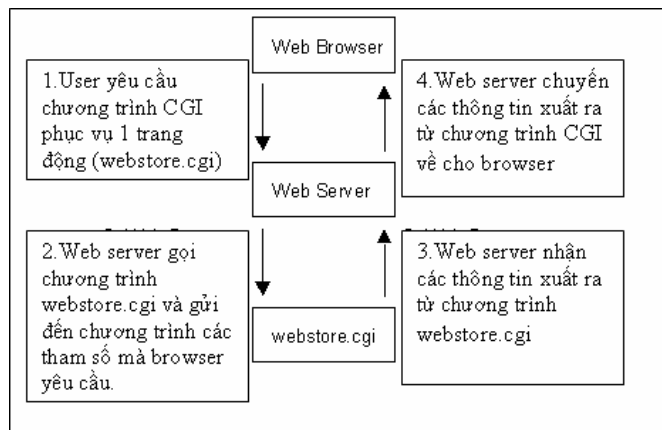
Là những chương trình duyệt Web ở phía người dùng, như Internet Explorer, Netscape Communicator..., để hiển thị những thông tin trang Web cho người dùng. Web client sẽ gửi yêu cầu đến Web Server. Sau đó, đợi Web Server xử lý trả kết quả về cho web client hiển thị cho người dùng. Tất cả mọi yêu cầu đều được xử lý bởi Web Server.

1.4. Web động

Một trong các nội dung động (thường gọi tắt là Web động) cơ bản là các trang Web được tạo ra để đáp ứng các dữ liệu nhập vào của người dùng trực tiếp hay gián tiếp.

Cách cổ điển nhất được dùng phổ biến nhất cho việc tạo nội dung động là sử dụng Common Gateway Interface (CGI). Cụ thể là CGI định nghĩa cách thức Web server chạy một chương trình cục bộ, sau đó nhận kết quả và trả về cho Web browser của người dùng đã gửi yêu cầu.

Web browser thực sự không biết nội dung của thông tin là động, bởi vì CGI về cơ bản là một giao thức mở rộng của Web Server. Hình vẽ sau minh họa khi Web browser yêu cầu một trang Web động phát sinh từ một chương trình CGI.



Một giao thức mở rộng nữa của HTTP là HyperText Transmission Protocol Secure (HTTPS) dùng để bảo mật các thông tin “nhạy cảm” khi chuyển chúng xuyên qua mạng

II. Giới thiệu Apache

Apache là một phần mềm có nhiều tính năng mạnh và linh hoạt dùng để làm Web Server .

- Hỗ trợ đầy đủ những giao thức HTTP trước đây như HTTP/1.1
- Có thể cấu hình và mở rộng với những module của công ty thứ ba



- Cung cấp source code đầy đủ với license không hạn chế.
- Chạy trên nhiều hệ điều hành như Windows NT/9x, Netware 5.x, OS/2 và trên hầu hết các hệ điều hành Unix.

II.1. Cài đặt Apache

Ta chỉ cần cài đặt package `httpd-2.0.40-21.i386.rpm` (trên Fedora) trong hệ điều hành Linux.

```
#rpm -ivh httpd-2.0.40-21.i386.rpm
```

Vị trí cài đặt Apache trong môi trường Linux là `/etc/httpd`. Trong thư mục này lưu giữ những tập tin cấu hình của Apache

II.2. Tạm dừng và khởi động lại Apache

Để tạm dừng hay khởi động lại apache dùng script sau:

```
# chkconfig httpd on
```

```
#/etc/init.d/httpd start/stop/restart
```

Hoặc dùng lệnh :

```
# chkconfig httpd on
```

```
#service httpd restart
```

II.3. Sự chứng thực, cấp phép, điều khiển việc truy cập

Khi nhận một yêu cầu truy cập tài nguyên, web server sẽ xử lý như thế nào để trả kết quả về cho client. Apache có những hướng xử lý khác nhau như chứng thực, cấp phép và điều khiển truy cập.

II.3.1 Basic Authentication

Đối với những thông tin cần bảo mật, khi có yêu cầu truy xuất thông tin này, Web Server phải chứng thực những yêu cầu này có hợp lệ hay không. Thông thường, thông tin chứng thực bao gồm username và password.

- + Nếu một tài nguyên được bảo vệ với sự chứng thực. Apache sẽ gửi một yêu cầu “401 Authentication” thông báo cho người dùng nhập vào username và password của mình. Nhận được yêu cầu này, client sẽ trả lời 401 đến server trong đó có chứa username và password. Server sẽ kiểm tra những thông số này khi nhận được. Nếu hợp lệ server sẽ trả về những thông tin yêu cầu, ngược lại nó sẽ trả về một thông báo lỗi.
- + Bởi vì giao thức HTTP là một tiêu chuẩn không của riêng ai và cũng không thuộc một quốc gia nào, cho nên mỗi yêu cầu đều được xem như nhau.
- + Username và password bạn cung cấp chỉ có tác dụng trong lần giao dịch của browser với server lúc đó. Nếu lần sau truy cập lại website này, bạn phải nhập lại username và password.
- + Song song với trả lời 401, toàn bộ thông tin sẽ trả ngược lại cho client. Trong những trường hợp riêng biệt, server sẽ cấp lại cho client một thẻ chứng thực để bảo vệ website. Thẻ này được gọi là realm hay là một tên chứng thực. Browser sẽ lưu lại



username và password mà bạn đã cung cấp cùng với realm. Như thế, nếu truy cập những tài nguyên khác mà có cùng realm, username và password thì user không cần nhập trở lại những thông tin chứng thực. Thông thường, việc lưu trữ này chỉ có tác dụng trong giao dịch hiện hành của browser. Nhưng cũng có một vài browser cho phép bạn lưu chúng một cách cố định để bạn chẳng bao giờ nhập lại username và password.

Các bước cấu hình chứng thực:

- + Bước 1: tạo tập tin password, cấp quyền truy xuất cho tập tin mật khẩu dùng lệnh **chmod 755 <tập_tin_mật_khẩu_được_tạo_ở_bước_1>**
- + Bước 2: cấu hình apache
- + Bước 3: tạo tập tin group (nếu muốn chứng thực cho nhóm)

Bước 1: Tạo tập tin password dùng lệnh htpasswd, Cách sử dụng lệnh htpasswd theo cú pháp như sau:

```
#htpasswd -c <vị_trí_tập_tin_password> <username>
```

Ví dụ:

```
# htpasswd -c /etc/httpd/conf/passwords rbowen
```

htpasswd sẽ yêu cầu bạn nhập password, và sau đó nhập lại một lần nữa.

```
New password: mypassword
```

```
Re-type new password: mypassword
```

- + Tùy chọn `-c` sẽ tạo một tập tin password mới. Nếu tập tin này đã tồn tại nó sẽ xóa nội dung cũ và ghi vào nội dung mới. Khi tạo thêm một người dùng, tập tin password đã tồn tại bạn không cần dùng tùy chọn `-c`.
- + `<vị_trí_tập_tin_password>`: thông thường nó tạo tại thư mục gốc của apache

Bước 2: Cấu hình sự chứng thực trên Apache:

```
<Directory /upload>
```

```
EnablePut On
```

```
AuthType Basic
```

```
AuthName Temporary
```

```
AuthUserFile /etc/httpd/conf/passwd
```

```
EnableDelete Off
```

```
umask 007
```

```
<Limit PUT>
```

```
require user rbowen sungo
```

```
</Limit>
```

```
</Directory>
```



- + AuthType: khai báo loại authentication sẽ sử dụng. Trong trường hợp này là Basic
- + AuthName: đặt tên cho sự chứng thực
- + AuthUserFile: vị trí của tập tin password
- + AuthGroupFile: vị trí của tập tin group
- + Require: những yêu cầu hợp lệ được cho phép truy cập tài nguyên.

Bước 3: Tạo tập tin group: Nhằm tạo điều kiện thuận lợi cho người quản trị trong việc quản lý sự chứng thực, Apache hỗ trợ thêm tính năng nhóm người dùng. Người quản trị có thể tạo những nhóm người dùng được phép truy cập đến tài nguyên, thêm hay xóa những thành viên trong group ngoài việc chỉnh sửa lại tập tin cấu hình apache và khởi động lại apache. Định dạng của tập tin group :

<tên nhóm> : user1 user2 user3 ... user n

Ví dụ:

authors: rich daniel allan

Sau khi tạo tập tin nhóm, bạn cần cấu hình để apache để chỉ ra tập tin nhóm này bằng những directive sau :

<Directory /upload>

AuthType Basic

AuthName "Apache Admin Guide Authors"

AuthUserFile /etc/httpd/conf/passwords

AuthGroupFile /etc/httpd/conf/groups

Require group authors

</Directory>

II.3.2 Digest Authentication

Digest authentication cung cấp một phương pháp bảo vệ nội dung web một cách luân phiên. Digest authentication được cung cấp bởi module mod_auth_digest. Với phương pháp này tên user và mật khẩu sẽ không được gửi ở dạng plain text mà chúng được mã hóa (thông qua thuật toán MD5)

Cấu hình: Tương tự như sự chứng thực cơ bản, cấu hình này cũng gồm 2 hoặc 3 bước sau:

- Bước 1: Tạo file mật khẩu.
- Bước 2: Cấu hình /etc/httpd/conf/httpd.conf để sử dụng file mật khẩu ở bước 1.
- Bước 3: Tạo group file.

Bước 1: Tạo tập tin password dùng lệnh htdigest -c <vi_ trí_tập_tin_password> realm <username>

Bước 2: Cấu hình /etc/httpd/conf/httpd.conf để sử dụng file mật khẩu

<Directory /upload>

AuthType Digest



```
AuthName "Private"
AuthDigestFile /usr/local/apache/passwd/digest
AuthDigestGroupFile /usr/local/apache/passwd/digest.groups
Require group admins
</Directory>
```

Bước 3: Tạo tập tin nhóm(bước này chỉ thực hiện khi ta muốn chứng thực cho nhóm), Cấu trúc của tập tin nhóm cũng tương tự như tập tin nhóm của basic authentication.

admins: joy danne sue

II.4. Điều khiển truy cập

Ngoài việc bảo mật nội dung của website bằng sự chứng thực (username và password), apache còn giới hạn việc truy cập của người dùng dựa trên những thông tin khác được đề cập trong Access Control. Sử dụng directive Allow/Deny để cho phép/cấm việc truy cập tài nguyên dựa trên tên máy tính hoặc địa chỉ IP.

Allow/Deny Directive:

Cú pháp khai báo Allow/Deny như sau

Allow/Deny from [address]

- + Allow có nghĩa cho phép các host/network/domain truy xuất vào Web server.
- + Deny có nghĩa cấm các host/network/domain truy xuất vào Web server.
- + address là địa chỉ IP/địa chỉ đường mạng hay tên máy tính, tên miền.

Ví dụ:

```
Deny from 11.22.33.44
Deny from host.example.com
Deny from 192.101.205
Deny from exampleone.com example
```

Bạn sử dụng Order để kết hợp giữa Allow và Deny trong việc giới hạn việc truy cập. Nếu thứ tự của Order là Deny, Allow thì Deny được kiểm tra trước tiên và bất kỳ những client nào không phù hợp với Deny hoặc phù hợp với một Allow thì được phép truy cập đến server. Ngược lại, nếu thứ tự của Order là Allow, Deny thì Allow được kiểm tra trước và bất kỳ client nào không phù hợp với một điều kiện Allow hoặc phù hợp với một điều kiện Deny thì bị cấm truy cập đến server.

Ví dụ về một điều khiển truy cập ít giới hạn nhất.

```
<Directory "/usr/web">
Options Indexes FollowSymLinks MultiViews
AllowOverride All
Order allow,deny
Allow from all
</Directory>
```



Satisfy directive:

- Satisfy directive được dùng để chỉ ra nhiều tiêu chuẩn cần phải được xem xét trong việc bảo mật nội dung website. Satisfy có 2 giá trị là all và any. Mặc định, Satisfy nhận giá trị all, điều này có nghĩa là nếu nhiều tiêu chuẩn được chỉ ra thì tất cả những tiêu chuẩn này phải thoả mãn thì người dùng mới được phép truy cập tài nguyên. Còn giá trị any có nghĩa là một trong những tiêu chuẩn này hợp lệ thì user được phép truy cập đến tài nguyên.
- Một ứng dụng của việc sử dụng access control là giới hạn, những người dùng bên ngoài mạng khi truy cập tài nguyên cần phải có username và password còn tất cả những máy tính trong mạng thì không cần.

```
<Directory /usr/local/apache/htdocs/sekrit>
```

```
AuthType Basic
AuthName intranet
AuthUserFile /etc/httpd/conf/users
AuthGroupFile /etc/httpd/conf/groups
Require group customers
Allow from internal.com
Satisfy any
```

```
</Directory>
```

II.5. Khảo sát log file trên apache

Apache có nhiều tập tin log khác nhau nhằm ghi lại những hoạt động của Web Server. Sau đây mô tả tính năng của từng tập tin.

File error_log:

Là một tập tin log quan trọng nhất. Tên và vị trí của nó được xét trong ErrorLog directive. ErrorLog là nơi mà httpd sẽ gửi những thông tin nhận dạng và bất kỳ những lỗi nào gặp phải trong quá trình xử lý những yêu cầu. Tập tin này chính là nơi mà ta cần xem xét đầu tiên khi gặp phải những lỗi khởi động httpd hay những thao tác của server, vì nó lưu những thông tin chi tiết về những lỗi và cách sửa lỗi. Định dạng của tập tin error_log không bị bó buộc. Nội dung của file error_log như sau:

```
[Wed Oct 11 14:32:52 2000] [error] [client 127.0.0.1] client denied by server configuration:
/export/home/live/ap/htdocs/test
```

Cột đầu tiên chỉ ra ngày giờ entry này được tạo ra. Cột thứ 2 chỉ ra đây là entry lỗi. Cột thứ 3 cho biết địa chỉ IP của client tạo ra lỗi. Tiếp theo là message có nội dung chỉ ra rằng server được cấu hình để từ chối việc truy cập của client. Tiếp theo là đường dẫn của document mà client cần truy cập. Trong quá trình kiểm tra, bạn có thể theo dõi error_log một cách liên tục bằng dòng lệnh sau:

```
#tail -f /var/log/httpd/error_log
```

File access_log:



Access_log là nơi ghi lại tất cả những yêu cầu được xử lý bởi server. Vị trí và nội dung của access log được điều khiển bởi CustomLog directive. Bạn có thể dùng LogFormat directive trong việc định dạng nội dung của tập tin access_log. LogFormat chỉ ra những thông tin mà server cần theo dõi để ghi lại trong access log. Để theo dõi yêu cầu xử lý trên Web Server ta dùng lệnh:

```
#tail -f /var/log/httpd/access_log
```

Luân chuyển log file:

Theo thời gian, thông tin lưu trong các tập tin log lớn làm cho kích thước của các tập tin này có thể vượt quá 1MB. Thật là cần thiết khi bạn xóa hoặc di chuyển hay sao lưu những tập tin log này một cách luân phiên và có chu kỳ. Ta có thể thực hiện như sau :

```
mv          access_log  access_log.old
mv          error_log   error_log.old
apachectl  graceful
sleep 600
gzip access_log.old error_log.old
```

III. Cấu hình Web Server

Các tập tin và thư mục cấu hình của Apache :

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

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

III.1. Định nghĩa về ServerName

III.1.1 Chỉ định một số thông tin cơ bản

Cấu hình tên máy tính (hostname) của server. Nó được dùng trong việc tạo ra những URL chuyển tiếp (redirection URL). Nếu không chỉ ra, server sẽ cố gắng suy luận từ địa chỉ IP của nó. Tuy nhiên, điều này có thể không tin cậy hoặc không trả ra tên máy tính đúng. Cú pháp khai báo:

```
ServerName <hostname>
```

Ví dụ:

```
ServerName www.soft.com
```

ServerAdmin:

Địa chỉ Email của người quản trị hệ thống

Cú pháp :

```
ServerAdmin <địa chỉ email>
```



Ví dụ:

ServerAdmin root@soft.com

ServerType:

Qui định cách nạp chương trình. Có hai cách :

- + inetd: chạy từ hệ thống.
- + standalone : chạy từ các init level.

Cú pháp :

ServerType <inetd/standalone>

Ví dụ:

ServerType standalone

III.2. Thư mục Webroot và một số thông tin cần thiết

Chỉ định DocumentRoot: Cấu hình thư mục gốc lưu trữ nội dung của Website. Web Server sẽ lấy những tập tin trong thư mục này phục vụ cho yêu cầu của client.

Cú pháp :

DocumentRoot <đường_dẫn_thư_mục>

Ví dụ:

DocumentRoot /usr/web

Một yêu cầu <http://www.soft.com> sẽ được đưa vào trang web /usr/web/index.html

ServerRoot: Vị trí cài đặt web server.

Cú pháp:

ServerRoot <vị_trí_thư_mục_cài_đặt_apache>

Mặc định:

ServerRoot /usr/local/apache (trong Linux là /etc/httpd)

Error log: Chỉ ra tập tin để server ghi vào bất kỳ những lỗi nào mà nó gặp phải

Cú pháp:

ErrorLog <vị_trí_tập_tin_log>

Ví dụ:

ErrorLog logs/error_log

Nếu đường dẫn vị trí không có dấu / thì vị trí tập tin log liên quan đến ServerRoot.

DirectoryIndex: Các tập tin mặc định khi truy cập tên web site.

Cú pháp:

DirectoryIndex <danh_sách_các_tập_tin>

Ví dụ:

DirectoryIndex index.html index.htm index.shtml index.php index.php4 index.php3 index.cgi



III.3. Cấu hình mạng

MaxClients: Qui định số yêu cầu tối đa từ các client có thể gọi đồng thời đến server .

Cú pháp:

```
MaxClients <number>
```

Ví dụ:

```
MaxClients 256
```

Listen: Qui định địa chỉ IP hoặc Cổng mà Apache nhận kết nối từ Client.

Cú pháp:

```
Listen <Port/IP>
```

Ví dụ:

```
Listen 80
```

BindAddress: Qui định địa chỉ card mạng để chạy Apache trên Server.

Cú pháp:

```
BindAddress <IP/*>
```

Sử dụng dấu "*" để có thể sử dụng tất cả các địa chỉ có trên máy.

Ví dụ:

```
BindAddress 172.29.7.225
```

Mặc định là : BindAddress *

TimeOut: quy định thời gian sống của một kết nối (được tính bằng giây).

Cú pháp:

```
TimeOut <time>
```

Ví dụ:

```
TimeOut 300
```

KeepAlive: cho phép hoặc không cho phép Client gửi được nhiều yêu cầu dựa trên một kết nối với Web Server.

Cú pháp:

```
KeepAlive <On/Off>
```

Ví dụ:

```
KeepAlive On
```

MaxKeepAliveRequests: số Request tối đa trên một kết nối (nếu cho phép nhiều Request trên một kết nối).

Cú pháp:

```
MaxKeepAliveRequests <số Request>
```

Ví dụ:

```
MaxKeepAliveRequests 100
```



KeepAliveTimeout: qui định thời gian để chờ cho một Request kế tiếp từ cùng một Client trên cùng một kết nối (được tính bằng giây).

Cú pháp:

```
KeepAliveTimeout <time>
```

Ví dụ:

```
KeepAliveTimeout 15
```

III.4. Alias

Cung cấp cơ chế ánh xạ đường dẫn cục bộ (không nằm trong DocumentRoot) thành đường dẫn địa chỉ URL.

Cú pháp:

```
Alias <đường_dẫn_http> <đường_dẫn_cục_bộ>
```

Ví dụ:

```
Alias /doc /usr/share/doc
```

Khi truy cập `http://www.soft.com/doc` sẽ đưa vào `/usr/share/doc`

Để giới hạn việc truy cập của người dùng ta có thể kết hợp với Directory directive.

Ví dụ:

```
Alias /doc /usr/share/doc
<Directory /usr/share/doc>
    AuthType Basic
    AuthName intranet
    AuthUserFile /etc/httpd/passwd
    Require user hally tom
    Allow from internal.com
```

```
</Directory>
```

III.5. UserDir

Cho phép người dùng tạo Home page của user trên WebServer

Cấu hình:

```
<IfModule mod_userdir.c>
```

```
    #UserDir disable
```

```
UserDir www ; thư mục Web của user.
```

```
</IfModule>
```

```
<Directory /home/*/www>
```

```
...
```

```
</Directory>
```




Trong thư mục Home Directory của người dùng tạo thư mục www. Ví dụ /home/nva/www. Khi đó cú pháp truy cập từ Web Browser có dạng: http://www.soft.com/~<tênUser>. Ví dụ: http://www.soft.com/~nva.

Khi người dùng cố gắng truy cập đến thư mục của mình có thể gặp một message lỗi “Forbidden” . Điều này có thể là quyền truy cập đến home directory của người dùng bị giới hạn. Bạn có thể giới hạn lại quyền truy cập home directory của người dùng với những câu lệnh như sau:

```
chown jack /home/jack /home/jack/www
chmod 750 /home/jack /home/jack/www
```

III.6. VirtualHost

Là tính năng của Apache giúp ta duy trì nhiều hơn một web server trên một máy tính. Nhiều tên cùng chia sẻ một địa chỉ IP gọi là named-based virtual hosting, và sử dụng những địa chỉ IP khác nhau cho từng domain gọi là IP-based virtual hosting.

III.6.1 IP-based Virtual Host

VirtualHost dựa trên IP yêu cầu những server phải có một địa chỉ IP khác nhau cho mỗi virtualhost dựa trên IP. Như vậy, một máy tính phải có nhiều interface hay sử dụng cơ chế virtual interface mà những hệ điều hành sau này hỗ trợ. Nếu máy của bạn có một địa chỉ IP, 97.158.253.26, bạn có thể cấu hình một địa chỉ IP khác trên cùng một card mạng như sau:

```
ifconfig eth0:1 97.158.253.27 netmask 255.255.255.0 up
```

Sau đó ta mô tả thông tin cấu hình trong file httpd.conf

```
<VirtualHost *>          ; VirtualHost default
    ...
    DocumentRoot /tmp
        ServerName www.domain
    ...
</VirtualHost>
<VirtualHost 97.158.253.26> ; VirtualHost cho site1
    ...
    DocumentRoot /home/www/site1
        ServerName www1.domain
    ...
</VirtualHost>
<VirtualHost 97.158.253.27>; VirtualHost cho site2
    ...
    DocumentRoot /home/www/site2
        ServerName www2.domain
```



...

</VirtualHost>

III.6.2 Named-based Virtual Hosts:

IP-based Virtual Host dựa vào địa chỉ IP để quyết định **Virtual Host** nào đúng để truy cập. Vì thế, bạn cần phải có địa chỉ IP khác nhau cho mỗi **Virtual Host**. Với Named-based **Virtual Host**, server dựa vào HTTP header của client để biết được hostname. Sử dụng kỹ thuật này, một địa chỉ IP có thể có nhiều tên máy tính khác nhau. Named-based **Virtual Host** rất đơn giản, bạn chỉ cần cấu hình DNS sao cho nó phân giải mỗi tên máy đúng với một địa chỉ IP và sau đó cấu hình Apache để tổ chức những web server cho những miền khác nhau.

Cấu hình: Tham khảo đoạn cấu hình VirtualHost cho www.hcm.vn và www.tatavietnam.hcm.vn, www.ntc.hcm.vn sử dụng chung một IP 172.29.14.150

```
NameVirtualHost 172.29.14.150
#Virtualhost mặc định
<VirtualHost *>
    ServerAdmin webmaster@dummy-host.example.com
    DocumentRoot /tmp
    RewriteEngine    on
    RewriteLogLevel  0
    ServerName dummy-host.example.com
    ErrorLog logs/dummy-host.example.com-error_log
    CustomLog logs/dummy-host.example.com-access_log common
</VirtualHost>
<VirtualHost 172.29.14.150>#Virtualhost cho WebServer chính
    ServerAdmin webmaster@dummy-host.example.com
    RewriteEngine    on
    RewriteLogLevel  0
    DocumentRoot /var/www/html
    ServerName www.hcm.vn
    ErrorLog logs/dummy-host.example.com-error_log
    CustomLog logs/dummy-host.example.com-access_log common
</VirtualHost>
<VirtualHost 172.29.14.150>#virtualhost cho host Web Server
tatavietnam
    ServerAdmin webmaster@dummy-host.example.com
```



```
DocumentRoot /webdata
RewriteEngine    on
RewriteLogLevel  0
ServerName www.tatavietnam.hcm.vn
ErrorLog logs/dummy-host.example.com-error_log
CustomLog logs/dummy-host.example.com-access_log common
</VirtualHost>
<VirtualHost 172.29.14.150>#virtualhost cho host Web Server ntc
ServerAdmin webmaster@dummy-host.example.com
DocumentRoot /ntc
RewriteEngine    on
RewriteLogLevel  0
ServerName www.ntc.hcm.vn
ErrorLog logs/dummy-host.example.com-error_log
CustomLog logs/dummy-host.example.com-access_log common
</VirtualHost>
```



BÀI 15 MAIL SERVER

Tóm tắt

Lý thuyết: 8 tiết - Thực hành: 10 tiết.

Mục tiêu	Các mục chính	Bài tập bắt buộc	Bài tập làm thêm
Bài học giới thiệu cơ chế tổ chức và quản trị hệ thống Mail. Cung cấp cho người dùng hệ thống có thể sử dụng E-mail thông qua Mail POP Client và Webmail.	<ol style="list-style-type: none"> I. Những giao thức mail II. Giới thiệu về hệ thống mail III. Những chương trình mail và một số khái niệm IV. DNS và Sendmail V. Những tập tin cấu hình Sendmail VI. Tập tin /etc/aliases VII. Cấu hình Mail Server với Sendmail VIII. Một số file cấu hình trong sendmail IX. Cấu hình POP Mail Server X. Cài đặt và cấu hình Webmail - Openwebmail. 	Bài tập 5.1 (Dịch vụ Mail)	



I. Những giao thức mail

Hệ thống mail được xây dựng dựa trên một số giao thức sau: Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP), Multipurpose Internet Mail Extensions (MIME) và Interactive Mail Access Protocol (IMAP), được định trong RFC 1176 là một giao thức quan trọng được thiết kế để thay thế POP, nó cung cấp nhiều cơ chế tìm kiếm văn bản, phân tích message từ xa mà ta không tìm thấy trong POP. Trong phần này ta chỉ qua tâm tới ba giao thức SMTP, POP, MIME trong hệ thống mail.

I.1. SMTP(Simple Mail Transfer Protocol)

SMTP là giao thức tin cậy chịu trách nhiệm phân phát mail. Nó chuyển mail từ hệ thống mạng này sang hệ thống mạng khác, chuyển mail trong hệ thống mạng nội bộ. Giao thức SMTP được định nghĩa trong RFC 821, SMTP là một dịch vụ tin cậy, hướng kết nối(connection-oriented) được cung cấp bởi giao thức TCP(Transmission Control Protocol), nó sử dụng số hiệu cổng (well-known port) 25. Sau đây là danh sách các tập lệnh trong giao thức SMTP.

Tập lệnh SMTP		
Lệnh	Cú pháp	chức năng
Hello	HELO <sending-host>	Lệnh nhận diện SMTP
From	MAIL FROM:<from-address>	Địa chỉ người gửi
Recipient	RCPT TO:<to-address>	Địa chỉ người nhận
Data	DATA	Bắt đầu gửi thông điệp
Reset	RSET	Hủy bỏ thông điệp
Verify	VERFY <string>	Kiểm tra username
Expand	EXPN <string>	Mở rộng danh sách mail
Help	HELP [string]	Yêu cầu giúp đỡ
Quit	QUIT	Kết thúc phiên giao dịch SMTP



Để sử dụng các lệnh SMTP ta dùng lệnh telnet theo port 25 trên hệ thống ở xa sau đó gửi mail thông qua cơ chế dòng lệnh. Kỹ thuật này thỉnh thoảng cũng được sử dụng để kiểm tra hệ thống SMTP server, nhưng điều chính yếu ở đây là chúng ta sử dụng SMTP để minh họa làm cách nào mail được gửi qua các hệ thống khác nhau. Trong ví dụ sau minh họa quá trình gửi mail thông qua cơ chế dòng lệnh SMTP của Daniel trên máy peanut.nuts.com tới almond.nuts.com của Tyler.

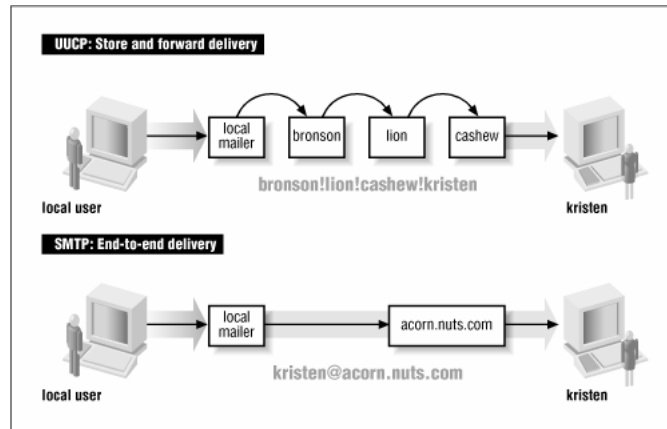
```
%telnet almond.nuts.com 25
Trying 172.16.12.1 ...
Connected to almond.nuts.com.
Escape character is '^]'
220 almond Sendmail 4.1/1.41 ready at Tue, 29 Mar 94 17:21:26
EST
helo peanut.nuts.com
250 almond Hello peanut.nuts.com, pleased to meet you
//địa chỉ người gửi
mail from:<daniel@peanut.nuts.com>
250 <daniel@peanut.nuts.com>... Sender ok
//địa chỉ người nhận
rcpt to:<tyler@almond.nuts.com>
250 <tyler@almond.nuts.com>... Recipient ok
//bắt đầu viết nội dung thư.
data
354 Enter mail, end with "." on a line by itself
Hi Tyler!
.
250 Mail accepted
//thoát ra khỏi phiên giao dịch
quit
221 almond delivering mail
Connection closed by foreign host.
```

Ngoài ra còn có một số lệnh khác như: SEND, SOML, SAML, và TURN được định trong RFC 821 là những câu lệnh tùy chọn và không được sử dụng thường xuyên.

Lệnh HELP in ra tóm tắt các lệnh được thực thi. Ví dụ ta dùng lệnh HELP RSET chỉ định các thông tin được yêu cầu khi sử dụng lệnh RSET, Lệnh VRFY và EXPN thì hữu dụng hơn nhưng nó thường bị khoá vì lý do an ninh mạng bởi vì nó cung cấp cho người dùng chi tiết về mạng. Ví dụ lệnh EXPN <admin> yêu cầu liệt kê ra danh sách địa chỉ email nằm trong nhóm mail admin. Lệnh VRFY để lấy các thông tin cá nhân của một tài khoản nào đó, ví dụ lệnh VRFY <mac>, mac là một tài khoản cục bộ. Trường hợp ta dùng lệnh VRFY <jane>, jane là một bí danh nằm trong file /etc/aliases thì giá trị trả về là địa chỉ email được tìm thấy trong file aliases này.

SMTP là hệ thống phân phát mail trực tiếp từ đầu đến cuối (từ nơi bắt đầu phân phát cho đến trạm phân phát cuối cùng), điều này rất hiếm khi sử dụng. hầu hết hệ thống mail sử dụng giao thức store and forward như UUCP và X.400, hai giao thức này di chuyển mail đi qua mỗi hop, nó lưu trữ thông điệp tại mỗi hop và sau đó chuyển tới hệ thống tiếp theo, thông điệp được chuyển tiếp cho tới khi nó tới hệ thống phân phát cuối cùng.

Trong hình sau minh họa cả hai kỹ thuật store and forward và phân phát trực tiếp tới hệ thống mail. Địa chỉ UUCP chỉ định đường đi mà mail đi qua để tới người nhận, trong khi đó địa chỉ mail SMTP ngụ ý là hệ thống phân phát cuối cùng.



Phân phát trực tiếp(Direct delivery) cho phép SMTP phân phát E-mail mà không dựa vào host trung gian nào. Nếu như SMTP phân phát bị lỗi thì hệ thống cục bộ sẽ thông báo cho người gửi hay nó đưa mail vào hàng đợi mail để phân phát sau. Bất lợi của việc phân phát trực tiếp(direct delivery) là nó yêu cầu hai hệ thống cung cấp đầu đủ các thông tin điều khiển mail, một số hệ thống không thể điều khiển mail như PCs các hệ thống mobile như laptops, những hệ thống này thường tắt máy vào cuối ngày hay thường xuyên không trực tuyến(offline). Để điều khiển những trường hợp này cần phải có hệ thống DNS được sử dụng để chuyển thông điệp tới máy chủ mail thay cho hệ thống phân phát mail trực tiếp. Mail sau đó được chuyển từ server tới máy trạm khi máy trạm kết nối mạng trở lại(online), giao thức mạng POP cho phép thực hiện chức năng này.

I.2. Post Office Protocol

Có hai phiên bản của POP được sử dụng rộng rãi là POP2, POP3. POP2 được định nghĩa trong RFC 937, POP3 được định nghĩa trong RFC 1725. POP2 sử dụng 109 và POP3 sử dụng Port 110. Các câu lệnh trong hai giao thức này không giống nhau nhưng chúng cùng thực hiện chức năng cơ bản là kiểm tra tên đăng nhập và password của user và chuyển mail của người dùng từ server tới hệ thống đọc mail cục bộ của user. Trong khi đó tập lệnh của POP3 hoàn toàn khác với tập lệnh của POP2.

Table 3.3: POP3 Commands

Lệnh	Chức năng
USER username	Cho biết thông tin về username cần nhận mail
PASS password	Password của username cần nhận mail
STAT	Hiển thị số thông điệp chưa được đọc tính bằng bytes
RETR n	Nhận thông điệp thứ n
DELE n	Xoá thông điệp thứ n



Table 3.3: POP3 Commands

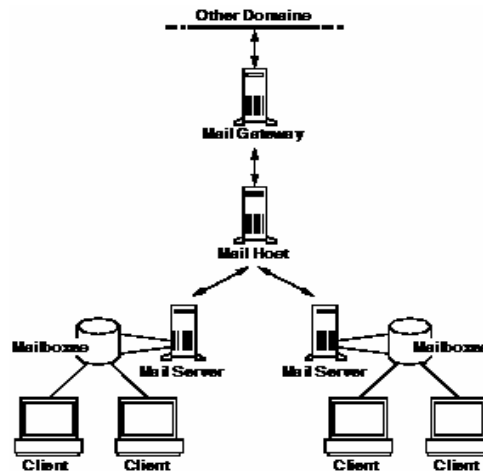
Lệnh	Chức năng
LAST	Hiển thị thông tin message cuối cùng.
LIST [n]	Hiển thị kích thước của thông điệp thứ n
RSET	Không xoá tất cả thông điệp, và quay lại thông điệp đầu tiên
TOP n l	In ra các HEADER và dòng thứ n của thông điệp
NOOP	Không làm gì
QUIT	Kết thúc phiên giao dịch POP3

Mặc dù các câu lệnh của POP3 và POP2 khác nhau như chúng cùng thực hiện một chức năng, sau đây là ví dụ về phiên giao dịch POP3 :

```
% telnet almond 110
Trying 172.16.12.1 ...
Connected to almond.nuts.com.
Escape character is '^]'.
+OK almond POP3 Server Process 3.3(1) at Mon 15-May-95 4:48PM-EDT
user hunt
+OK User name (hunt) ok. Password, please.
pass Watts?Watt?
+OK 3 messages in folder NEWMAIL (V3.3 Rev B04)
stat
+OK 3 459
retr 1
+OK 146 octets
The full text of message 1
dele 1
+OK message # 1 deleted
retr 2
+OK 155 octets
The full text of message 2
dele 2
+OK message # 2 deleted
retr 3
+OK 158 octets
The full text of message 3
dele 3
+OK message # 3 deleted
quit
+OK POP3 almond Server exiting (0 NEWMAIL messages left)
Connection closed by foreign host.
```


II. Giới thiệu về hệ thống mail

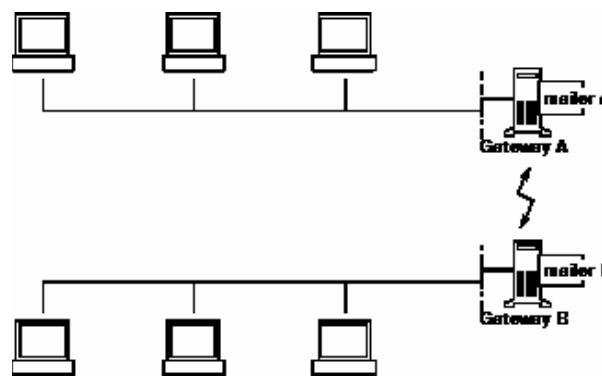
Những thành phần trong một hệ thống mail Một hệ thống mail yêu cầu phải có ít nhất hai thành phần, nó có thể định vị trên hai hệ thống khác nhau hoặc trên cùng một hệ thống, mail server và mail client. Ngoài ra, nó còn có những thành phần khác như Mail Host, Mail Gateway. Sơ đồ về một hệ thống email đầy đủ các thành phần:



II.1. Mail gateway

Một mail gateway là máy kết nối giữa các mạng dùng các giao thức truyền thông khác nhau hoặc kết nối các mạng khác nhau dùng chung giao thức. Ví dụ một mail gateway có thể kết nối một mạng TCP/IP với một mạng chạy bộ giao thức Systems Network Architecture (SNA).

Một mail gateway đơn giản nhất dùng để kết nối 2 mạng dùng chung giao thức hoặc mailer. Khi đó mail gateway chuyển mail giữa domain nội bộ và các domain bên ngoài. Mail gateway cũng kết nối 2 mạng dùng mailer khác nhau như hình vẽ dưới. Gateway giữa 2 giao thức truyền khác nhau:



II.2. Mail Host

Một mail host là máy giữ vai trò máy chủ mail chính trong hệ thống mạng. Nó dùng như thành phần trung gian để chuyển mail giữa các vị trí không kết nối trực tiếp được với nhau.

Mail host phân giải địa chỉ người nhận để chuyển giữa các mail server hoặc chuyển đến mail gateway.

Một ví dụ về mail host là máy trong mạng cục bộ LAN có modem được thiết lập liên kết PPP hoặc UUCP dung phone line . Mail host cũng có thể là máy chủ đóng vai trò router giữa mạng nội bộ và mạng Internet.

II.3. Mail Server

Mail Server chứa mailbox của người dùng. Mail Server nhận mail từ mail client gửi đến và đưa vào hàng đợi để gửi đến Mail Host. Mail Server nhận mail từ Mail Host gửi đến và đưa vào mailbox của người dùng. Người dùng sử dụng NFS (Network File System) để mount thư mục chứa mailbox trên Mail Server để đọc. Nếu NFS không được hỗ trợ thì người dùng phải login vào Mail Server để nhận thư.

Trong trường hợp Mail Client hỗ trợ POP/IMAP và trên Mail Server cũng hỗ trợ POP/IMAP thì người dùng có thể đọc thư bằng POP/IMAP.

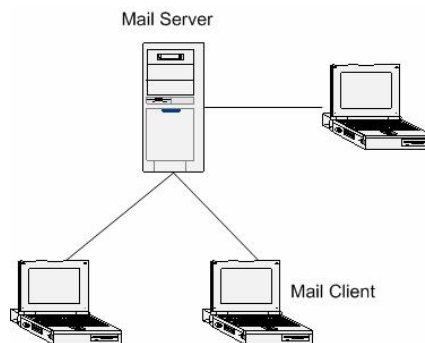
II.4. Mail Client

Là những hệ thống mà nó cho phép tập tin mail spool của user được đọc thông qua cơ chế mount của NFS thư mục /var/mail từ mail hub, nếu không có thư mục /var/mail thì ta phải mount tự động thư mục /var/mail trong tập tin vfstab từ server.

II.5. Một số sơ đồ hệ thống mail thường dùng

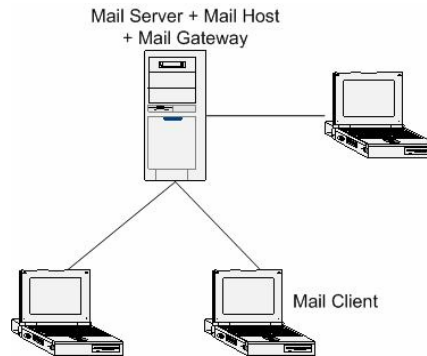
II.5.1 Hệ thống mail cục bộ

Cấu hình hệ thống mail đơn giản gồm một hoặc nhiều trạm làm việc kết nối vào một Mail Server. Tất cả mail đều chuyển cục bộ.



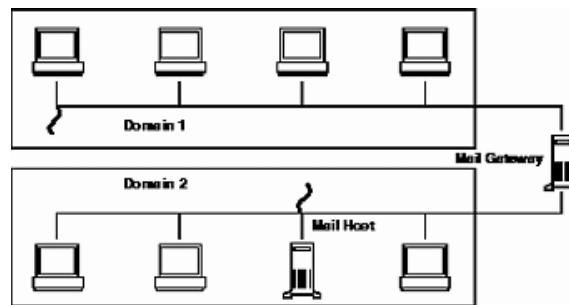
II.5.2 Hệ thống mail cục bộ có kết nối từ xa:

Hệ thống mail trong một mạng nhỏ gồm một mail server, một mail host và một mail gateway kết nối với hệ thống bên ngoài. Không cần DNS server



II.5.3 Hệ thống hai domain và một gateway

Cấu hình dưới đây gồm 2 domain và một mail gateway. Trong cấu hình này mail server, mail host, và mail gateway (hoặc gateways) cho mỗi domain hoạt động như một hệ thống độc lập. Để quản trị và phân phối mail cho 2 domain thì dịch vụ DNS buộc phải có.



III. Những chương trình mail và một số khái niệm

III.1. Mail User Agent (MUA)

MUA : là những chương trình mà người sử dụng dùng để đọc, soạn thảo và gửi mail.

III.2. Mail Transfer Agent (MTA)

MTA : là chương trình chuyển thư giữa các máy Mail Hub. Sendmail là một Mail Transfer Agent (MTA) dùng giao thức SMTP để đóng vai trò là một SMTP Server làm nhiệm vụ định tuyến trong việc phân thư. Nó nhận mail từ những Mail User Agent (MUA) và những MTA khác, sau đó chuyển mail đến các MTA trên máy khác hay MTA trên máy của mình. Để nó không đóng vai trò là một trạm phân thư đến cho người dùng, ta phải dùng một chương trình khác như POP, IMAP để thực hiện việc này.

III.3. Mailbox

Mailbox là một tập tin lưu trữ tất cả các mail của người dùng. Trên hệ thống Unix, khi ta thêm một tài khoản người dùng vào hệ thống đồng thời sẽ tạo ra một mailbox cho người dùng đó. Thông thường, tên của mailbox trùng với tên đăng nhập của người dùng. Tập tin này đặt trong thư mục /var/spool/mail. Khi có mail gửi đến cho người dùng, chương trình xử lý mail của server cục bộ sẽ phân phối mail này vào mailbox tương ứng. Trong tập tin mailbox, mỗi mail bắt đầu bằng dòng có từ khoá From và kết thúc bằng một dòng trắng.



Khi người dùng đăng nhập vào hệ thống và sử dụng mail client để nhận mail (hoặc telnet trực tiếp vào mailserver để nhận), POP Server sẽ vào thư mục `/var/spool/mail` lấy mail từ mailbox chuyển cho người dùng.

Thông thường, sau khi client nhận mail, các mail trong mail box sẽ bị xóa. Tuy nhiên, người dùng cũng có thể yêu cầu giữ lại mail trên mailbox, điều này thực hiện nhờ vào một tùy chọn của mail client.

III.4. Hàng đợi (queue)

Các mail gửi đi có thể được chuyển đi ngay hoặc cũng có thể được chuyển vào hàng đợi. Có nhiều nguyên nhân khiến một mail bị giữ lại trong hàng đợi :

- Khi mail đó tạm thời chưa thể chuyển đi được hoặc có một số địa chỉ trong danh sách người nhận chưa thể chuyển đến được vào thời điểm hiện tại.
- Khi tùy chọn cấu hình phân phát mail có giá trị là True, khi đó tất cả các mail đều bị giữ lại cho đến khi việc phân phối hoàn tất.
- Khi giá trị DeliverMode(d)bằng queue-only hoặc defer thì tất cả các mail đều bị giữ lại trong hàng đợi.
- Khi số lượng tiến trình phân phối bị tắc nghẽn vượt quá giới hạn quy định bởi tùy chọn QueueLA(x).

III.5. Alias

Một số vấn đề phức tạp thường gặp trong quá trình phân thư là :

- Phân phối đến cho cùng một người qua nhiều địa chỉ khác nhau.
- Phân phối đến nhiều người nhưng qua cùng một địa chỉ.
- Kết nối thư với một tập tin để lưu trữ hoặc dùng cho các mục đích khác nhau.
- Lọc thư thông qua các chương trình hay các script.

Để giải quyết các vấn đề trên ta phải sử dụng alias. Đó là sự thay thế một địa chỉ người nhận bằng một hay nhiều địa chỉ khác. Địa chỉ dùng thay thế có thể là một người nhận, một danh sách người nhận, một chương trình, một tập tin hay là sự kết hợp của những loại này.

Các thông tin về alias lưu trong tập tin aliases. Tập tin này được sendmail xác định qua 2 tùy chọn trong tập tin cấu hình là ServiceSwitchFile và AliasFile. Tùy chọn thứ nhất chỉ ra phương thức tìm kiếm các alias(chẳng hạn tìm kiếm trong các tập tin), tùy chọn thứ hai chỉ ra tập tin aliases sẽ được sử dụng.

III.5.1 Tập tin aliases

Cấu trúc của tập tin này là các dòng text. Các dòng trống, các dòng chú thích sẽ bị bỏ qua khi sendmail sử dụng tập tin này. Các dòng bắt đầu với một khoảng trắng hoặc một khoảng tab được xem là tiếp tục của dòng trên nó. Tất cả các dòng khác là các dòng mô tả các alias. Mỗi dòng alias có dạng như sau :

Alias: local



Phần local đặt ở đầu dòng là một địa chỉ người dùng cục bộ, tiếp theo đó là dấu hai chấm (có thể có các khoảng trắng ở giữa). Nếu không có dấu hai chấm thì dòng đó xem như không hợp lệ . Sau dấu hai chấm là phần alias, đó có thể là một hoặc nhiều địa chỉ cách nhau bởi dấu phẩy, giữa các địa chỉ có thể có khoảng trắng. Địa chỉ có thể hiểu là địa chỉ email, tên một chương trình xử lý mail, tên tập tin để gắn mail vào hoặc tên của một tập tin chứa các địa chỉ khác.

Phần local phải là một user cục bộ. Khi sendmail đọc một tên local, nó sẽ thực hiện các bước chuẩn hóa và thẩm định tên đó. Việc chuẩn hóa địa chỉ thực hiện bằng cách tách lấy phần địa chỉ, chuyển thành ký tự thường rồi viết lại theo rule set 3 và 0 để kiểm tra xem, với địa chỉ đó thì có thể tìm được trạm phân phối cục bộ nào không .

Ví dụ: Một dòng alias có nội dung như sau :

```
geogre : gw
```

Sau khi được chuẩn hóa và kiểm tra thấy hợp lệ , sendmail sẽ lưu lại trong cơ sở dữ liệu của nó thông tin như sau :

```
geogre : gw
```

Khi có thư đến địa chỉ của geogre, sendmail viết lại địa chỉ đó theo rule set 3 và 0. Rule set 0 dùng để chọn một trạm phân phối cục bộ . Chỉ trong trường hợp chọn được trạm phân phối thì sendmail mới tìm một địa chỉ trong tập tin aliases. Trong trường hợp trên, địa chỉ geogre sẽ được tìm và thay thế bằng địa chỉ gw. Sau đó sendmail đánh dấu geogre và xem như địa chỉ này đã giải quyết xong , và thêm địa chỉ gw vào danh sách các người nhận. Lúc này gw được xem như một địa chỉ mới và quá trình chuẩn hóa lại tiếp tục diễn ra. Quá trình như trên sẽ diễn ra cho đến khi không còn tìm được một địa chỉ mới nào nữa. Sendmail đánh dấu địa chỉ geogre thay vì xóa hẳn là để tránh trường hợp các địa chỉ tạo thành chu trình :

```
geogre : gw
```

```
gw : geogre
```

Nếu sendmail phát hiện một chu trình như vậy, nó sẽ thải hồi mail đó. Một vấn đề có thể gặp phải khi tạo alias là với cùng một tên ta lại tạo nhiều dòng alias.

Ví dụ:

```
staff : bob
```

```
staff : geogre
```

Hai dòng này sẽ gây ra lỗi trùng tên và kết quả là dòng đầu có thể bị bỏ qua. Tuy nhiên, việc này có thể khắc phục bằng cách mô tả vào tập tin cấu hình dòng tùy chọn sau:

```
OAliasFile=dbm:-A /etc/aliasdir/groups
```

Khi đó sendmail sẽ tự động nối hai dòng trên thành :

```
staff : bob, geogre
```

III.5.2 Các hình thức phân phối thư thông qua alias

Ta xét phần bên phải của một dòng alias, phần này gồm có 4 dạng như sau:

- Local: user
- Local: /file



- Local: |program
- Local::include:list

Hình thức local: user

user chỉ đến một tên, tên này có thể là đích đến cuối cùng hoặc có thể là phần local của một alias khác. Tuy nhiên, nếu user đó là cục bộ và trước user đó có dấu \ thì các alias tiếp theo của user đó sẽ không dùng đến, thư sẽ được chuyển đến hộp thư của user đó.

Hình thức local: /file

Thay vì phân thư đến hộp thư của user, ta có thể chỉ ra một tập tin để sendmail khi nội dung của thư vào tiếp theo phần cuối của tập tin đó. Ở đây, file là tập tin để viết thư vào.

Hình thức local:|program:

Một hình thức phân thư khác nữa là chuyển thư đó cho một chương trình xử lý khác. Program là tên chương trình đó, ta phải đặt nó trong dấu ngoặc kép (cùng với các tham số nếu có). Khi sử dụng hình thức này ta nên đặt tham số cho program vì khi sendmail thực hiện việc phân thư, nó sẽ sắp xếp các địa chỉ lại và sẽ bỏ ra các địa chỉ trùng lặp, lúc này tên program cũng được xem là một địa chỉ. Vấn đề là nếu ta dùng một chương trình để xử lý thư cho nhiều người (nhiều dòng alias sử dụng cùng tên program), khi đó nếu không có tham số cho mỗi chương sendmail sẽ xem đó như các địa chỉ trùng lặp và chỉ giữ lại một, kết quả là một số người sẽ không nhận được thư.

Hình thức local::include:list: Hình thức này, thư sẽ được xử lý và gửi đến một danh sách các người nhận.

III.5.3 Các alias đặc biệt

Cách vận hành của sendmail đòi hỏi phải có 2 alias đặc biệt định nghĩa trong tập tin aliases, đó là Postmaster và MAILDER-DAEMON

Postmaster :

- Chuẩn RFC822 yêu cầu mỗi site có 1 alias tên là postmaster. Các mail gửi đến postmaster được chuyển đến cho người có khả năng giải quyết các vấn đề về mail. Nếu postmaster không phải là một alias hoặc một người thực sự thì sendmail sẽ báo lỗi.
- Trong trường hợp một site không có một tài khoản thực sự mang tên postmaster, bạn phải tạo một alias mang tên đó. Alias này phải chỉ đến một hay nhiều người thực, mặc dù nó cũng có thể chỉ đến các tập tin lưu trữ hoặc một chương trình lọc.

MAILDER-DEAMON :

Khi một mail bị lỗi và trả lại, địa chỉ của người gửi thông báo lỗi thường sẽ lấy bằng giá trị của macro \$n và giá trị đó thường là mailer-daemon. Người dùng thường vô tình trả lời lại các mail thông báo lỗi, do đó cần phải có một alias cho mailer-daemon với địa chỉ chuyển thư tiếp theo là postmaster hoặc null.



III.5.4 Mailing list và forward

Sendmail có thể lấy danh sách địa chỉ người nhận từ tập tin aliases hoặc từ một tập tin ngoài. Một mailing list là tên của một user mà khi sendmail phân tích ra sẽ trở thành một danh sách người nhận. Các mailing list có thể là nội bộ (cả người nhận đều có trong tập tin alias) hoặc ngoài (danh sách người nhận được liệt kê trong các tập tin ngoài), hoặc có thể là kết hợp của hai loại trên.

Mailing list nội bộ:

Một mailing list nội bộ là một mục trong tập tin aliases với phần bên phải có nhiều hơn một người nhận. Ví dụ trong tập tin aliases có các dòng sau :

```
admin : bob, jim, phil
bob : \bob, /u/bob/admin/maillog
```

admin và bob chính là 2 mailing list vì nó được phân tích ra thành nhiều địa chỉ người nhận.

Các mailing list nội bộ có thể trở nên rất phức tạp khi được tổ chức trên diện rộng. Một ví dụ đơn giản như sau :

```
research : user1,user2
applications: user3, user4
admins: user5, user6
advertising: user7, user8
engineering:research, applications
frontoffice: admin, advertising
everyone: engineering,frontoffice
```

Ở đây, chỉ có 4 alias đầu tiên (research, applications, admins, advertising) được phân tích thành những người nhận thực sự. Ba dòng kế đó là sự kết hợp của 4 alias trước đó. Và dòng cuối là bao hàm tất cả những người nhận. Khi số lượng mailing list ít và không thường xuyên thay đổi, ta có thể quản lý rất hiệu quả bằng tập tin aliases. Tuy nhiên, khi số lượng mailing list khá lớn thì việc quản lý tập tin aliases sẽ rất khó khăn. Để khắc phục khó khăn này, các mailing list sẽ được khai báo trong các tập tin ngoài.

Các mailing list dạng INCLUDE:

Kí tự :include: ở bên phải của một alias báo hiệu cho sendmail biết là phải đọc danh sách người nhận từ một tập tin ngoài. Chỉ thị :include: được viết trong tập tin aliases như sau :

```
Localname: :include:/path
```



Với /path là đường dẫn tuyệt đối đến tập tin lưu danh sách người nhận. Nếu /path là đường dẫn gián tiếp thì nó phải tham chiếu đến thư mục hàng đợi của sendmail. Trong trường hợp sendmail không mở được tập tin này nó sẽ báo lỗi và bỏ qua tất cả những người nhận có trong tập tin đó. Sendmail đọc tập tin danh sách từng dòng một, các dòng trắng hoặc các dòng bắt đầu bằng ký tự # sẽ được bỏ qua. Trên cùng một dòng có thể có nhiều địa chỉ được phân cách nhau bởi dấu phẩy. Bản thân mỗi địa chỉ có thể là một alias trong tập tin aliases hoặc các loại địa chỉ khác như địa chỉ người dùng, tên chương trình hoặc tên tập tin. Ngoài ra, trong tập tin include có thể chứa một chỉ thị :include khác. Việc đọc tập tin ngoài được điều khiển bằng tùy chọn TimeOut.fileopen trong tập tin cấu hình. Tùy chọn này qui định thời gian tối đa cho phép để mở một tập tin và bao gồm phần kiểm tra tính an toàn.

Sendmail kiểm tra tính an toàn mỗi khi mở một tập tin. Nếu người dùng lúc đó là root thì tất cả các thành phần của đường dẫn cũng sẽ được kiểm tra. Trong lúc kiểm tra các thành phần của đường dẫn, sendmail sẽ in các lời cảnh báo khi phát hiện các thành phần này có thuộc tính group – hoặc world-writable. Sau khi mở tập tin, sendmail chuyển người dùng hiện tại thành chủ sở hữu của tập tin đó. Khi đó, người dùng sẽ cung cấp các định danh uid và gid của người gửi khi phân phối thư từ hàng đợi. Trong một số trường hợp sau, tập tin :include: sẽ không được phân phối bởi chương trình hoặc kết nối vào một tập tin khác :

- Nếu người sở hữu tập tin :include: có một shell mà shell đó không được khai báo trong thư mục /etc/shells.
- Nếu tập tin :include: có thuộc tính world-writable.
- Nếu tập tin :include: có thuộc tính greoup-writable và tùy chọn UnsafeGroupWrites được đặt giá trị True

Forwarder:

Chương trình sendmail cho phép mỗi người dùng có một tập tin lưu danh sách các địa chỉ sẽ nhận mail của mình. Tập tin này được chỉ định trong tùy chọn ForwardPath(J), và nó có tên là .forward nằm trong Home Directory của người dùng. Trong tập tin .forward chỉ ra địa chỉ email cần chuyển mail đến.

Ví dụ: Nội dung tập tin .forward

```
nvan@yahoo.com
```

IV. DNS và Sendmail

DNS và Sendmail là 2 dịch vụ có mối quan hệ mật thiết với nhau. Sendmail dựa vào dịch vụ DNS để chuyển mail từ mạng bên trong ra bên ngoài và ngược lại. Khi chuyển mail, Sendmail tìm MX record để xác định máy chủ nào cần chuyển mail đến. Cú pháp record MX:

```
[domain name] IN MX 0 [mail server]
```

Ví dụ:

```
t3h.com. IN MX 0 mailserver.t3h.com.
```

Một địa chỉ email thường có dạng sau:

```
username@subdomain...subdomain2.subdomain1.top-level-domain.
```

Thành phần bên phải dấu @ là địa chỉ miền. Tên miền có thể là một tổ chức hoặc một vùng địa lý nào đó. Nó phân biệt chữ hoa và chữ thường.



V. Những tập tin cấu hình Sendmail

Sendmail hoạt động dựa trên nhiều tập tin cấu hình khác nhau. Hai tập tin thường thao tác nhất là `/etc/aliases` và `/etc/sendmail.cf`. Trong đó tập tin `/etc/sendmail.cf` là tập tin cấu hình chính và quan trọng nhất của sendmail. Sendmail dựa vào tập tin cấu hình này để xử lý, phân phối mail nhận được.

V.1. Tập tin `/etc/sendmail.cf`

Thông tin cấu hình trong tập tin `sendmail.cf`. Tập tin này có cấu trúc dạng text. Nội dung tập tin được chia thành 3 nhóm thông tin cấu hình chính:

- Nhóm thứ nhất là những tham số cấu hình môi trường hoạt động của sendmail. Ví dụ các tùy chọn như: thời gian kết nối, thời gian tối đa một mail ở trong hàng đợi... và các đường dẫn đến các tập tin dữ liệu liên quan cần dùng khác.
- Nhóm thứ hai là phần định nghĩa cách hoạt động của sendmail như sendmail nhận chuyển mail cho miền nào đó, ...
- Nhóm thứ ba là phần mô tả các rule set mà người dùng định nghĩa lại phương thức xử lý của sendmail như : địa chỉ người gửi, người nhận và chọn các mailer xử lý. Tất cả những rule set này đều do người dùng thiết lập. Có một số rule set có ý nghĩa quan trọng như rule set 0, 1, 2, 3 v 4 sẽ được giới thiệu trong phần sau.

Trong tập tin `sendmail.cf` có những ký hiệu đặc tả những thông tin như sau:

Từ khóa	Ý nghĩa	Cú pháp
#	Từ khoá đầu dòng cho biết dòng này là dòng chú thích	#[chú thích] VD: # Đây là chú thích
M	Định nghĩa một mailer(Mail delivery agent)	Mname,field1=value1 Mprog,P=/bin/sh,FIsD,A=sh -c - \$u
D	Định nghĩa một macro	DXchuỗikýtự : định nghĩa macro X có giá trị là chuỗikýtự (Ví dụ Dxmailbox.hcmuns.edu.vn). D{Tênmacro}giátrị : định nghĩa một macro tên dài. Truy xuất macro này bằng \${Tênmacro} (Ví dụ : D{REMOTE}vnuhcm.edu.vn)
V	Định nghĩa phiên	Vn (n là số version)



	bản của tập tin cấu hình	
R	Định nghĩa một luật mới	Rlhs rhs chú thích. (Ví dụ : R\$+ \$:\$>22 gọi rule set 22)
S	Bắt đầu một rule set mới	Snn nn : tên rule set (Ví dụ : S96)
C	Định nghĩa một class macro	CXgiátrị1 giátrị2... : định nghĩa một class macro X với các giá trị là giátrị1, giátrị2... (Ví dụ : Cwlocalhost myhost)
F	Định nghĩa một class macro lấy giá trị từ một tập tin	FX/path/filename : class macro có tên X lấy giá trị từ tập tin filename. (Ví dụ : Fw/etc/mail/host_aliases)
O	Thiết lập một tùy chọn	OXoption cácthamsố (Ví dụ : OL9 #thiết lập log level là 9)
H	Định nghĩa một dòng header	H?mailerflag?name:template (Ví dụ : H?F?From:\$q)
P	Thiết lập giá trị độ ưu tiên của mail tùy theo loại mail	Pclass=nn (Ví dụ : Pjunk=-100)

V.2. Macro

Có những giá trị ta dùng lặp lại rất nhiều lần trong tập tin cấu hình sendmail.cf. Để thuận lợi trong việc sử dụng giá trị này, như tập trung về một chỗ để dễ dàng chỉnh sửa khi có một thay đổi nào đó, bằng cách định nghĩa một macro cho giá trị đó. Sau đó, bạn sử dụng macro đã được định nghĩa tại nhiều vị trí trong tập tin sendmail.cf một cách dễ dàng. Như đã giới thiệu ở trên, bạn dùng kí tự đặc tả D để định nghĩa một macro.

Ví dụ:

```
DRvnuhcm.edu.vn
D{REMOTE}vnuhcm.edu.vn
```

Trong đó:



R và {REMOTE} là tên của macro được định nghĩa

vnuhcm.edu.vn là giá trị của macro

class macro

Class macro cũng tương tự như macro. Tuy nhiên class macro khác macro ở đặc điểm là nó có thể có nhiều giá trị cùng một lúc. Để định nghĩa một class macro ta dùng kí tự đặc tả C

Ví dụ:

```
CW localhost vnuhcm.edu.vn
C{MY_NAMES} localhost vnuhcm.edu.vn
```

Trong đó W và {MY_NAMES} là tên class macro được định nghĩa. Chúng cùng lúc có 2 giá trị localhost và vnuhcm.edu.vn. Một số macro được sendmail định nghĩa sẵn:

Tên macro	Mô tả
N	Nhận dạng lỗi trong message của người gửi
V	Phiên bản của sendmail
W	Tên ngắn của máy tính(short hostname)
J	Tên bí danh của máy tính (canonical hostname)
M	Tên miền
K	UUCP node name
B	Ngày theo định dạng RFC1123

V.3. Sendmail macro

File macros của sendmail được lưu trong file /etc/mail/sendmail.mc, trong file này chứa các chỉ dẫn giúp quản trị hệ thống mail. Mỗi chỉ dẫn của sendmail.mc thường bắt đầu bằng từ khóa DOMAIN, FEATURE, or OSTYPE, theo sau các từ khóa này là các tham số.

Ví dụ:

```
FEATURE(`mailertable',`hash -o /etc/mail/mailertable.db')dnl
```

Ta có thể dùng m4 để dịch từ file sendmail.mc thành file /etc/mail/sendmail.cf

```
m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf
```

V.4. Tùy chọn (Option)

Trong quá trình cấu hình sendmail có rất nhiều tùy chọn mà bạn cần quan tâm. Tùy chọn được thiết lập bằng kí tự O ở đầu dòng. Nếu tên tùy chọn chỉ có một ký tự thì tên này sẽ đứng liền sau O và liền sau tên là giá trị của tùy chọn. Ngược lại nếu là tên dài, tên này cách O đúng một khoảng trắng và gán giá trị cho tùy chọn bằng dấu =.



Ví dụ:

OA/etc/aliases #chỉ đường dẫn đến tập tin aliases

O Timeout.queuereturn=5d # nếu mail ở trong hàng đợi quá 5 ngày, nó sẽ bị trả lại cho người gửi.

O QueueDirectory=/var/spool/mqueue #chỉ đường đến thư mục hàng đợi

O Timeout.queuewarn=4h #sau 4 giờ, nếu mail chưa chuyển đi được thì sẽ có một khuyến cáo phát sinh.

V.5. Định nghĩa các mailer

Một mailer có thể là một MTA hoặc là Mail Delivery Agent (trạm phân thư sau cùng). Do các mail có thể được phân đến nhiều loại địa chỉ khác nhau(địa chỉ người dùng, tập tin, chương trình...) nên ta cần phải định nghĩa các mailer khác nhau để làm những việc này.

Việc định nghĩa các mailer là một vấn đề quan trọng và rất cần thiết vì tất cả các mail cần phải được chuyển đến một mailer nào đó để tiếp tục đi đến người nhận. Rule set 0 sẽ đảm nhiệm việc chọn một mailer tiếp theo để chuyển mail. Ví dụ: một mail gửi cho một user cục bộ sẽ được chuyển đến một mailer cục bộ để từ đó chuyển đến hộp thư của người dùng. Ta có thể định nghĩa một mailer bằng kí tự đặc tả M. Ví dụ ta định nghĩa một mailer cục bộ như sau :

Mlocal, P=/bin/mail, F=IsDFMfSn, S=10, R=20, A=mail -d \$u

Trong ví dụ trên ta định nghĩa một mailer cục bộ có tên là local. Những thông số cho mailer bao gồm :

- Từ khóa P= : chỉ ra đường dẫn đến chương trình sẽ nhận và xử lý mail
- Từ khóa F= : chỉ ra các cờ của sendmail dùng cho mailer này.
- Từ khoá S=, R= : chỉ ra các rule set sẽ được dùng để viết lại địa chỉ người gửi và người nhận. Tùy theo đặc điểm của từng mailer mà ta dùng những rule set cho thích hợp. Hai từ khóa này cũng có thể được dùng để viết lại địa chỉ trên bì thư (envelope) và trên header. Khi đó ta có thể dùng S=21/31 để cấu hình sendmail dùng rule set 21 để viết lại địa chỉ trên bì thư và dùng rule set 31 để viết lại địa chỉ trên header
- Từ khóa A= : dùng để gọi các tham số cho chương trình xử lý mail
- Ngoài ra còn có từ khóa T= DNS/RFC822/SMTP : dùng để liệt kê 3 trường thông tin về mailer. Trường thứ nhất là loại MTA, ở đây ta dùng DNS để tìm địa chỉ nên trường thứ nhất có giá trị là DNS. Trường thứ 2 là loại địa chỉ người dùng. Trường thứ 3 là loại thông điệp lỗi sẽ được phát sinh.

V.6. Rule

Rule là phần quan trọng trong tập tin cấu hình sendmail.cf. Bạn định nghĩa một rule nhằm mục đích viết lại một địa chỉ này thành một địa chỉ khác. Kí tự R được dùng để định nghĩa một rule.

Ví dụ:

RS+<@\$*hcmussh.edu.vn.> S#relay\$@mailhost-XHNV-22.local\$: \$1<@\$2hcmussh.edu.vn>

Mỗi rule gồm 3 phần, các phần cách nhau một hay nhiều tab:

Rlhs rhs comment

Trong đó:

Lhs gọi là phần bên trái của luật

Rhs gọi là phần bên phải của luật

Hoạt động của một rule là: nếu điều kiện ở lhs thỏa thì rhs sẽ được thực hiện, ngược lại sẽ bỏ qua rule đó và thực hiện rule kế tiếp.

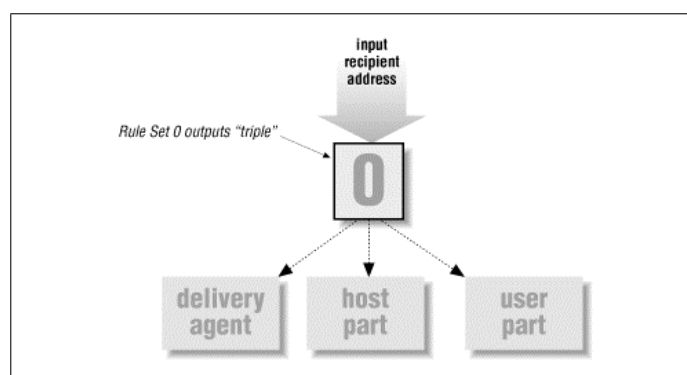
V.7. Rule set

Một tập hợp các rule tạo thành một rule set. Ký tự S dùng để định nghĩa một rule set. Theo sau S là một con số để phân biệt giữa các rule set và một rule set kết thúc khi gặp một rule set khác. Mỗi một rule set có một chức năng riêng do người dùng định nghĩa. Tuy nhiên từ rule set 0 đến rule set 5 được sendmail định nghĩa trước và chúng có những chức năng đặc trưng của mình.

Rule set	Nhiệm vụ
0	Kiểm tra lỗi và chọn trạm phân thư
1	Xử lý địa chỉ người gửi
2	Xử lý địa chỉ người nhận
3	Xử lý trước tất cả các địa chỉ để cho các rule set khác đọc được.
4	Viết lại địa chỉ dưới dạng bình thường (sau tất cả những xử lý ở rule set 3 và 96)
5	Rewrite unaliased local users

Rule set 0

Rule set 0 được gọi duy nhất một lần khi nhận được mail để tách địa chỉ mail thành 3 phần sau đó chọn ra mail delivery agent, như hình sau :



Rule set 0 phân giải một triple

Rule set 3

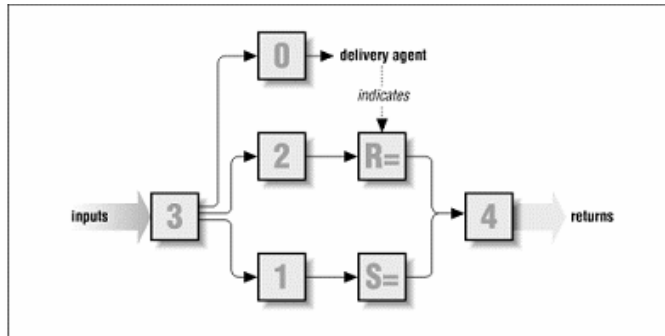


Từ rule set 1 đến rule set 4 có chức năng thay đổi địa chỉ người nhận và người gửi ở phần header, envelope thành dạng chuẩn mà sendmail có thể sử dụng được vì địa chỉ có thể viết ở nhiều dạng khác nhau như:

From : address (Full Name and other comments)

From : Full Name <address>

Sau khi được xử lý bởi rule set 3 địa chỉ sẽ được viết dưới dạng chuẩn (bỏ hết full name, ghi chú và các dấu ngoặc).



Luồng xử lý các địa chỉ qua các rule set

VI. Tập tin /etc/aliases

Tập tin /etc/aliases dùng để cấu hình alias cho người dùng. Có nghĩa là một người dùng có thể nhận mail với một tên bí danh khác.

Ví dụ: Trong trường hợp ta có một người dùng cục bộ là netadmin và người dùng này muốn nhận mail thông qua một tên là quanly, bạn sẽ khai báo trong tập tin /etc/aliases như sau :

quanly: netadmin

Sau đó thực thi lệnh `#newaliases`

VII. Cấu hình Mail Server với Sendmail

/etc/sendmail.cf là một tập tin cấu hình chính của sendmail. Khi cấu hình Mail Server với sendmail, bạn cần quan tâm đến một vài tham số quan trọng sau:

Các tham số cần cấu hình	Giải thích
Cwlocalhost vnuhcm.edu.vn	Cấu hình sendmail nhận mail cho miền vnuhcm.edu.vn
#Smart relay host Dsvnuser.vnuhcm.edu.vn	Các mail sẽ được chuyển lên máy vnuser.vnuhcm.edu.vn để gửi đi (relay host)
#maximum number of recipients per SMTP envelope	Giới hạn số người nhận đối với một mail



O MaxRecipientsPerMessage=50	
#maximum message size O MaxMessageSize=3000000	Giới hạn kích thước tối đa của một mail (tính bằng byte)

Ngoài ra ta phải cấu hình cho sendmail nhận chuyển mail cho miền nào bằng cách đã khai báo chúng trong tập tin `/etc/sendmail.cf`. Ví dụ, bạn muốn chuyển mail cho miền `"vnuhcm.edu.vn"`. Khi đó, bạn cấu hình tập tin `/etc/mail/access` như sau:

```
vnuhcm.edu.vn      RELAY
```

Dòng khai báo này nhằm mục đích cho phép các client trong miền `vnuhcm.edu.vn` được gửi mail thông qua mail server này. Bên cạnh đó, nó còn có mục đích khác là chống relay nghĩa là những mail nào nằm ngoài miền này sẽ không được mail server này chuyển đi. Sau khi chỉnh sửa tập tin `/etc/aliases`, bạn cần phải chuyển tập tin từ dạng văn bản sang dạng chuẩn để sendmail có thể đọc được bằng lệnh sau :

```
#makemap hash access < access
```

Khi đã cấu hình xong các bước trên ta có thể khởi động lại sendmail bằng một trong những dòng lệnh sau :

```
#chkconfig sendmail on
#/etc/rc.d/init.d/sendmail restart
```

VIII. Một số file cấu hình trong sendmail

Thông thường các file cấu hình của sendmail được đặt trong thư mục `/etc/mail`.

VIII.1. File `/etc/mail/access`

Chỉ định các sendmail sẽ RELAY hoặc REJECT cho host hoặc network gửi thư qua mail server. Cú pháp khai báo như sau:

```
localhost.localdomain    RELAY
localhost                 RELAY
127.0.0.1                RELAY
192.168.1.16             RELAY
my-site.com               RELAY
abc@yahoo.com            REJECT
```

Dùng lệnh `#makemap hash access<accesss` để chuẩn hóa file access text thành file access.db

VIII.2. File `/etc/mail/local-host-names`

File `/etc/mail/local-host-names` hay `/etc/mail/sendmail.cw` cho phép chỉ định danh sách các host và domain mà Mail Server chịu trách nhiệm quản lý mail. Cú pháp khai báo file `local-host-names`:

```
my-site.com
another-site.com
```



Ta chỉ định Mail Server quản lý mail cho hai miền my-site.com và other-site.com. tuy nhiên ta cần chỉ định MX RR trong DNS để thông báo cho các Mail Server khác biết rằng Mail Server nào chịu trách nhiệm nhận mail cho miền other-site.com.

```
another-site.com. IN MX 10 mail.my-site.com.
```

VIII.3.File /etc/mail/virtusertable

Chỉ định tập hợp các chỉ dẫn cơ bản hỗ trợ cho các vấn đề nhận thư. Cú pháp khai báo:

Dòng 1: webmaster@another-site.com webmasters

Dòng 2: @another-site.com marc

Dòng 3: sales@my-site.com sales@another-site.com

Dòng 4: paul@my-site.com paul

Dòng 5: finance@my-site.com paul

Dòng 6: @my-site.com error:nouser User unknown

Ý nghĩa:

- Dòng 1: Tất cả các mail của webmaster@another-site.com sẽ được gửi vào local user: webmasters
- Dòng 2: Tất cả các mail gửi vào miền other-site.com sẽ được chuyển vào local user: marc
- Dòng 3: Tất cả các mail gửi vào địa chỉ sales@my-site.com sẽ được gửi tới sales@another-site.com.
- Dòng 4,5: Tất cả các mail gửi vào mail paul@ my-site.com, finance@my-site.com, sẽ chuyển vào local user: paul.
- Dòng 6: các mail gửi vào domain my-site.com sẽ được thông báo lỗi trở lại người gửi là nouser User unknown

Dùng lệnh #makemap hash virtusertable<virtusertable để chuẩn hóa file virtusertable text thành file virtusertable.db

VIII.4.File /etc/mail/mailertable

Được sử dụng để chuyển mail tới một máy mail server khác. Cú pháp của file mailertable:

```
domain smtp:<mailer_address>
```

Ví dụ:

```
domain.com smtp:mail.newserver.com
```

```
domain2.com smtp:[mail.otherserver.com]
```

Hoặc ta có thể khai báo mailer như sau:

```
.vlth.hcmuns.edu.vn relay:vlth-svr.hcmuns.edu.vn
```

```
csc-tata.hcmuns.edu.vn relay:[172.29.8.13]
```

Khi mail gửi vào miền csc-tata.hcmuns.edu.vn thì mail server sẽ chuyển cho máy 172.29.8.13 xử lý.



Dùng lệnh `#makemap hash mailtable<mailtable` để chuẩn hóa file mailtable text thành file mailtable.db

VIII.5. File /etc/mail/domaintable

Khai báo danh sách các domain tương ứng với domain cục bộ. Hỗ trợ trong việc thay đổi tên miền, khai báo hai hay nhiều tên miền trở về cùng một mailbox. Cú pháp khai báo như sau:

```
olddomain.com      newdomain.com
```

Dùng lệnh `#makemap hash domaintable<domaintable` để chuẩn hóa file domaintable text thành file domaintable.db

IX. Cấu hình POP Mail Server

Có hai cách cài đặt POP Server:

Cách 1: Ta cần phải cài đặt gói tin `imap-2002d-3.i386.rpm` vì trong package này có chứa POP Server, trong các đĩa CDROM của Fedora chưa có package này do đó ta phải download từ site: <http://rpmfind.net>. Khởi động POP Server ta dùng lệnh sau:

```
#chkconfig pop3 on
#service xinetd restart
```

Hoặc sau khi ta cài đặt IMAP package xong ta dùng lệnh `setup ->System Services -> IPOP3`, sau đó dùng lệnh `#!/etc/init.d/xinetd restart`.

Cách 2: Cài đặt gói `dovecot-0.99.10.5-0.FC2.rpm` từ CDROM Fedora Core 2, sau đó ta mở file cấu hình `/etc/dovecot.conf` để thay đổi các thông số sau:

- `protocols = imap imaps pop3 pop3s` ; chỉ định các protocol sử dụng
- `imap_listen = *` ; chỉ định trạng thái listen trên card mạng cho IMAP
- `pop3_listen = *` ; chỉ định trạng thái listen trên card mạng cho POP3

sau đó thực thi lệnh :

```
#chkconfig dovecot on
#service dovecot restart
```

X. Cài đặt và cấu hình Webmail - Openwebmail

Open Webmail là hệ thống Webmail dựa trên chương trình Neomail version 1.14. Open Webmail được thiết kế để chạy trên hệ thống Unix & Linux cung cấp cho người dùng sử dụng Mail qua Web. Trên Linux ta có thể download file *.rpm từ địa chỉ:

<http://openwebmail.org/openwebmail/download/redhat/rpm/release/>

<http://openwebmail.org/openwebmail/download/redhat/rpm/packages/>

Tuy nhiên nếu ta muốn cài đặt Open Webmail từ source code (*.tar.gz) từ địa chỉ:

```
http://openwebmail.org/openwebmail/download/release/
http://openwebmail.org/openwebmail/download/packages/
```



Ta có thể vào Website sau để xem trợ giúp về chương trình:

<http://openwebmail.org/openwebmail/help/en/index.html>

X.1. Cài đặt và cấu hình Open Webmail

X.1.1 Cài đặt từ file nhị phân *.rpm

Bước 1: Ta dùng lệnh `rpm -ivh package*.rpm`

Đối với Fedora Core ta cần các package sau:

- perl-Compress-Zlib-1.33-6.i386.rpm
- perl-suidperl-5.8.3-18.1.i386.rpm
- perl-Text-Iconv-1.2-fc1.i386.rpm
- openwebmail-2.51-1.i386.rpm

Đối với phiên bản trước của Linux thì ta cần tham khảo thêm Website <http://openwebmail.org/openwebmail/download/> để biết rõ hơn.

Bước 2: Đối với Fedora Core yêu cầu phải có MIME-Base64-3.0 cho nên ta cần cài thêm phần mềm này:

- #tar xzvf MIME-Base64-3.00.tar.gz
- #cd MIME-Base64-3.00/
- #perl Makefile.PL
- #make
- #make install

Bước 3: Thực thi lệnh `# /var/www/cgi-bin/openwebmail/openwebmail-tool.pl --init`

Bước 4: Sau đó Open Webmail yêu cầu thay đổi thông tin trong file `/var/www/cgi-bin/openwebmail/etc/defaults/dbm.conf`

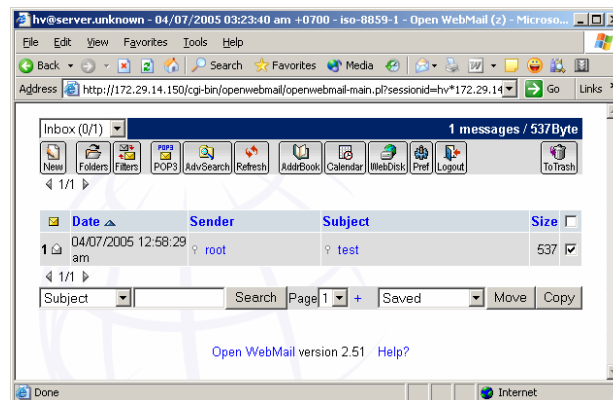
```
dbm_ext          .db
dbmopen_ext      .db
dbmopen_haslock no
```

Bước 5: Thực thi lại lệnh `# /var/www/cgi-bin/openwebmail/openwebmail-tool.pl --init`

Bước 6: Truy cập vào địa chỉ <http://server/cgi-bin/openwebmail/openwebmail.pl> để truy xuất vào Webmail Server để sử dụng



Bước 7: Login và sử dụng OpenWebmail



X.1.2 Một số thông tin cấu hình cơ bản

Mọi thông tin cấu hình của Open Webmail nằm trong file `/var/www/cgi-bin/openwebmail/etc/openwebmail.conf`. Ta cần tham khảo các thông tin cấu hình sau:

```
releasedate      20050228      #ngày cuối cập nhật phiên bản
domainnames     hcm.vn        #chỉ định tên domain
auth_module     auth_unix.pl
mailspooldir    /var/spool/mail # chỉ định spool mail cho user.
ow_cgidir       /var/www/cgi-bin/openwebmail
ow_cgiurl       /cgi-bin/openwebmail
ow_htmlidir    /var/www/data/openwebmail
ow_htmlurl     /data/openwebmail
logfile         /var/log/openwebmail.log
```

X.2. Cài đặt Open Webmail từ Source code

Ta download phần mềm sau từ địa chỉ <http://openwebmail.org/openwebmail/download/packages/>. Apache Web server cho phép thực thi chương trình cgi.

- Perl 5.005 or later
- CGI.pm-3.05.tar.gz
- MIME-Base64-3.01.tar.gz



- libnet-1.19.tar.gz
- Digest-1.08.tar.gz
- Digest-MD5-2.33.tar.gz
- Text-Iconv-1.2.tar.gz
- libiconv-1.9.1.tar.gz (required nếu hệ thống không hỗ trợ iconv)
- openwebmail-2.51.tar.gz

Tuy nhiên ta cần tham khảo địa chỉ sau để cập nhật thông tin cho hợp lệ để chọn các gói trên tại địa chỉ: <http://openwebmail.org/openwebmail/doc/readme.txt>. Sau khi ta download xong các phần mềm trên ta thực hiện các bước như sau:

Bước 1: Cài phần mềm CGI.pm

```
cd /tmp
tar -zxvf CGI.pm-3.05.tar.gz
cd CGI.pm-3.05
perl Makefile.PL
make
make install
```

Bước 2: Cài phần mềm MIME-Base64

```
cd /tmp
tar -zxvf MIME-Base64-3.01.tar.gz
cd MIME-Base64-3.01
perl Makefile.PL
make
make install
```

Bước 3: Cài phần mềm libnet

```
cd /tmp
tar -zxvf libnet-1.19.tar.gz
cd libnet-1.19
perl Makefile.PL (ans 'no' if asked to update configuration)
make
make install
```

Bước 4: cài phần mềm Text-Iconv-1.2

```
cd /tmp
tar -zxvf libiconv-1.9.1.tar.gz
```



```

cd libiconv-1.9.1
./configure
make
make install
cd /tmp
tar -zxvf Text-lconv-1.2.tar.gz
cd Text-lconv-1.2
perl Makefile.PL
make
make test
make install

```

Bước 5: cài đặt OPENWEBMAIL

Phiên bản mới nhất của Open Webmail được cung cấp tại Website:

<http://openwebmail.org/openwebmail/>

1. cd /var/www

```

tar -zxvBpf openwebmail-X.XX.tar.gz
mv data/openwebmail html/
rmdir data

```

2. cd /var/www/cgi-bin/openwebmail/etc

Thay đổi auth_unix.conf từ defaults/auth_unix.conf

- a. set passwdfile_encrypted to '/etc/shadow'
- b set passwdmkdb to 'none'

Thay đổi openwebmail.conf

Đặt mailspooldir thành '/var/spool/mail'

Đặt ow_htmlidir thành '/var/www/html/openwebmail'

Đặt ow_cgidir thành '/var/www/cgi-bin/openwebmail'

```

Đặt spellcheck thành /usr/bin/ispell -a -S -w "-" -d
@@@DICTIONARY@@@ -p @@@PDICNAME@@@

```

3. Thêm thông tin

```

/var/log/openwebmail.log {
    postrotate
        /usr/bin/killall -HUP syslogd
    endscrip

```



```
}
```

Tới file /etc/logrotate.d/syslog để ghi nhận log của openwebmail.log

4. Thực thi lệnh /var/www/cgi-bin/openwebmail/openwebmail-tool.pl –init



BÀI 16

PROXY SERVER

Tóm tắt

Lý thuyết: 5 tiết - Thực hành: 5 tiết.

Mục tiêu	Các mục chính	Bài tập bắt buộc	Bài tập làm thêm
Bài học giới thiệu cơ chế tổ chức và quản trị dịch vụ Proxy để hỗ trợ chia sẻ kết nối Internet và thiết lập chính sách bảo mật cho hệ thống mạng nội bộ.	I. Giới thiệu Firewall II. Giới thiệu Squid Proxy II. Cấu hình Squid Proxy	Bài tập 6.1 (Dịch vụ Proxy)	



I. Firewall

Internet là một hệ thống mở, đó là điểm mạnh và cũng là điểm yếu của nó. Chính điểm yếu này làm giảm khả năng bảo mật thông tin nội bộ của hệ thống. Nếu chỉ là mạng LAN thì không có vấn đề gì, nhưng khi đã kết nối Internet thì phát sinh những vấn đề hết sức quan trọng trong việc quản lý các tài nguyên quý giá - nguồn thông tin - chống việc truy cập bất hợp pháp trong khi vẫn cho phép người được ủy nhiệm sử dụng các nguồn thông tin mà họ được cấp quyền, và phương pháp chống rò rỉ thông tin trên các mạng truyền dữ liệu công cộng (Public Data Communication Network). Yêu cầu xây dựng hệ thống an ninh ngày càng quan trọng vì những lý do sau:

- Các đối thủ cạnh tranh luôn tìm cách để lấy được mọi thông tin của nhau.
- Các tay hacker tìm cách xâm nhập phá hoại hệ thống mạng nội bộ ...

I.1. Giới thiệu về Firewall

Thuật ngữ Firewall có nguồn gốc từ một kỹ thuật thiết kế trong xây dựng để ngăn chặn, hạn chế hỏa hoạn. Trong công nghệ thông tin, Firewall là một kỹ thuật được tích hợp vào hệ thống mạng để chống lại việc truy cập trái phép, bảo vệ các nguồn tài nguyên cũng như hạn chế sự xâm nhập vào hệ thống của một số thông tin khác không mong muốn. Cụ thể hơn, có thể hiểu firewall là một cơ chế bảo vệ giữa mạng tin tưởng (trusted network), ví dụ mạng intranet nội bộ, với các mạng không tin tưởng mà thông thường là Internet. Về mặt vật lý, firewall bao gồm một hoặc nhiều hệ thống máy chủ kết nối với bộ định tuyến (router) hoặc có chức năng router. Về mặt chức năng, Firewall có nhiệm vụ:

- Tất cả các trao đổi dữ liệu từ trong ra ngoài và ngược lại đều phải thực hiện thông qua firewall.
- Chỉ có những trao đổi được cho phép bởi hệ thống mạng nội bộ (trusted network) mới được quyền lưu thông qua firewall.
- Các phần mềm quản lý an ninh chạy trên hệ thống máy chủ bao gồm :

Quản lý xác thực (Authentication): có chức năng ngăn cản truy cập trái phép vào hệ thống mạng nội bộ. Mỗi người sử dụng muốn truy cập hợp lệ phải có một tài khoản (account) bao gồm một tên người dùng (username) và mật khẩu (password).

Quản lý cấp quyền (Authorization): cho phép xác định quyền sử dụng tài nguyên cũng như các nguồn thông tin trên mạng theo từng người, từng nhóm người sử dụng.

Quản lý kế toán (Accounting Management): cho phép ghi nhận tất cả các sự kiện xảy ra liên quan đến việc truy cập và sử dụng nguồn tài nguyên trên mạng theo từng thời điểm (ngày/giờ) và thời gian truy cập đối với vùng tài nguyên nào đã được sử dụng hoặc thay đổi bổ sung ...

I.2. Những chính sách Firewall

Bước đầu tiên trong việc cấu hình Firewall là thiết lập các chính sách:

- Những dịch vụ nào cần ngăn chặn.
- Những host nào cần phục vụ.
- Mỗi nhóm cần truy xuất những dịch vụ nào.
- Mỗi dịch vụ sẽ được bảo vệ như thế nào.



I.3. Các loại Firewall và cách hoạt động

I.3.1 Packet filtering (Bộ lọc gói tin)

Loại Firewall này thực hiện việc kiểm tra số nhận dạng địa chỉ của các packet để từ đó cấp phép cho chúng lưu thông hay ngăn chặn. Các thông số có thể lọc được của một packet như:

- Địa chỉ IP nơi xuất phát (source IP address).
- Địa chỉ IP nơi nhận (destination IP address).
- Cổng TCP nơi xuất phát (source TCP port).
- Cổng TCP nơi nhận (destination TCP port).

Loại Firewall này cho phép kiểm soát được kết nối vào máy chủ, khóa việc truy cập vào hệ thống mạng nội bộ từ những địa chỉ không cho phép. Ngoài ra, nó còn kiểm soát hiệu suất sử dụng những dịch vụ đang hoạt động trên hệ thống mạng nội bộ thông qua các cổng TCP tương ứng.

I.3.2 Application gateway

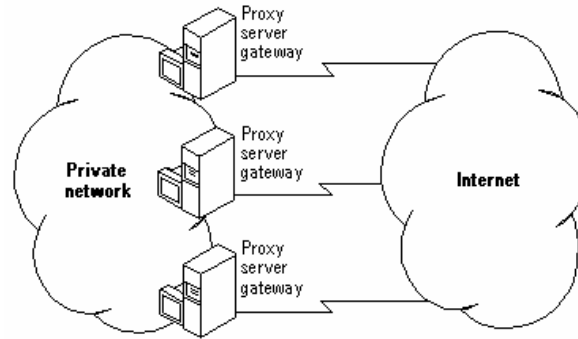
Đây là loại Firewall được thiết kế để tăng cường chức năng kiểm soát các loại dịch vụ dựa trên những giao thức được cho phép truy cập vào hệ thống mạng. Cơ chế hoạt động của nó dựa trên mô hình Proxy Service. Trong mô hình này phải tồn tại một hay nhiều máy tính đóng vai trò Proxy Server. Một ứng dụng trong mạng nội bộ yêu cầu một đối tượng nào đó trên Internet, Proxy Server sẽ nhận yêu cầu này và chuyển đến server trên Internet. Khi server trên Internet trả lời, Proxy Server sẽ nhận và chuyển ngược lại cho ứng dụng đã gửi yêu cầu. Cơ chế lọc của packet filtering kết hợp với cơ chế “đại diện” của application gateway cung cấp một khả năng an toàn và uyển chuyển hơn, đặc biệt khi kiểm soát các truy cập từ bên ngoài.

Ví dụ: Một hệ thống mạng có chức năng packet filtering ngăn chặn các kết nối bằng TELNET vào hệ thống ngoại trừ một máy duy nhất - TELNET application gateway là được phép. Một người muốn kết nối vào hệ thống bằng TELNET phải qua các bước sau:

- Thực hiện telnet vào máy chủ bên trong cần truy cập.
- Gateway kiểm tra địa chỉ IP nơi xuất phát của người truy cập để cho phép hoặc từ chối.
- Người truy cập phải vượt qua hệ thống kiểm tra xác thực.
- Proxy Service tạo một kết nối Telnet giữa gateway và máy chủ cần truy cập.
- Proxy Service liên kết lưu thông giữa người truy cập và máy chủ trong mạng nội bộ.

Cơ chế bộ lọc packet kết hợp với cơ chế proxy có nhược điểm là hiện nay các ứng dụng đang phát triển rất nhanh, do đó nếu các proxy không đáp ứng kịp cho các ứng dụng, nguy cơ mất an toàn sẽ tăng lên.

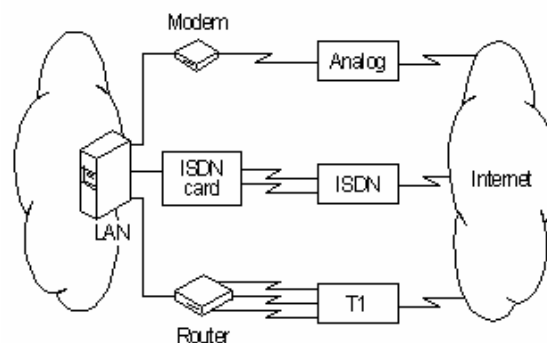
Thông thường những phần mềm Proxy Server hoạt động như một gateway nối giữa hai mạng, mạng bên trong và mạng bên ngoài.



Đường kết nối giữa Proxy Server và Internet thông qua nhà cung cấp dịch vụ Internet (Internet Service Provider - ISP) có thể chọn một trong các cách sau:

- Dùng modem analog: sử dụng giao thức SLIP/PPP để kết nối vào ISP và truy cập Internet. Dùng dial-up thì tốc độ bị giới hạn, thường là 28.8 Kbps - 36.6 Kbps. Hiện nay đã có modem analog tốc độ 56 Kbps nhưng chưa được thử nghiệm nhiều. Phương pháp dùng dial-up qua modem analog thích hợp cho các tổ chức nhỏ, chỉ có nhu cầu sử dụng dịch vụ Web và e-mail.
- Dùng đường ISDN: Dịch vụ ISDN (Integrated Services Digital Network) đã khá phổ biến ở một số nước tiên tiến. Dịch vụ này dùng tín hiệu số trên đường truyền nên không cần modem analog, cho phép truyền cả tiếng nói và dữ liệu trên một đôi dây. Các kênh thuê bao ISDN (đường truyền dẫn thông tin giữa người sử dụng và mạng) có thể đạt tốc độ từ 64 Kbps đến 138,24 Mbps. Dịch vụ ISDN thích hợp cho các công ty vừa và lớn, yêu cầu băng thông lớn mà việc dùng modem analog không đáp ứng được.

Phần cứng dùng để kết nối tùy thuộc vào việc nối kết trực tiếp Proxy Server với Internet hoặc thông qua một router. Dùng dial-up đòi hỏi phải có modem analog, dùng ISDN phải có bộ phối ghép ISDN cài trên server.



Việc chọn lựa cách kết nối và một ISP thích hợp tùy thuộc vào yêu cầu cụ thể của công ty, ví dụ như số người cần truy cập Internet, các dịch vụ và ứng dụng nào được sử dụng, các đường kết nối và cách tính cước mà ISP có thể cung cấp.



II. Squid Proxy

II.1. Giới thiệu Squid

Squid là một chương trình internet proxy-caching có vai trò tiếp nhận các yêu cầu từ các client và chuyển cho Internet server thích hợp. Đồng thời, nó sẽ lưu lên đĩa những dữ liệu được trả về từ Internet server – gọi là caching. Chương trình này dùng để cấu hình Proxy Server. Vì vậy ưu điểm của squid là khi một dữ liệu mà được yêu cầu nhiều lần thì Proxy Server sẽ lấy thông tin từ cache trả về cho client. Điều này làm cho tốc độ truy xuất Internet nhanh hơn và tiết kiệm băng thông. Squid dựa trên những đặc tả của giao thức HTTP nên nó chỉ là một HTTP Proxy. Do đó Squid chỉ có thể là một proxy cho những chương trình mà chúng dùng giao thức này để truy cập Internet.

II.2. Những giao thức hỗ trợ trên Squid

Squid proxy hỗ trợ những giao thức sau:

- Proxying and caching of HTTP, FTP, and other URLs.
- Proxying for SSL.
- Cache hierarchies.
- ICP, HTCP, CARP, Cache Digests.
- Transparent caching.
- WCCP - Web Cache Communication Protocol (Squid v2.3 and above).
- Extensive access controls.
- HTTP server acceleration.
- SNMP.
- Caching of DNS lookups.

II.3. Trao đổi cache

Squid có khả năng chia sẻ dữ liệu giữa những cache với nhau. Việc chia sẻ này mang lại những lợi ích như :

- User Base: nếu số lượng client truy cập Internet thông qua proxy càng nhiều thì khả năng một đối tượng nào đó được yêu cầu 2 lần sẽ cao hơn.
- Giảm tải truy xuất (Reduce load) cho đường truyền.
- Disk space: Nếu bạn chuyển cân bằng giữa các cache với nhau sẽ tránh được việc sao lại dữ liệu đã lưu. Do đó dung lượng đĩa cứng dành cho việc lưu trữ cache sẽ giảm.

II.4. Cài đặt Squid Proxy

II.4.1 Các thư mục mặc định của Squid

- /usr/local/squid: thư mục cài đặt squid
- /usr/local/squid/bin: thư mục lưu binary squid và những tool được hỗ trợ.
- /usr/local/squid/cache: thư mục lưu những dữ liệu được cache. Đây là thư mục mặc định, bạn có thể thay đổi vị trí thư mục này.



- /usr/local/squid/etc: những tập tin cấu hình squid nằm trong thư mục này.
- /usr/local/squid/src: thư mục lưu source code squid được download từ net.

II.4.2 Cài đặt squid từ package rpm

- Khi cài đặt squid trong hệ điều hành Linux, vị trí các thư mục mặc định có những điểm khác sau:
 - /usr/sbin: Lưu những thư viện của Squid .
 - /etc/squid: Lưu các tập tin cấu hình squid.
 - /var/log/squid: Lưu các tập tin log của squid.
- Bạn dùng lệnh sau để cài squid:
- rpm -i squid-version.i386.rpm

II.5. Cấu hình

II.5.1 Tập tin cấu hình

Tất cả những tập tin cấu hình Squid được lưu trong thư mục /usr/local/squid/etc (Linux: /etc/squid). Một tập tin cấu hình quan trọng nhất quyết định sự hoạt động của Squid là squid.conf. Trong tập tin cấu hình này có 125 tag tùy chọn, nhưng chỉ có một số tùy chọn được cấu hình, và những dòng chú thích bắt đầu bằng dấu "#". Bạn chỉ cần thay đổi 8 tùy chọn cơ bản là squid hoạt động được. Những tùy chọn còn lại bạn có thể tìm hiểu thêm để hiểu rõ những tính năng mà Squid hỗ trợ.

II.5.2 Những tùy chọn cơ bản

Bạn cần phải thay đổi một số tùy chọn cơ bản để squid hoạt động. Mặc định squid cấm tất cả browser truy cập. Sau đây là những miêu tả về các tùy chọn này.

http_port: cấu hình cổng HTTP mà squid sẽ lắng nghe những yêu cầu được gửi đến.

Cú pháp: http_port <cổng>

Mặc định: http_port 3128. Ta thường thay đổi cổng này là 8080 và được khai báo như sau:

```
http_port 8080
```

Những tùy chọn ảnh hưởng đến cache:

Cache_mem ; Chỉ định bộ nhớ thích hợp cho các đối tượng (In-Transit objects, Hot Objects, Negative-Cached objects).

Cache_swap_low ; Chỉ định kích thước thấp nhất của cache object khi thay thế (được tính bằng % với vùng nhớ cache)

Cache_swap_high ; Chỉ định kích thước cao nhất của cache object khi thay thế (được tính bằng % với vùng nhớ cache)

Đường dẫn các tập tin log và thư mục cache:

Cache_dir: cấu hình thư mục lưu trữ dữ liệu được cache, Mặc định cache_dir được khai báo như sau:

```
cache_dir /usr/local/squid/cache 100 16 256
```



Thư mục cache có kích thước mặc định là 100Mbps, 16 level-1 subdirectory của thư mục /usr/local/squid/cache, level-2 subdirectory cho mỗi level-1.

Cache_access_log: Lưu trữ các activity request của client yêu cầu đến proxy server để truy xuất Web.

cache_access_log /var/log/squid/access.log

Cache_log: Lưu trữ các thông tin chung về cache.

cache_log /var/log/squid/cache.log

Cache_store_log: Lưu trữ các thông tin về đối tượng được cache trên proxy, thời gian lưu trữ,...

Cache_effective_user, cache_effective_group: người dùng và nhóm có thể thay đổi squid.

Ví dụ:

cache_effective_user squid

cache_effective_group squid

Access Control List và Access Control Operators:

Bạn có thể dùng Access Control List và Access Control Operators để ngăn chặn, giới hạn việc truy xuất dựa vào tên miền, địa chỉ IP đích (IP của máy hoặc mạng). Mặc định, squid từ chối phục vụ tất cả. Vì vậy, bạn phải cấu hình lại tham số này. Cú pháp định nghĩa Access List dùng tag acl.

acl aclname acltype string1 ..

acl aclname acltype "file" ...

Ví dụ: Một số ví dụ mẫu về acl

```
acl aclname src ip-address/netmask ... (clients IP address)
acl aclname dst ip-address/netmask ... (range of addresses)
acl aclname srcdomain .foo.com ... # reverse lookup, client IP
acl aclname dst ip-address/netmask ... (URL host's IP address)
acl aclname dstdomain .foo.com ... # Destination server from URL
acl aclname time [day-abbrevs] [h1:m1-h2:m2]
acl aclname url_regex [-i] ^http:// ... # regex matching on whole URL
acl aclname port 80 70 21 ... 0-1024
acl aclname proto HTTP FTP ...
acl aclname method GET POST ...
```

Thẻ (Tag) điều khiển truy xuất HTTP (dấu ! để chỉ phủ định của aclname)

http_access allow|deny [!]aclname ...

Thẻ (Tag) điều khiển truy xuất cache_peer

cache_peer_access cache-host allow|deny [!]aclname ...

Ví Dụ: Bạn chỉ cho phép mạng 172.16.1.0/24 được dùng proxy server bằng từ khóa src trong acl



```
acl MyNetwork src 172.16.1.0/255.255.255.0
http_access allow MyNetwork.
http_access deny all
```

Bạn cũng có thể cấm các máy truy xuất đến những site không được phép (những site có nội dung phù hợp) bằng từ khóa `dstdomain` trong `acl`,

Ví dụ:

```
acl BadDomain dstdomain yahoo.com
http_access deny BadDomain
http_access deny all
```

Nếu danh sách cấm truy xuất đến các site dài quá, bạn có thể lưu chúng vào một tập tin dạng văn bản. Nội dung của tập tin này là danh sách các địa chỉ. Ví dụ như sau:

```
acl BadDomain dstdomain "/etc/squid/danhsachcam"
http_access deny BadDomain
```

Theo như ví dụ trên, tập tin `"/etc/squid/danhsachcam"` lưu các địa chỉ không được phép truy xuất. Các địa chỉ này được ghi lần lượt theo từng dòng. Nếu có nhiều `acl`, ứng với mỗi `acl` phải có một `http_access`. Xem ví dụ minh họa sau:

```
acl MyNetwork src 172.16.1.0/255.255.255.0
acl BadDomain dstdomain www.yahoo.com
http_access deny BadDomain
http_access allow MyNetwork
http_access deny all
```

Như vậy cấu hình trên cho thấy proxy server cấm các máy truy xuất đến site `www.yahoo.com` và chỉ có đường mạng `172.16.1.0/32` là được phép dùng proxy. `"http_access deny all"` : cấm tất cả ngoài những truy cập còn lại.

Giới hạn thời gian truy xuất: ta dùng `acl` type kiểu là `time`, trong đó `MTWHF` tương ứng là thứ hai, thứ ba, thứ tư, thứ năm, thứ sáu.

```
acl business_hours time MTWHF 9:00-17:00
http_access allow business_hours
```

Chỉ định hostname cho Server: `Visible_hostname <hostname>` để chỉ định hostname cho squid proxy.

Cache_peer: Nếu proxy không kết nối trực tiếp đến internet (không có địa chỉ IP thật) hoặc proxy nằm sau một firewall thì ta phải cấu hình proxy này truy vấn đến proxy khác bằng tham số: `cache_peer`. Cú pháp của `cache_peer`:

```
cache_peer hostname type http_port icp_port
type = 'parent','sibling' hoặc multicast
```

Ví dụ: Các trường thành viên trong ĐHQG khai báo như sau:

```
cache_peer vnuserv.vnuhcm.edu.vn parent 8080 8082
```



Cấu hình trên cho thấy, proxy sẽ truy vấn đến proxy “cha” vnuser.vnuhcm.edu.vn với tham số parent thông qua cổng http_port là 8080 và icp_port là 8082. Ngoài ra, trong cùng một mạng nếu có nhiều proxy, bạn có thể cấu hình các proxy này truy vấn lẫn nhau như sau:

```
cache_peer proxy2.vnuhcm.edu.vn sibling 8080 8082
cache_peer proxy3.vnuhcm.edu.vn sibling 8080 8082
```

sibling: có nghĩa chỉ định proxy khai báo là proxy ngang cấp với proxy hiện tại.

II.6. Khởi động Squid

Sau khi đã cài đặt và cấu hình squid, bạn phải tạo thư mục cache - trước khi khởi động - squid bằng lệnh: **squid -z**. Nếu trong quá trình tạo tập tin cache bị lỗi, bạn chú ý đến các quyền truy xuất thư mục cache được khai báo trong tham số cache_dir. Có thể thư mục đó không có quyền được phép ghi. Khi đó, bạn phải thay đổi bằng dòng lệnh sau:

```
chown squid:squid /var/spool/squid
chmod 770 /var/spool/squid
```

Sau khi tạo xong thư mục cache, khởi động squid bằng lệnh :

```
/usr/local/squid/squid -D&
```

Trong môi trường Linux, bạn không cần phải tạo cache. Khi khởi động bằng script, nó sẽ tự động tạo cache cho bạn:

```
#chkconfig squid on
#/etc/init.d/squid start/stop/restart
```



BÀI 17

Linux Security

Tóm tắt

Lý thuyết: 10 tiết - Thực hành: 10 tiết.

Mục tiêu	Các mục chính	Bài tập bắt buộc	Bài tập làm thêm
Bài học giới thiệu các công cụ hỗ trợ cách thiết lập Firewall trên môi trường Linux như: iptables, tcp_wrappers. Sử dụng iptables để thực thi các kỹ thuật NAT, Routing.	<ul style="list-style-type: none"> I. Log File II. Giới hạn user III. Network security 	Bài tập 7.1 (Linux security)	



I. Log File

Một số file log chính trong hệ thống:

- File /var/log/messages: Chứa các thông tin log của hệ thống được daemon syslogd ghi nhận.
- File /var/log/secure : chứa các thông tin về login fail, add user,...
- File /var/log/wtmp lưu các log về logon/reboot thành công vào hệ thống(ta có thể sử dụng last tool để xem thông tin này).
- File /var/run/utmp lưu các session hiện tại đang logon vào hệ thống(ta có thể dùng lệnh who, w để xem thông tin này).

II. Giới hạn user

Thông qua tập tin /etc/nologin, ta có thể ngăn chặn việc login của user trong hệ thống trừ user root.

Thư mục /etc/security/ cho phép người quản trị có thể giới hạn user CPU time, kích thước tối đa của file, số kết nối vào hệ thống(file /etc/security/limits.conf).

/etc/security/access.conf để giới hạn việc login của user và nhóm từ 1 vị trí cụ thể nào đó.

Tham khảo về cú pháp của file /etc/security/limits.conf

```
<Domain> <type> <item> <value>
```

Trong đó:

```
<domain> :username, groupname(sử dụng theo cú pháp @groupname)
```

```
<type> : hard, soft.
```

```
<item>: core, data, fsize,...(ta tham khảo file /etc/security/limits.conf)
```

III. Network security

Linux phân chia Network security thành hai loại chính:

- Loại 1: host based security
- Loại 2: port based security

III.1. Host Based security

Tcp_wrappers cung cấp host based access control list cho nhiều loại network services như: xinetd, sshd, portmap,...

Tcp_wrappers cung cấp hai file cấu hình /etc/hosts.allow và /etc/hosts.deny để ngăn chặn hoặc cho phép các host request đến các dịch vụ trong hệ thống. Cú pháp của 2 file này như sau:

Service : hosts [EXCEPT] hosts

Ví dụ:

```
ALL: ALL EXCEPT .domain.com
```



III.2. Port based security

Linux kernel cho phép thực thi chức năng packet filtering trong hệ thống thông qua công cụ iptables, ipchains.

III.2.1 Giới thiệu về iptables

Iptables do Netfilter organization viết ra để tăng tính năng bảo mật trên hệ thống Linux. Iptables cung cấp các tính năng sau:

- Tích hợp tốt với kernel của Linux.
- Có khả năng phân tích package hiệu quả.
- Lọc package dựa vào MAC và một số cờ hiệu trong TCP Header.
- Cung cấp chi tiết các tùy chọn để ghi nhận sự kiện hệ thống.
- Cung cấp kỹ thuật NAT
- Có khả năng ngăn chặn một số cơ chế tấn công theo kiểu từ chối dịch vụ(denial of service (DoS) attacks)

III.2.2 Cài đặt iptables

Iptables được cài đặt mặc định trong hệ thống Linux, package của iptables là iptables-1.2.9-1.0.i386.rpm, ta có thể dùng lệnh rpm để cài đặt package này:

```
Rpm -ivh iptables-1.2.9-1.0.i386.rpm
```

Khởi động iptables và xác định trạng thái của iptables

Cho phép iptables start vào thời điểm hệ thống khởi động:

```
#chkconfig iptables on
start/stop/restart dịch vụ DNS:
#service iptables restart
Xác định trạng thái của iptables
#service iptables status
```

III.2.3 Cơ chế xử lý package trong iptables

Iptables sẽ kiểm tra tất cả các package khi nó đi qua iptables host, quá trình kiểm tra này được thực hiện một cách tuần tự từ entries đầu tiên đến entry cuối cùng.

Có ba loại bảng trong iptables:

- **Mangle table:** chịu trách nhiệm biến đổi quality of service bits trong TCP header. Thông thường loại table này được ứng dụng trong SOHO.
- **Filter queue:** chịu trách nhiệm thiết lập bộ lọc packet(packet filtering), có ba loại built-in chains được mô tả để thực hiện các chính sách về firewall (firewall policy rules).
 - + Forward chain: Lọc packets đi qua firewall.
 - + Input chain: Lọc packets đi vào firewall.
 - + Output chain: Lọc packets đi ra firewall.
- **NAT queue:** thực thi chức năng NAT, cung cấp hai loại build-in chains sau đây:

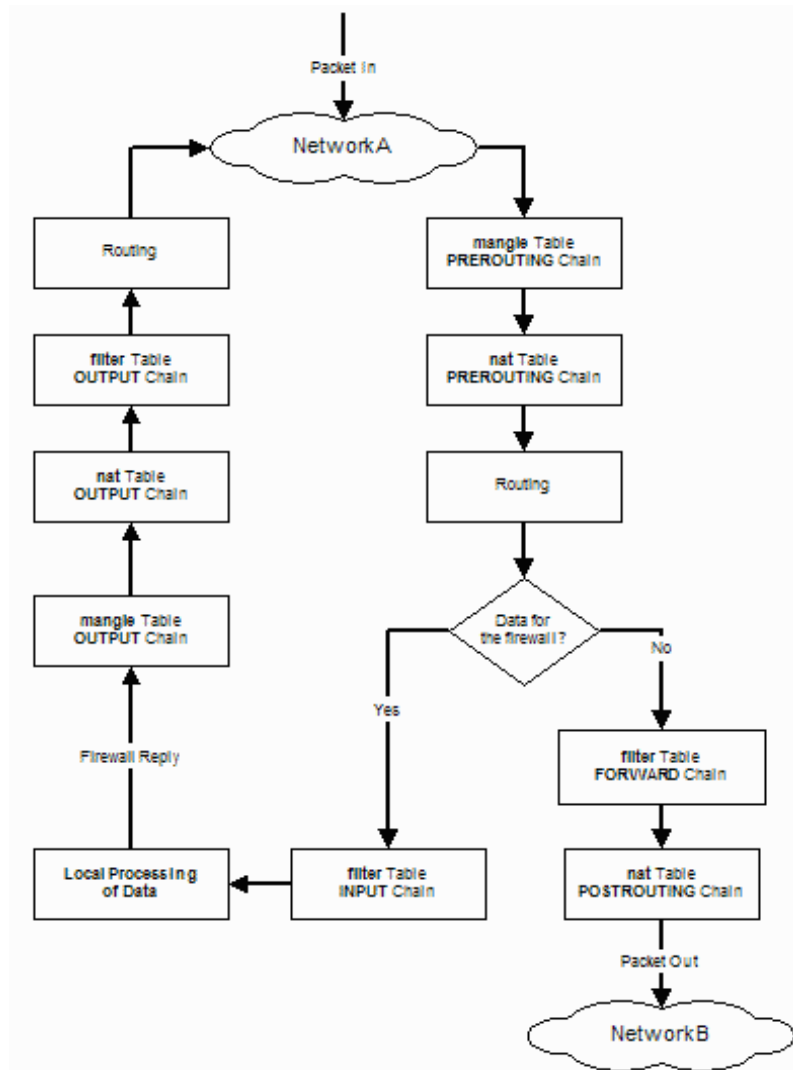


- + Pre-routing chain: NATs packets khi destination address của packet cần thay đổi (NAT từ ngoài vào trong nội bộ).
- + Post-routing chain: NATs packets khi source address của packet cần thay đổi(NAT từ trong ra ngoài)

Loại hàng đợi (Queue Type)	Chức năng của hàng đợi (Queue Function)	Thay đổi packet trong hàng đợi (Packet transformation chain in Queue)	Chức năng của Chain(Chain Function)
Filter	Packet filtering	FORWARD	Cho phép packet chuyển qua firewall (Filters packets to servers accessible by another NIC on the firewall)
		INPUT	Filters packets cho những gói tin đi vào firewall (destined to the firewall)
		OUTPUT	Filters packets cho những gói tin đi ra firewall (originating from the firewall)
Nat	Network Address Translation	PREROUTING	Quá trình NAT sẽ thực hiện trước khi thực thi cơ chế routing. Điều này thuận lợi trong việc thay đổi địa chỉ đích(NAT trong ra ngoài) để địa chỉ đích cơ thể tương thích với bảng định tuyến của firewall, khi cấu hình ta có thể dùng từ khoá DNAT để mô tả cho kỹ thuật này.
		POSTROUTING	Quá trình NAT sẽ thực hiện sau quá trình định tuyến. quá trình này ngụ ý rằng ta không cần thay đổi địa chỉ đích của packet, ta chỉ cần thay đổi địa chỉ nguồn của packet. Kỹ thuật này được gọi là NAT one-to-one hoặc many-to-one. (được gọi là source NAT, hoặc SNAT)



Loại hàng đợi (Queue Type)	Chức năng của hàng đợi (Queue Function)	Thay đổi packet trong hàng đợi (Packet transformation chain in Queue)	Chức năng của Chain(Chain Function)
		OUTPUT	Trong loại này firewall thực hiện quá trình NAT
Mangle	Thay đổi TCP header	PREROUTING POSTROUTING OUTPUT INPUT FORWARD	Thay đổi quality of service bits của TCP Header.



Sơ đồ lưu chuyển packet trong iptables

III.2.4 Targets và Jumps

Targets là cơ chế hoạt động trong iptables dùng để nhận diện và kiểm tra packet.

Jump là cơ chế chuyển một packet đến một target nào đó để xử lý thêm một số thao tác khác. Danh sách các target được xây dựng sẵn trong iptables:

Target	Mô tả	Những tùy chọn thông dụng
ACCEPT	iptables chấp nhận chuyển data đến đích.	
DROP	iptables block packet.	



Target	Mô tả	Những tùy chọn thông dụng
LOG	<p>Thông tin của packet sẽ gửi vào syslog daemon</p> <p>iptables tiếp tục xử lý luật tiếp theo trong bảng mô tả luật.</p> <p>Nếu luật cuối cùng không match thì sẽ drop packet.</p>	<p>--log-prefix "string"</p> <p>(iptables sẽ ghi nhận lại những messages bắt đầu bằng chuỗi "string").</p>
REJECT	<p>Ngăn chặn packet và gửi thông báo cho sender.</p>	<p>--reject-with qualifier</p> <p>(qualifier chỉ định loại reject message sẽ được gửi lại cho người gửi. các loại Qualifiers sau:</p> <ul style="list-style-type: none"> icmp-port-unreachable (default) icmp-net-unreachable icmp-host-unreachable icmp-proto-unreachable icmp-net-prohibited icmp-host-prohibited tcp-reset echo-reply
DNAT	<p>Thay đổi địa chỉ đích của packet (rewriting the destination IP address of the packet)</p>	<p>--to-destination ipaddress</p> <p>(iptables sẽ thay thế địa chỉ đích bằng địa chỉ ipaddress)</p>
SNAT	<p>Thay đổi địa chỉ nguồn của packet</p>	<p>--to-source <address>[-<address>][:<port>-<port>]</p> <p>(Chỉ định địa chỉ nguồn và port nguồn sẽ được sử dụng)</p>



Target	Mô tả	Những tùy chọn thông dụng
MASQUERADE	Được sử dụng để thực hiện kỹ thuật NAT (giả mạo địa chỉ nguồn với địa chỉ của firewall's interface)	[--to-ports <port>[-<port>]] (Chỉ định dãy port nguồn ánh xạ với dãy port ban đầu)

III.2.5 Thực thi lệnh trong iptables

Bảng mô tả về iptables command:

Iptables command	Mô tả(Description)
Switch	
-t <table>	Chỉ định bảng cho iptables bao gồm: filter, nat, mangle tables.
-j <target>	nhảy đến một target chain khi packet thoả(phù hợp) luật hiện tại.
-A	Đưa luật vào cuối iptables chain.
-F	Xoá tất cả các luật trong bảng lựa chọn
-p <protocol-type>	Mô tả các protocol bao gồm: icmp, tcp, udp, and all
-s <ip-address>	Chỉ định source IP address
-d <ip-address>	Chỉ định destination IP address
-i <interface-name>	Chỉ định "input" interface nhận packet.
-o <interface-name>	Chỉ định "output" interface.



Ví dụ:

Firewall chấp nhận cho bất kỳ TCP packet đi vào interface eth0 đến địa chỉ 192.168.1.1

```
iptables -A INPUT -s 0/0 -i eth0 -d 192.168.1.1 -p TCP -j ACCEPT
```

Đặt Firewall cho TCP packet ta tham khảo bảng mô tả sau:

Khoá chuyển(Switch)	Mô tả(Description)
-p tcp --sport <port>	TCP source port: Có thể chỉ định một giá trị hoặc một dãy giá trị theo định dạng: start-port-number:end-port-number
-p tcp --dport <port>	TCP destination port Có thể chỉ định một giá trị hoặc một dãy giá trị theo định dạng: starting-port:ending-port
-p tcp --syn	Nhận diện TCP connection request mới ! --syn không phải tcp connection request mới.
-p udp --sport <port>	UDP source port Có thể chỉ định một giá trị hoặc một dãy giá trị theo định dạng: starting-port:ending-port
-p udp --dport <port>	UDP destination port Có thể chỉ định một giá trị hoặc một dãy giá trị theo định dạng: starting-port:ending-port

Ví dụ:

Firewall chấp nhận TCP packet được định tuyến khi nó đi vào interface eth0 và đi ra interface eth1 để đến đích 192.168.1.58 với port nguồn bắt đầu từ 1024 tới 65535 và port đích 80.

```
iptables -A FORWARD -s 0/0 -i eth0 -d 192.168.1.58 -o eth1 -p TCP
```




```
--sport 1024:65535 --dport 80 -j ACCEPT
```

Đặt Firewall cho ICMP packet ta tham khảo bảng mô tả sau:

--icmp-type	Mô tả
--icmp-type <type>	Mô tả hai loại echo-reply và echo-request

Ví dụ: Firewall cho phép gửi icmp echo-request và icmp echo-reply.

```
iptables -A OUTPUT -p icmp --icmp-type echo-request -j ACCEPT
```

```
iptables -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT
```

Ví dụ:

Chỉ định số lượng yêu cầu phù hợp cho 1 đơn vị thời gian theo định dạng(/second, /minute, /hour, /day)

```
iptables -A INPUT -p icmp --icmp-type echo-request -m limit --limit 1/s -i eth0 -j ACCEPT
```

Đặc điểm giới hạn số lượng connection này ta sẽ chống được các cơ chế tấn công theo kiểu như SYN flood attacks và một số loại tấn công theo kiểu tấn công **denial of service attack**. Một số thông số mở rộng khi mô tả luật:

Khoá chuyển(switch)	Mô tả
-m multiport --sport <port, port>	Mô tả nhiều dãy sport phải cách nhau bằng dấu “,” và dùng tùy chọn -m
-m multiport --dport <port, port>	Mô tả nhiều dãy dport phải cách nhau bằng dấu “,” và dùng tùy chọn -m
-m multiport --ports <port, port>	Mô tả dãy port phải cách nhau bằng dấu “,” và dùng tùy chọn -m
-m --state <state>	kiểm tra trạng thái: ESTABLISHED: đã thiết lập connection NEW: bắt đầu thiết lập connection RELATED: thiết lập connection thứ hai(FTP data transfer, hoặc ICMP error)

Ví dụ:

Firewall chấp nhận TCP packet(mô tả trong dòng 1) từ bất kỳ địa chỉ nào đi vào interface eth0 đến địa chỉ 192.168.1.58 qua interface eth1, source port từ 1024 tới 65535 và dest port là 80 và 443. Packet trả về(mô tả trong dòng 2) cũng được chấp nhận từ 192.168.1.58

```
iptables -A FORWARD -s 0/0 -i eth0 -d 192.168.1.58 -o eth1 -p TCP --sport 1024:65535 -m multiport --dport 80,443 -j ACCEPT
```

```
iptables -A FORWARD -d 0/0 -o eth0 -s 192.168.1.58 -i eth1 -p TCP -m state --state ESTABLISHED -j ACCEPT
```



III.2.6 Sử dụng Chain tự định nghĩa

Thay vì sử dụng các chain đã được xây dựng sẵn trong iptables, ta có thể sử dụng User Defined chains để định nghĩa một chain name mô tả cho tất cả protocol-type cho packet. Ta có thể dùng User Defined Chains thay thế chain dài dòng bằng cách sử dụng chain chính chỉ đến nhiều chain con.

Ví dụ:

```
iptables -A INPUT -i eth0 -d 206.229.110.2 -j fast-input-queue
iptables -A OUTPUT -o eth0 -s 206.229.110.2 -j fast-output-queue
```

```
iptables -A fast-input-queue -p icmp -j icmp-queue-in
iptables -A fast-output-queue -p icmp -j icmp-queue-out
```

```
iptables -A icmp-queue-out -p icmp --icmp-type echo-request
-m state --state NEW -j ACCEPT
iptables -A icmp-queue-in -p icmp --icmp-type echo-reply -j ACCEPT
```

Chain	Mô tả
INPUT	Xây dựng INPUT chain trong iptables
OUTPUT	Xây dựng OUTPUT chain trong iptables
fast-input-queue	Input chain nhận diện các giao thức và chuyển packet đến protocol trong chain
fast-output-queue	Output chain nhận diện các giao thức và chuyển packet đến protocol trong chain
icmp-queue-out	Output cho ICMP
icmp-queue-in	Input cho ICMP

III.2.7 Lưu iptables script

Lệnh **service iptables save** để lưu trữ cấu hình iptables trong file /etc/sysconfig/iptables khi ta khởi động lại hệ thống thì chương trình iptables-restore sẽ đọc file script này và kích hoạt lại thông tin cấu hình. Định dạng của file này như sau:

```
# Generated by iptables-save v1.2.9 on Mon Nov 8 11:00:07 2004
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [144:12748]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
```



```
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -p icmp -m icmp --icmp-type 255 -j ACCEPT
-A RH-Firewall-1-INPUT -p esp -j ACCEPT
-A RH-Firewall-1-INPUT -p ah -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Mon Nov 8 11:00:07 2004
```

Trong Fedora ta có thể dùng lệnh sau để lưu script file cho iptables, #lokkit lưu cấu hình iptables firewall vào trong file /etc/sysconfig/iptables

III.2.8 Phục hồi script khi mất script file.

Ta có thể thực hiện các lệnh sau đây để phục hồi script

```
# iptables-save > firewall-config
# cat firewall-config
# Generated by iptables-save v1.2.9 on Mon Nov 8 11:00:07 2004
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [144:12748]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -p icmp -m icmp --icmp-type 255 -j ACCEPT
-A RH-Firewall-1-INPUT -p esp -j ACCEPT
-A RH-Firewall-1-INPUT -p ah -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Mon Nov 8 11:00:07 2004
```

Sau đó sửa file firewall-config file, và nạp lại iptables thông qua lệnh iptables-restore

```
# iptables-restore < firewall-config
```

Cuối cùng ta dùng lệnh:

```
# service iptables save
```



III.2.9 Load kernel module cần cho iptables

Ứng dụng iptables yêu cầu load một số module sau:

- iptable_nat module cho NAT.
- ip_conntrack_ftp module cần cho FTP support
- ip_conntrack module để theo dõi trạng thái của TCP connection.
- ip_nat_ftp module cần cho việc load FTP servers sau NAT firewall.

Nếu /etc/sysconfig/iptables file không hỗ trợ load các module thì ta sẽ thêm các mô tả (statement) sau vào /etc/rc.local file để chạy chúng sau mỗi lần khởi động lại hệ thống.

```
# File: /etc/rc.local
# Module to track the state of connections
modprobe ip_conntrack
# Load the iptables active FTP module, requires ip_conntrack
modprobe ip_conntrack_ftp
# Load iptables NAT module when required
modprobe iptable_nat
# Module required for active an FTP server using NAT
modprobe ip_nat_ftp
```

III.2.10 Một số ví dụ về firewall

Ví dụ 1:

Cho phép truy xuất DNS đến Firewall:

```
iptables -A OUTPUT -p udp -o eth0 --dport 53 --sport 1024:65535 -j ACCEPT
iptables -A INPUT -p udp -i eth0 --sport 53 --dport 1024:65535 -j ACCEPT
```

Ví dụ 2:

Cho phép WWW và SSH truy xuất tới Firewall

```
#-----
# Allow previously established connections
# - Interface eth0 is the internet interface
#-----iptables -A OUTPUT -o eth0 -m state --state
ESTABLISHED,RELATED -j ACCEPT
#-----
# Allow port 80 (www) and 22 (SSH) connections to the firewall
#-----
iptables -A INPUT -p tcp -i eth0 --dport 22 --sport 1024:65535 -m state --state NEW -j ACCEPT

iptables -A INPUT -p tcp -i eth0 --dport 80 --sport 1024:65535 -m state --state NEW -j ACCEPT
```

Ví dụ 3: Cho phép Firewall truy xuất Internet

```
#-----
# Allow port 80 (www) and 443 (https) connections from the firewall
#-----
```



```
iptables -A OUTPUT -j ACCEPT -m state --state NEW,ESTABLISHED,RELATED -o eth0 -p tcp -m
multiport --dport 80,443 -m multiport --sport 1024:65535
```

```
#-----
# Allow previously established connections
# - Interface eth0 is the internet interface
#-----
iptables -A INPUT -j ACCEPT -m state --state ESTABLISHED,RELATED -i eth0 -p tcp
```

Nếu ta muốn tất cả các TCP traffic bắt đầu từ Firewall được chấp nhận thì ta bỏ dòng:

```
-m multiport --dport 80,443 -m multiport --sport 1024:65535
```

Ví dụ 4: Cho phép mạng nội bộ truy xuất tới Firewall

```
# Allow all bidirectional traffic from your firewall to the
# protected network
# - Interface eth1 is the private network interface
#-----
iptables -A INPUT -j ACCEPT -p all -s 192.168.1.0/24 -i eth1
iptables -A OUTPUT -j ACCEPT -p all -d 192.168.1.0/24 -o eth1
```

III.2.11 Khắc phục sự cố trên iptables

Kiểm tra Firewall Logs

Firewall Logs được ghi nhận vào /var/log/messages file

Để cho phép iptables ghi log vào /var/log/messages ta phải cấu hình như sau:

```
#-----
# Log and drop all other packets to file /var/log/messages
# Without this we could be crawling around in the dark
#-----

iptables -A OUTPUT -j LOG
iptables -A INPUT -j LOG
iptables -A FORWARD -j LOG

iptables -A OUTPUT -j DROP
iptables -A INPUT -j DROP
iptables -A FORWARD -j DROP
```

Sau đây là một số ví dụ về log output file

```
Firewall denies replies to DNS queries (UDP port 53) đến server 192.168.1.102 trên home network.
Feb 23 20:33:50 bigboy kernel: IN=wlan0 OUT= MAC=00:06:25:09:69:80:00:a0:c5:e1:3e:88:08:00
SRC=192.42.93.30 DST=192.168.1.102 LEN=220 TOS=0x00 PREC=0x00 TTL=54 ID=30485
PROTO=UDP SPT=53 DPT=32820 LEN=200
```

Firewall denies Windows NetBIOS traffic (UDP port 138)



```
Feb 23 20:43:08 bigboy kernel: IN=wlan0 OUT= MAC=ff:ff:ff:ff:ff:ff:00:06:25:09:6a:b5:08:00
SRC=192.168.1.100 DST=192.168.1.255 LEN=241 TOS=0x00 PREC=0x00 TTL=64 ID=0 DF
PROTO=UDP SPT=138 DPT=138 LEN=221
```

Firewall denies Network Time Protocol (NTP UDP port 123)

```
Feb 23 20:58:48 bigboy kernel: IN= OUT=wlan0 SRC=192.168.1.102 DST=207.200.81.113 LEN=76
TOS=0x10 PREC=0x00 TTL=64 ID=0 DF PROTO=UDP SPT=123 DPT=123 LEN=56
```

III.2.12 iptables không khởi động

Khi ta khởi động iptables thì ta dùng lệnh `/etc/init.d/iptables start`, lúc này iptables gọi iptables startup script trong file `/etc/sysconfig/iptables`. Do đó nếu file này không tồn tại hoặc bị lỗi thì iptables có thể không hoạt động được.

Khi ta thay đổi cấu hình trên iptables thì ta phải dùng lệnh `service iptables save` để lưu trữ lại các thông tin cấu hình sau đó mới tiến hành restart lại iptables script file.

Ví dụ:

```
# service iptables start
# touch /etc/sysconfig/iptables
# chmod 600 /etc/sysconfig/iptables
# service iptables save
```



BÀI 18

Webmin

Tóm tắt

Lý thuyết: 10 tiết - Thực hành: 10 tiết.

Mục tiêu	Các mục chính	Bài tập bắt buộc	Bài tập làm thêm
Bài học giới thiệu các công cụ hỗ trợ thao tác làm việc và quản trị hệ thống qua Web như Webmin, Usermin	<ul style="list-style-type: none"> I. Giới thiệu Webmin I. Cài đặt Webmin II. Cấu hình Webmin 	Bài tập 8.1 (Webmin)	



I. Giới thiệu Webmin

Là ứng dụng Web hỗ trợ cho công tác quản trị hệ thống Unix/Linux qua Web, hầu hết các chương trình ứng dụng của Webmin được Jamie Cameron phát triển. Thông qua Webmin người dùng có thể logon vào hệ thống Unix/Linux để thực hiện các thao tác quản trị hệ thống một cách bình thường. Webmin cho phép người quản trị có thể:

- Tổ chức tài khoản người dùng.
- Tổ chức và cài đặt các dịch vụ như: apache, DNS, Mail, ...
- Cập nhật các thông số cấu hình cho hệ thống.
- Cấu hình mạng.
- Cấu hình hardware.
- Cấu hình Cluster.
- Thực thi lệnh trên SHELL.
- Quản trị hệ thống từ xa qua telnet/ssh.
- Quản lý hệ thống tập tin và thư mục.

II. Cài đặt Webmin

II.1. Cài đặt từ file nhị phân

Webmin được cung cấp miễn phí tại Website <http://www.webmin.com>. Ta download package `webmin-1.190-1.noarch.rpm`. sau đó thực hiện lệnh:

```
rpm -ivh webmin-1.190-1.noarch.rpm
```

Tham khảo về output sau khi cài đặt Webmin.

```
warning: webmin-1.190-1.noarch.rpm: V3 DSA signature: NOKEY,  
key ID 11f63c51  
Preparing...  
##### [100%]  
Operating system is Redhat Linux Fedora 2  
1:webmin  
##### [100%]  
Webmin install complete. You can now login to http://server:10000/  
as root with your root password.
```

II.2. Cài đặt Webmin từ file nguồn *.tar.gz

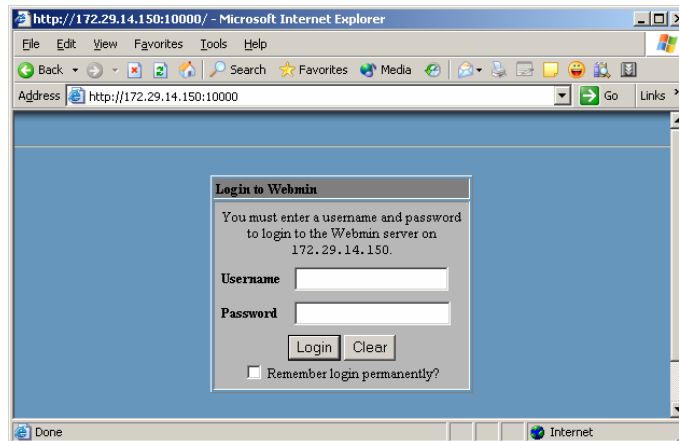
```
# tar zxvf webmin-0.87.tar.gz  
[root@delilah webmin-1.050]# ./setup.sh  
...  
Web server port (default 10000):  
Login name (default admin): root  
Login password:  
Password again:  
The Perl SSLeay library is not installed. SSL not available.  
Start Webmin at boot time (y/n): n
```

Sau khi cài đặt hoàn tất Webmin ta truy xuất Server theo địa chỉ: <http://delilah.swell:10000/>

III. Cấu hình Webmin

III.1. Đăng nhập vào Webmin Server

Sau khi cài xong Webmin ta có thể dùng Web Browser để truy xuất vào Webmin Server thông qua địa chỉ `http://server:10000/`



Màn hình đăng nhập

Nhập username : root và mật khẩu tương ứng để logon vào hệ thống



Giao diện Webmin

III.2. Cấu hình Webmin

Thay đổi mật khẩu cho Webmin Password bằng dòng lệnh:

```
#/usr/libexec/webmin/changepass.pl /etc/webmin root 123456
```

Restart Webmin bằng dòng lệnh:

```
#/etc/webmin/stop
```

```
#/etc/webmin/start
```



Tìm hiểu file cấu hình Webmin /etc/webmin/miniserv.conf cho phép ta thay đổi một số thông tin cấu hình Webmin Server

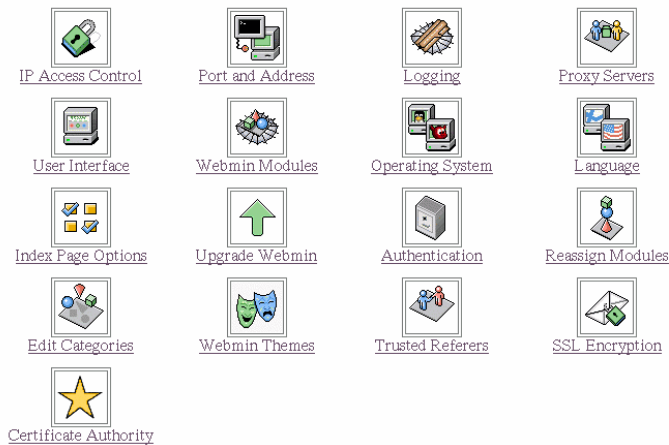
```
#chỉ định port number
port=10000
root=/usr/libexec/webmin
#chỉ định Webmin Type
mimetypes=/usr/libexec/webmin/mime.types
addtype_cgi=internal/cgi
realm=Webmin Server
#chỉ định logfile lưu trữ log cho Webmin
logfile=/var/webmin/miniserv.log
#lưu trữ error log
errorlog=/var/webmin/miniserv.error
#chỉ định pid file
pidfile=/var/webmin/miniserv.pid
logtime=168
ppath=
ssl=1
#khai báo biến môi trường lưu trữ thông tin cấu hình Webmin
env_WEBMIN_CONFIG=/etc/webmin
env_WEBMIN_VAR=/var/webmin
atboot=0
logout=/etc/webmin/logout-flag
#listen port
listen=10000
denyfile=\.pl$
log=1
blockhost_failures=5
blockhost_time=60
syslog=1
session=1
#chỉ file lưu trữ Webmin User
userfile=/etc/webmin/miniserv.users
keyfile=/etc/webmin/miniserv.pem
passwd_file=/etc/shadow
passwd_uindex=0
passwd_pindex=1
passwd_cindex=2
passwd_mindex=4
passwd_mode=0
passdelay=1
preroot=mscstyle3
```

III.3. Cấu hình Webmin qua Web Browser

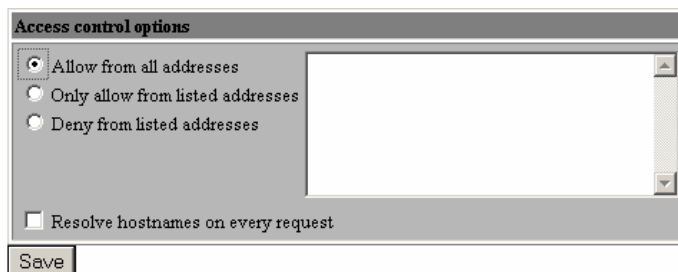
Sau khi đăng nhập vào Webmin Server ta chọn biểu tượng Webmin configuration

[Webmin Index](#)

Webmin Configuration



Cho phép hay cấm truy xuất Webmin từ host nào đó trên mạng thông qua IP Access Control.



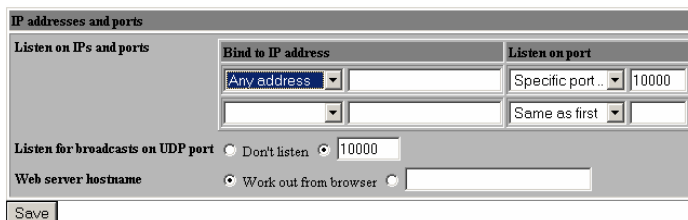
Allow from all addresses: cho phép tất cả các host khác truy xuất Webmin.

Only allow from listed addresses: Chỉ cho phép các host trong ListBox mới được sử dụng Webmin(ta có thể mô tả địa chỉ như sau 172.29.1.0/255.255.255.0 để chỉ định cho network address)

Deny from listed addresses: cho phép tất cả các host khác được truy xuất Webmin nhưng cấm các host nằm trong ListBox.

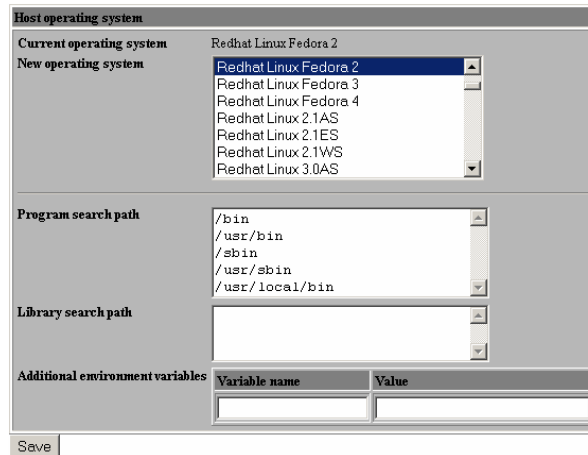
Save: Lưu trữ lại những gì ta thay đổi.

Port and Addresses: Cho phép hiệu chỉnh Webmin hoạt động trên địa chỉ IP và Port, nếu ta muốn Webmin hoạt động cổng khác thì ta có thể vào mục này để hiệu chỉnh lại cho phù hợp.



Bind to IP address và Listen on port chỉ định Webmin listen 10000 tại địa chỉ IP(mặc định Webmin listen port 10000 trên tất cả các IP của Server)

Operating System and Environment: Chỉ định loại hệ điều hành và một số đường dẫn chương trình



Host operating system

Current operating system: Redhat Linux Fedora 2

New operating system: Redhat Linux Fedora 2 (selected)

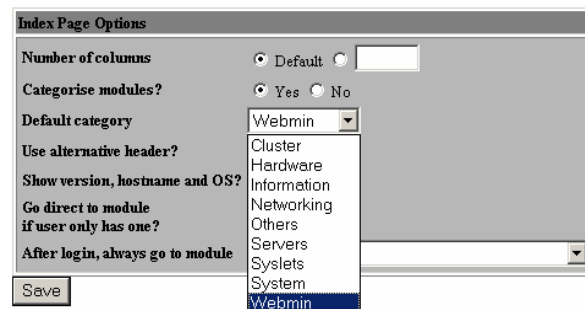
Program search path: /bin, /usr/bin, /sbin, /usr/sbin, /usr/local/bin

Library search path: (empty)

Additional environment variables: (table with Variable name and Value columns)

Save

Index Page Options: hiệu chỉnh màn hình chính của thực đơn Webmin



Index Page Options

Number of columns: Default []

Categorise modules?: Yes No

Default category: Webmin (selected)

Use alternative header? Cluster, Hardware, Information, Networking, Others, Servers, Syslets, System, Webmin

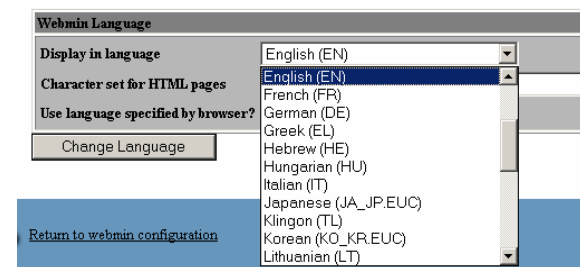
Show version, hostname and OS? (checkbox)

Go direct to module if user only has one? (checkbox)

After login, always go to module (dropdown)

Save

Chọn ngôn ngữ sử dụng cho Webmin



Webmin Language

Display in language: English (EN) (selected)

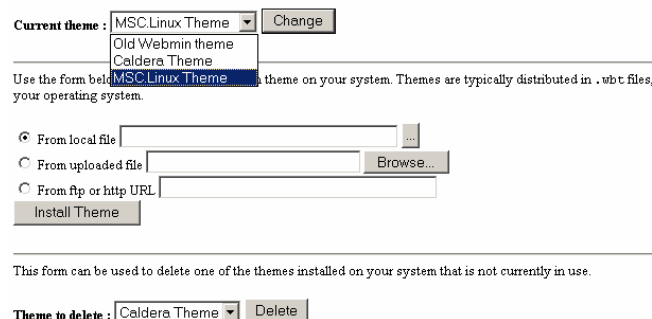
Character set for HTML pages: English (EN)

Use language specified by browser? (checkbox)

Change Language

Return to webmin configuration

Chọn Webmin Themes để hiệu chỉnh giao diện sử dụng cho Webmin như icons, colours, background, và cách trình bày Web page cho Webmin.



Current theme: MSC.Linux Theme (selected) Change

Old Webmin theme
Caldera Theme

Use the form below to install themes on your system. Themes are typically distributed in .wbz files, your operating system.

From local file [] ...

From uploaded file [] Browse...

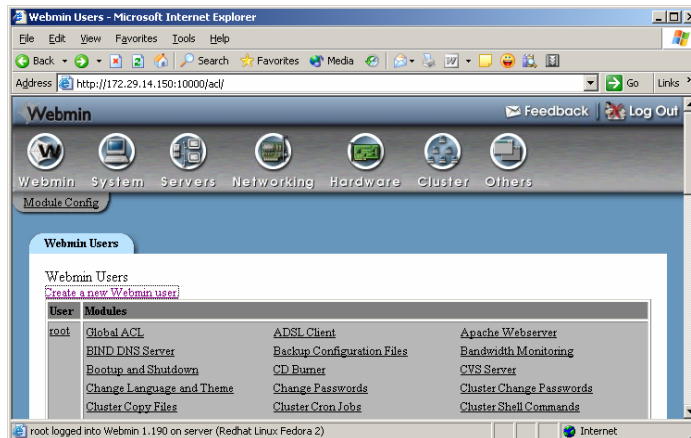
From ftp or http URL []

Install Theme

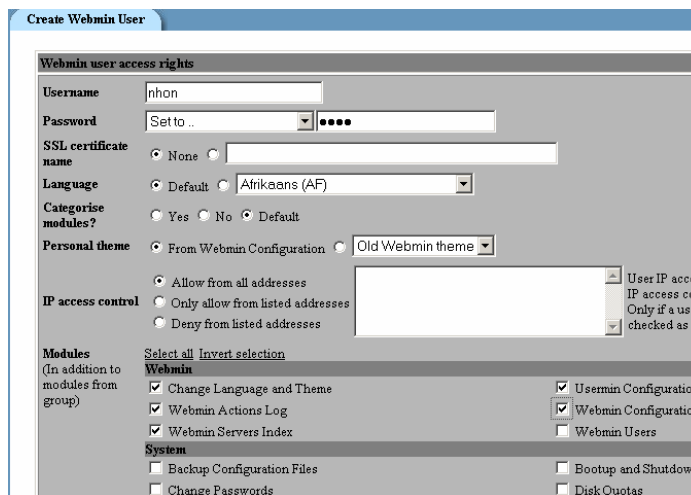
This form can be used to delete one of the themes installed on your system that is not currently in use.

Theme to delete: Caldera Theme Delete

III.4. Quản lý Webmin User



Tạo Webmin User thông qua mục Create a new Webmin user.



Ta nhập username, password, và đặt một số quyền hạn cho User....

III.5. Webmin cho Users(Usermin)

Với Webmin được sử dụng chủ yếu để quản trị hệ thống. Usermin là một công cụ cung cấp cho user có thể sử dụng hệ thống qua Web: Usermin có thể cung cấp cho user:

- sử dụng mail client qua Web(web-based mail client).
- Quản lý Java file applet.
- Cấu hình SSH configuration và client modules
- GnuPG encryption and decryption.
- Mail forwarding.
- Changing passwords
- Cron jobs
- web-based command shell
- Cài đặt Usermin:
- Cài đặt bằng file nhị phân

Trước khi cài Usermin ta phải cài Authen-PAM Perl module

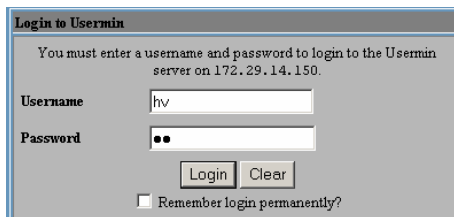
```
[root@server openwebmail]# rpm -ivh usermin-1.120-1.noarch.rpm
warning: usermin-1.120-1.noarch.rpm: V3 DSA signature:
NOKEY, key ID 11f63c51
Preparing...
##### [100%]
Operating system is Redhat Linux Fedora 2
1:usermin
##### [100%]
Usermin install complete. You can now login to
http://server:20000/
as any user on your system.
```

Cài đặt Usermin thông qua file .tar.gz

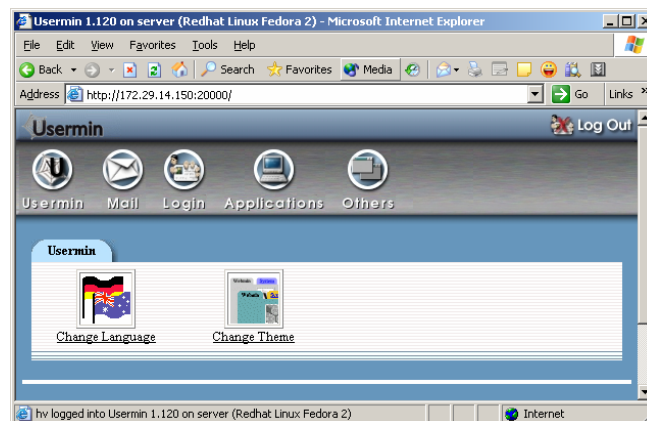
```
# cp usermin-0.6.tar.gz /usr/local
# cd /usr/local
# gunzip usermin-0.6.tar.gz
# tar xf usermin-0.6.tar
# cd usermin-0.6
# ./setup.sh
```

III.6. Sử dụng Usermin

Để login vào Usermin Server ta sử dụng địa chỉ http://server:20000/



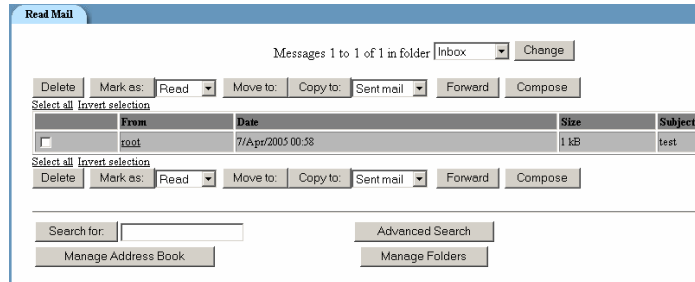
Nhập username và password để login vào hệ thống



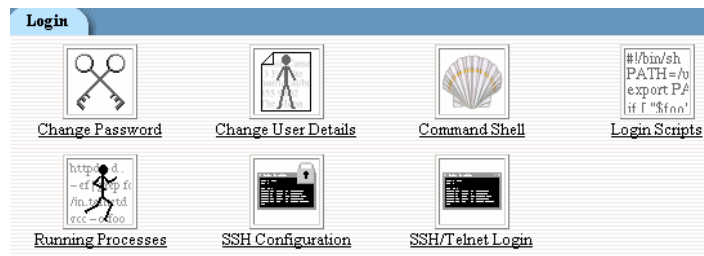


Mail : hỗ trợ các thao tác về việc sử dụng mail cho User

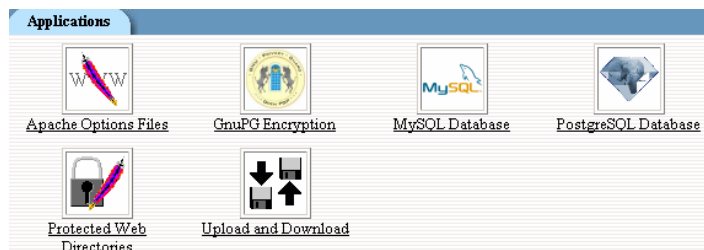
(Sau đây là ví dụ về sử dụng Usermin để đọc mail)



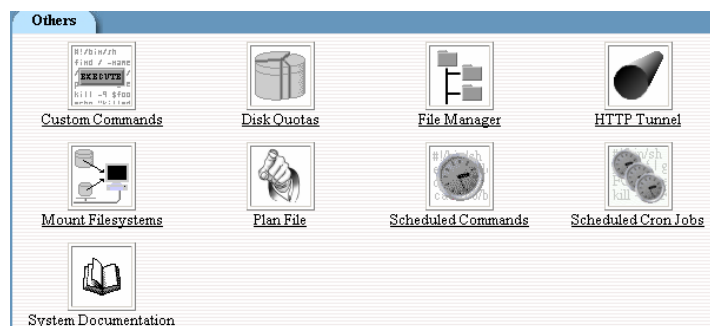
Login hỗ trợ user có thể sử dụng command shell, logon script....



Applications hỗ trợ cho user sử dụng một số ứng dụng như SQL, upload và download file...

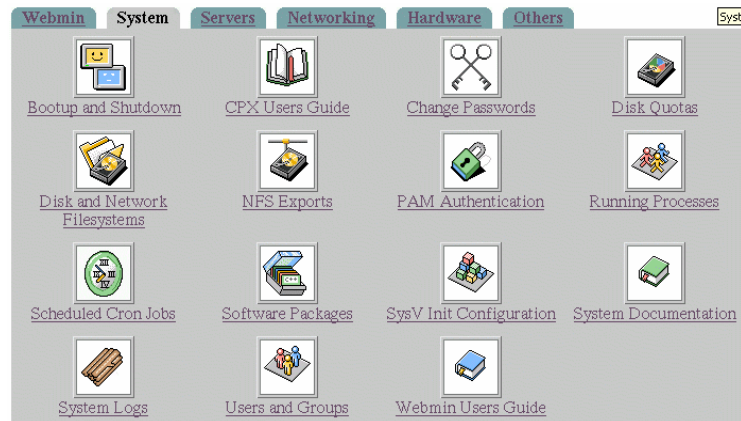


Others Hỗ trợ cho user có thể xem cấu trúc file, mount file, hiệu chỉnh lệnh,....



III.7. Cấu hình hệ thống qua Webmin

Thông qua Webmin ta có thể cấu hình các thông tin trong hệ thống như: shutdown/reboot, disk quota, NFS, User/Groups....



Trong màn hình trên là một số các biểu tượng để ta có thể sử dụng và cấu hình hệ thống tương ứng qua Webmin.

Công cụ	Chức năng
Backup Configuration Files	Hỗ trợ backup và restore thông tin cấu hình hệ thống.
Bootup and Shutdown	Cho phép hiệu chỉnh quá trình khởi động và dừng dịch vụ
Change Passwords	Thay đổi mật khẩu cho từng người dùng.
Disk Quotas	Thiết lập hạn ngạch cho người dùng.
Disk and Network Filesystems	Hỗ trợ việc mount và umount filesystem
Filesystem Backup	Backup hệ thống tập tin
LDAP Users and Groups	Quản lý LDAP user và group
Log File Rotation	Hỗ trợ việc quản lý và chuyển đổi log file.
PAM Authentication	Hỗ trợ quản lý các thông tin chứng thực cho dịch vụ hệ thống.
Running Processes	Theo dõi và quản lý các tiến trình hoạt động trong hệ thống
Scheduled Commands	Đặt lịch biểu thực thi lệnh
Scheduled Cron Jobs	Thiết lập và quản lý cron jobs
Security Sentries	Thiết lập một số thông tin bảo mật hệ thống.
Software Packages	Hỗ trợ cài đặt, nâng cấp và quản lý phần mềm.

SysV Init Configuration	Tạo một script thực thi cho từng runlevel
System Documentation	Tìm kiếm một số tài liệu trợ giúp có sẵn trong hệ thống.
System Logs	Quản lý system log file
Users and Groups	Quản lý người dùng và nhóm

III.8. Cấu hình Server và Daemon

Công cụ Server trên Webmin cho phép quản trị Server và một số ứng dụng đang chạy trong hệ thống.



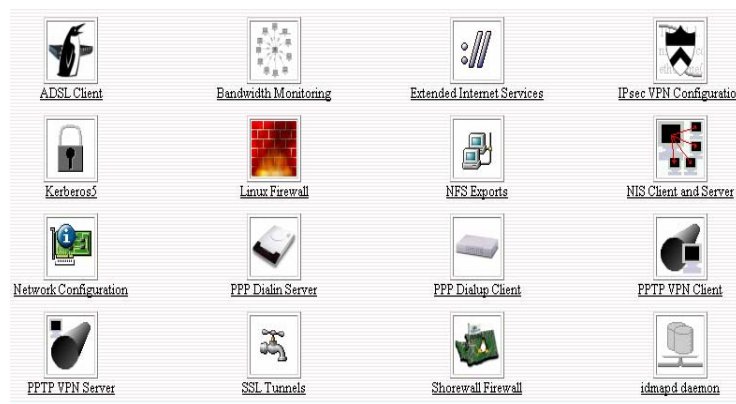
Giao diện Server trên Webmin:

Công cụ	Chức năng
Apache Webserver	Quản lý và cấu hình WebServer
BIND DNS Server	Quản lý và cấu hình DNS Server
CVS Server	Quản lý version cho hệ thống
DHCP Server	Quản lý DHCP Server
Fetchmail Mail Retrieval	Hỗ trợ việc nhận mail từ remote mail server thông qua mạng TCP/IP
Frox FTP Proxy	Cấu hình Frox FTP proxy
Jabber IM Server	Thiết lập và quản lý IM Server để hỗ trợ cho người dùng sử dụng dịch vụ Chat(one-to-one chat, multi-user chat)
Majordomo List Manager	Quản lý Internet Mailing list
MySQL Database Server	Quản lý hệ quản trị cơ sở dữ liệu MySQL.
OpenSLP Server	Cấu hình máy chủ Service Location Protocol hỗ trợ xác định sự tồn tại, vị trí và cấu hình dịch vụ mạng trong enterprise networks

Postfix Configuration	Cấu hình Postfix mail server
PostgreSQL Database Server	Cấu hình hệ quản trị cơ sở dữ liệu PostgreSQL Server
ProFTPD Server	Cấu hình FTP server sử dụng phần mềm ProFTPD Server
Procmail Mail Filter	Thiết lập bộ lọc thư cho các hệ thống mail
QMail Configuration	Cấu hình QMail Server
Read User Mail	Hỗ trợ việc đọc thư cho người dùng trong hệ thống
SSH Server	Thiết lập Server SSH để quản trị hệ thống từ xa
Samba Windows File Sharing	Quản lý SAMBA Service
Sendmail Configuration	Cấu hình Sendmail làm Mail Server
SpamAssassin Mail Filter	Thiết lập cơ chế chống spam thư
Squid Analysis Report Generator	Theo dõi và quản lý Internet connection qua Proxy
Squid Proxy Server	Cấu hình Proxy Server
WU-FTP Server	Cấu hình FTP Server sử dụng phần mềm WU-FTP
Webalizer Logfile Analysis	Quản lý và theo dõi Web log

III.9. Cấu hình mạng thông qua Webmin

Công cụ Networking trên Webmin hỗ trợ quản lý và cấu hình mạng trên hệ thống Unix/Linux.



Công cụ

ADSL Client

Bandwidth Monitoring

Chức năng

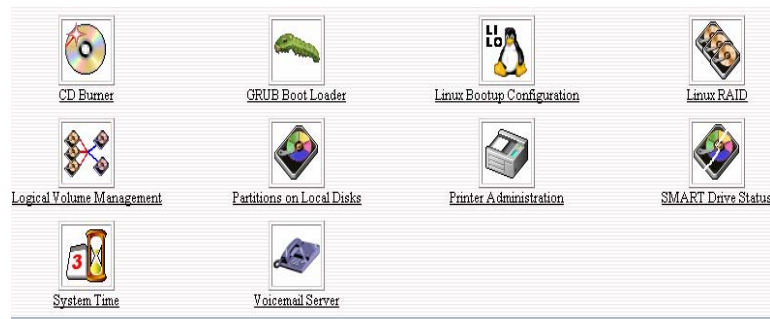
Cấu hình ADSL Client

Quản lý và theo dõi băng thông sử dụng trên hệ thống mạng.

Extended Internet Services	Quản lý và theo dõi Internet Services
IPsec VPN Configuration	Cấu hình IPsec VPN
Kerberos5	Cấu hình chứng thực Kerberos5
Linux Firewall	Cấu hình Linux Firewall dùng IPTable
NFS Exports	Export NFS Server
NIS Client and Server	Cấu hình NIS Server
Network Configuration	Cấu hình mạng(thêm card mạng, Ip address)
PPP Dialin Server	Thiết lập RAS Server
PPP Dialup Client	Thiết lập RAS Client
PPTP VPN Client	Thiết lập VPN Client
PPTP VPN Server	Thiết lập VPN Server
SSL Tunnels	Thiết lập đường ống SSL
Shorewall Firewall	Là một high-level Security tool hỗ trợ cấu hình Firewall trong hệ thống.
idmapd daemon	Cấu hình NFSV4 server và client

III.10. Cấu hình Hardware trên Webmin

Cung cấp một số công cụ hỗ trợ việc cài đặt quản lý thông tin cấu hình phần cứng trên hệ thống Unix/Linux

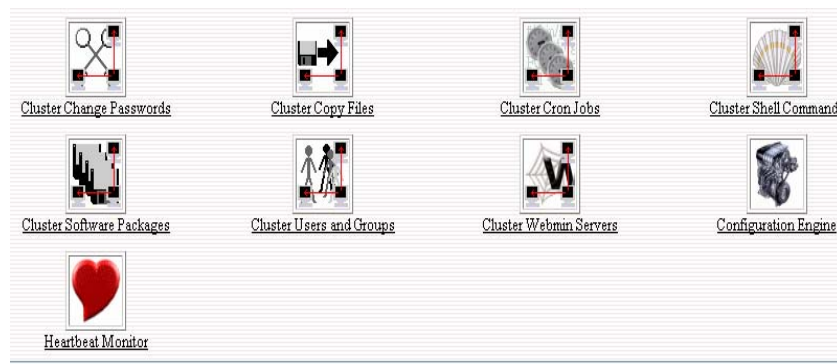


Công cụ	Chức năng
CD Burner	Hỗ trợ ghi dữ liệu vào CDROM
GRUB Boot Loader	Hiệu chỉnh và cấu hình grub loader
Linux Bootup Configuration	Cấu hình Lilo boot loader
Linux RAID	Thiết lập RAID trên Linux

Logical Volume Management	Quản lý logic Volume
Partitions on Local Disks	Quản lý các phân vùng đĩa
Printer Administration	Quản lý máy in
SMART Drive Status	Theo dõi SMART Drive
System Time	Thiết lập và quản lý timer cho hệ thống.
Voicemail Server	Thiết lập Voicemail Server

III.11. Linux Cluster trên Webmin

Clustering là một công nghệ máy chủ với khả năng chịu lỗi cao cung cấp những tính năng như: tính sẵn sàng và khả năng mở rộng. Công nghệ này nhóm các server và tài nguyên chung thành một hệ thống đơn có khả năng miễn dịch lỗi và tăng hiệu năng hoạt động. Các máy trạm tương tác với nhóm các server như thể nhóm các server này là một hệ thống đơn. Nếu một server trong nhóm bị hư, các server khác sẽ đảm trách phần việc của nó. Tham khảo các chức năng cấu hình Linux Cluster.



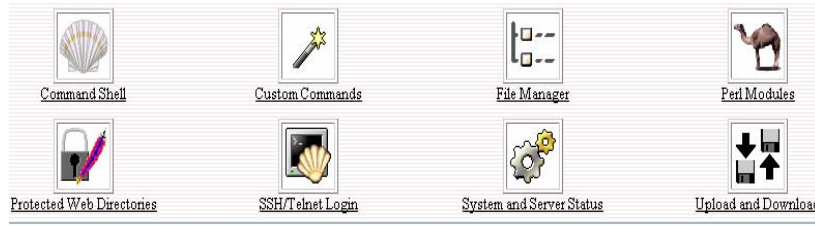
Công cụ	Chức năng
Cluster Change Passwords	Thay đổi mật khẩu trên Cluster server
Cluster Copy Files	Thực thi cơ chế sao chép file giữa các cluster server
Cluster Cron Jobs	Tạo Cron Jobs cho các cluster server.
Cluster Shell Commands	Thực thi Shell Commands trên cluster server
Cluster Software Packages	Quản lý package trên các cluster server.
Cluster Users and Groups	Quản lý User và group trên Cluster Server
Cluster Webmin Servers	Quản lý Cluster Webmin Server
Configuration Engine	Cấu hình một engine cho Cluster

Heartbeat Monitor

Theo dõi đồng bộ giữa các server.

III.12. Các thành phần khác(Others) trên Webmin

Các thành phần Others trên Webmin hỗ trợ một số tính năng như : Thực thi lệnh trên Shell, điều chỉnh lệnh, Quản lý file, bảo vệ thư mục Web,...



Công cụ	Chức năng
Command Shell	Cho phép thực thi lệnh
Custom Commands	Điều chỉnh và thêm một số lệnh mới trên Webmin
File Manager	Quản lý file
Perl Modules	Quản lý perl
Protected Web Directories	Bảo vệ thư mục Web data
SSH/Telnet Login	Login qua từ xa bằng SSH, Telnet qua Web.
System and Server Status	Quản lý và theo dõi trạng thái của Server.
Upload and Download	Cho phép upload và download file.

ĐỀ THI CUỐI HỌC PHẦN

I. Cấu trúc đề thi

<i>Môn</i>	<i>Cấu trúc đề thi</i>
Học phần IV: <ul style="list-style-type: none"> ▪ Hệ điều hành Linux - Dịch vụ mạng Linux 	<ul style="list-style-type: none"> ▪ Đề thi lý thuyết <ul style="list-style-type: none"> - Thời gian : 60 phút. - Điểm tối đa : 4/10 điểm. - Hình thức thi : Trắc Nghiệm - Tổng số câu : 45 Câu. - Điểm số chia đều cho mỗi câu: $4/45 = 0.089$ - Tham khảo tài liệu : Thí sinh không được tham khảo tài liệu. ➤ Nội dung bao gồm các phần sau: <ol style="list-style-type: none"> I. Giới thiệu về Linux II. Cài đặt Linux III. Quản lý hệ thống tập tin IV. Cài đặt phần mềm V. Những lệnh và tiện ích VI. Quản lý user, group và bảo mật VII. Quản lý tài nguyên đĩa cứng VIII. Kết nối mạng IX. NFS X. Samba XI. Những công cụ lập trình và shell script XII. Tiến trình XIII. DNS và BIND XIV. FTP Server - Vsftpd XV. Web server - Apache XVI. Mail Server – Sendmail XVII. Proxy Server - Squid XVIII. Linux security XIX. Webmin ▪ Thực hành <ul style="list-style-type: none"> - Thời gian : 120 phút. - Điểm tối đa : 6/10 điểm. - Hình thức thi : Thực hành trực tiếp trên máy.



- Tham khảo tài liệu : Thí sinh không được tham khảo tài liệu.

➤ **Nội dung bao gồm các phần sau:**

- Câu 1 (0.5 điểm): có nội dung liên quan đến hệ thống tập tin, hay những lệnh và tiện ích, hay kết nối mạng, hay cài đặt phần mềm, hay tiến trình, nfs.
- Câu 2 (0.5 điểm): có nội dung liên quan đến việc quản lý user và group.
- Câu 3 (0.5 điểm): có nội dung liên quan đến việc quyền hạn.
- Câu 4 (1 điểm): có nội dung liên quan đến quản lý tài nguyên đĩa cứng hay tiến trình hay samba.
- Câu 5 (1 điểm): lập trình shell.
- Câu 6 (2,5 điểm): có nội dung là 1 trong những trường hợp sau:
 - + Dịch vụ DNS + Web server
 - + Dịch vụ DNS + FTP server
 - + Dịch vụ DNS + mail server
 - + Dịch vụ DNS + proxy server
 - + Linux Security



II. Đề thi mẫu

II.1. Đề thi mẫu cuối môn - Hệ Điều Hành Linux

Đề Thi :

Hệ Điều Hành Linux
 Thời gian: 120 phút
 Ngày thi :/...../.....
 (Học viên không được sử dụng tài liệu)

Câu 1(1.đ)

Cấu hình hệ thống theo yêu cầu sau:

- Tên máy tính: ServerXX
- Địa chỉ IP: 172.168.10.100+XX
- Subnet Mask: 255.255.255.0
- Chỉ cài những phần mềm cần thiết.

Câu 2(1.5đ):

- a) Xem trong hệ thống phần mềm Sendmail hay không? Nếu không hãy cài đặt phần mềm này. Sau đó cho biết vị trí tất cả các tập tin của phần mềm sendmail(lưu trữ vào tập tin /root/sendmailfile)
- b) Tạo một tập tin ipaddress, nội dung tập tin này chỉ ra các cách cấu hình mạng trên linux, dùng xem bảng routing table và ghi kết quả vào cuối tập tin này.
- c) Tạo file backup *.tar cho thư mục /etc, sau đó nén tập tin backup này thành file *.tar.gz lưu trong /home, trong thư mục /home phục hồi tập tin nén trên.

Câu 3(2đ)

- a) Tạo user và group theo yêu cầu:
 - Group admins gồm các user admin, admin1, admin2.
 - Group hocvien gồm các user hv01, hv02.
- b) Cấp quyền cho những user trong group admins có quyền quản trị hệ thống tương đương với user root.
- c) Tạo thư mục /home/data. Sau đó cấp quyền cho tất cả những user chỉ có quyền read trên thư mục này, riêng những user trong group hocvien có quyền read, write và execute.

Câu 4(1đ)

Câu hình Secondary IP address có địa chỉ IP: 192.168.10.100+XX/24 cho card mạng, xem cấu hình card mạng và xuất vào tập tin /root/SECIP.

Câu 5(1đ)

Viết chương trình kiểm tra(thường xuyên) file nào đó nằm trong thư mục /var/log/, nếu dung lượng của nó lớn hơn 10Mbyte thì xoá đi, nếu nó lớn hơn 5M thì nén file này lại thành file *.gz.

Câu 6(2đ)

- a) Cài đặt và cấu hình dịch vụ Samba, chia sẻ tài nguyên /usr/soft cho group "hocviens" có quyền read, các user trong nhóm admins có toàn quyền truy cập tài nguyên này.
- b) Không cho phép những máy trong đường mạng 172.168.11.0 truy cập tài nguyên này.
- c) Vì dung lượng đĩa cứng trên server có hạn cho nên Anh, Chị hãy giới hạn mỗi user chỉ được quyền lưu trữ tài nguyên trên server là 5M.

Câu 7(1.5đ)

Dùng một trình tiện ích thích hợp có sẵn trên linux để thực hiện công việc sau:

- a) Xoá một filesystem /thu có sẵn trong hệ thống
- b) Tạo mới một filesystem với dung lượng 1000M.
- c) Định dạng filesystem này theo kiểu ext3 hoặc Linux.



- d) Kết buộc tự động (auto mount) vào mount point /soft để cho phép người dùng có thể sử dụng filesystem này khi logon vào hệ thống.
- e) xem trạng thái của các filesystem trong hệ thống, sau đó kết xuất vào file /root/filesystem.

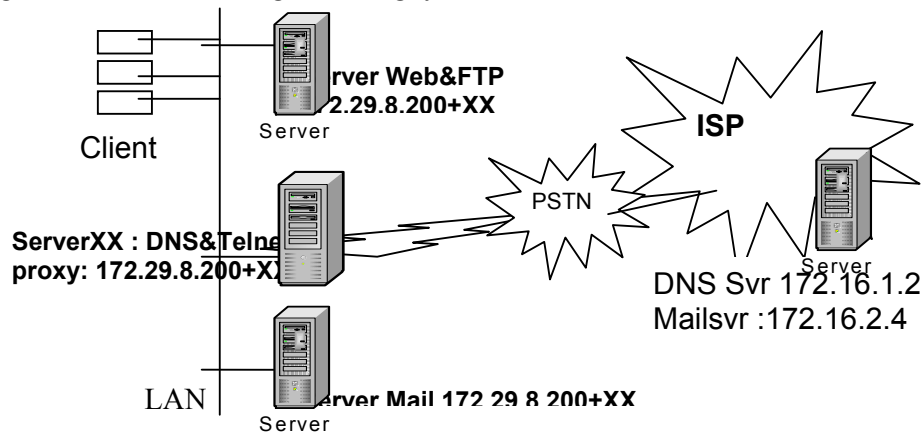
II.2. Đề thi cuối môn - Dịch Vụ Mạng Linux

Đề thi:

Môn : Linux Nâng cao
 Thời gian: 120 Phút
 (học viên không được sử dụng tài liệu)

XX là số máy đang ngồi

Biết rằng mô hình kết nối mạng của Công ty netXX như sau:



Địa chỉ đường mạng trong LAN: 172.29.8.0/24

Tên domain: netXX.com.

Yêu cầu

Cài đặt RH_LINUX và các PM sau: BIND, Apache, Samba, Sendmail.

- 1) Viết chương trình cho biết tổng số kết nối đang lắng nghe(Listen) tại máy cục bộ và tổng số kết nối đang ESTABLISHED tại port number 8080(1đ).
- 2) Cấu hình DNS server cho miền netXX.com với các yêu cầu sau(2đ):
 - a) Primary name server là serverXX, các máy tính cục bộ có thể phân giải tên miền ra ngoài internet.
 - b) Hãy tạo các record cần thiết (SOA, NS, A, CNAME, PTR) cho các server theo như sơ đồ trên trong các zone file netXX.com và 8.29.172.in-addr.arpa dùng cho phân giải tên thuận và nghịch.
 - c) Tạo MX để chuyển mail cho domain chính.
 - d) Mô tả slave zone cho domain bên cạnh.
- 3) Tổ chức Web server cho netXX.com với các yêu cầu sau(2đ):
 - a) Tạo thư mục /data/www. Đặt thư mục gốc của Web server là thư mục này. Tạo file HTML index.html trong thư mục gốc của server Web chính giới thiệu về netXX.com. Đặt trang index.html là trang chủ của Web site và tạo liên kết từ index.html trở đến dichvu.html (tạo ở dưới). Dùng Web browser phù hợp truy cập vào để kiểm tra địa chỉ webserver cho netXX.com như: http://www.netXX.com
 - b) Tạo thư mục /webdata và tạo bí danh (Alias) /data trở đến thư mục /data/webdata. Vì đây là thư mục chứa nhiều tài liệu bảo mật cho nên A/C chỉ cho phép user net mới có quyền truy cập vào tài nguyên này.
 - c) Tạo WebHosting (sử dụng NameBaseVirtualHost) cho hai địa chỉ www.tma.netXX.com và www.psv.netXX.com biết rằng /webtma là webroot của www.tma.netXX.com và thư mục /webpsv là webroot của www.psv.netXX.com.
- 4) Tổ chức mail server cho netXX.com với các yêu cầu sau(2đ):
 - a) Cấu hình nhận kết nối từ mọi địa chỉ IP



- i) Domain cục bộ: netXX.com
- ii) Máy chủ smart – host(mailgw) : 172.29.8.2
- iii) Kích thước message tối đa cho phép : 3000KByte
- b) Tạo các user các account mail theo các yêu cầu sau:
 - ketoan(Nguyễn Văn Nguyên, Lê Thanh Tông, Trần Thị Thuý Trang).
 - giamdoc(Võ Thị Thanh Thủy, Đỗ Thế Phong).
 - vanphong(Văn Thành Nhân, Nguyễn Thị Mỹ Lệ).
 - Kinhdoanh(Võ Thị Mỹ Yên, Nguyễn Hoàng Nhã Nguyễn Bá Phong, Đỗ Thị Phụng).
 - tiepthi(Võ Thị Be Thuý).
 - nhanvien(ketoan, vanphong, kinh doanh, tiepthi).
 - everyone(nhanvien, giamdoc).
- c) Cấu hình mail offline cho miền “gnt.netXX.com”, biết rằng account(user: usernet, password: net) chịu trách nhiệm nhận mail cho miền “gnt.netXX.com” này.
- d) Trong quá quản lý dịch vụ mail ta thấy rằng email: netuser@yahoo.com gửi vào server mail có chứa nhiều virus. Bạn hãy ngăn địa chỉ mail này.
- 5) Tổ chức proxy server cho hệ thống cục bộ với các yêu cầu sau(2đ):
 - a) Cấu hình nhận kết nối http từ cổng 8080 và kết nối icp từ cổng 8082.
 - b) Đặt cấu hình sao cho các máy tính trong lớp mạng 172.29.8.0/255.255.255.0 được truy cập Internet.
 - c) Khai báo proxy ngang hàng cho với máy có địa chỉ IP là 172.29.8.220 và proxy cha là 172.29.8.2.
 - d) Cho phép kết nối máy trạm chỉ kết nối 10 connection.
 - e) Cấm các user truy cập vào các địa chỉ thuộc domain yahoo.com và hackers.net.
 - f) Chỉ cho phép các host cục bộ sử dụng mạng trong giờ hành chính.



ĐỀ THI CUỐI HỌC PHẦN

I. Mẫu Đề thi lý thuyết

ĐỀ THI CUỐI HỌC PHẦN IV PHẦN LÝ THUYẾT Thời gian: 60 phút

(Học viên không được sử dụng tài liệu.)

- 1) Ai là người đầu tiên phát triển Linux?
 - a) Bill Gates
 - b) Linus Torvalds
 - c) Linus Tormalds
 - d) Linux Torvalds
- 2) Package nào sau đây có thể được sử dụng để thực hiện chức năng web caching?
 - a) Squid
 - b) Apache
 - c) Qmail
 - d) Samba
- 3) Kernel của hệ thống lưu trong thư mục nào?
 - a) /data
 - b) /boot
 - c) /proc
 - d) /krl
- 4) Tên của tập tin cấu hình được sử dụng để cấu hình dịch vụ http là gì?
 - a) http.conf
 - b) apache.cfg
 - c) httpd.conf
 - d) inet.cfg
- 5) Bạn muốn dừng tiến trình inetd ngay tức thì. Nó có mã tiến trình là 15. Bạn dùng lệnh nào sau đây để thực hiện yêu cầu trên?
 - a) Kill -1 15
 - b) Kill -15 9
 - c) Kill -9 15
 - d) Kill -3 15
- 6) Trong máy chủ Linux có tiến trình sau:
 - a) Tiến trình tương tác (interactive processes)
 - b) Tiến trình thực hiện theo lô (Batch processes)
 - c) Tiến trình ẩn trên bộ nhớ (Daemon processes)
 - d) Tất cả các câu trên
- 7) Định dạng mở rộng nào sau đây là chuẩn của Linux?
 - a) .txt
 - b) .tar
 - c) .taz
 - d) .lnx
- 8) Trong tập tin /etc/named.conf, tôi muốn định nghĩa một zone để cấu hình Primary Name Server. Anh/Chị hãy chọn một câu đúng nhất.
 - a) Zone “t3h.com” IN {



- ```

 Type masters;
 File "t3h.com";
 }
b) Zone "t3h.com." IN {
 Type master;
 File "t3h.com";
}
c) Zone "t3h.com" IN {
 Type master;
 File "t3h.com";
}
d) Zone "t3h.com" IN {
 Type master;
 File "t3h.com"
}

```
- 9) Mục đích của shell trong Linux
    - a) Giúp cho người dùng giao tiếp với hệ điều hành.
    - b) Shell được sử dụng để bảo vệ tài nguyên hệ thống.
    - c) Shell lưu giữ những user thông thường can thiệp vào hệ thống.
    - d) Tất cả các câu trên đều sai.
  - 10) Hệ thống bạn có mode mặc định là 666. Bạn chỉ ra giá trị umask là 222, quyền truy cập mặc định của tập tin khi tạo ra là bao nhiêu?
    - a) 444
    - b) 888
    - c) 222
    - d) 666
  - 11) Lệnh nào sau đây được dùng để tạo người dùng có tên susie từ dấu nhắc lệnh?
    - a) useradd susie
    - b) add susie
    - c) linuxconf add susie
    - d) addusers susie
  - 12) Câu nào sau đây giúp bạn tạo password cho cho người dùng có tên susie?
    - a) addpas susie
    - b) passwd susie
    - c) password susie
    - d) susie passwd
  - 13) Hoạt động của mỗi dịch vụ trong hệ thống gắn liền với một/nhiều port. Trong những câu sau đây, câu nào đúng nhất.
    - a) Web : 80; dns : 52; smtp :110; ftp : 20&21
    - b) Web : 80; dns : 52; smtp :110; ftp : 22&21
    - c) Web : 80; dns : 53; smtp :25; ftp : 20&21
    - d) Web : 80; dns : 53; smtp :110; ftp : 22&21
  - 14) Tập tin thiết bị đại diện cho đĩa mềm trong Red Hat Linux là tập tin nào?
    - a) /etc/fd0
    - b) /dev/flp
    - c) /dev/fl0
    - d) /dev/fd0
  - 15) Lệnh nào sau đây được dùng để tạo ra đĩa boot mềm?
    - a) mkdirdisk
    - b) mkbootdisk



- c) mkbootable
- d) mkbootdisk
- 16) Muốn thay đổi thư mục gốc của ftp server. Trong tập tin /etc/ftpaccess bạn cấu hình như sau:
  - a) Anonymous-root “/var/ftpdata”
  - b) Anonymous\_root /var/ftpdata
  - c) Anonymousroot /var/ftpdata
  - d) Anonymous-root /var/ftpdata
- 17) Vì một lý do nào đó, một người dùng muốn chuyển những mail đến địa chỉ mail của mình sang một địa chỉ khác. Khi đó, người quản trị sendmail sẽ thực hiện:
  - a) Trong tập tin alias định nghĩa địa chỉ mail mới.
  - b) Trong thư mục /etc tạo tập tin .forward
  - c) Trong home directory của user tạo tập tin forward
  - d) Trong home directory của user tạo tập tin .forward
- 18) Lệnh nào sau đây giúp bạn thay đổi người sở hữu của tập tin?
  - a) change owner
  - b) file -o
  - c) chown
  - d) change -o
- 19) Lệnh nào sau đây dùng để mount một filesystem có tính năng read-only?
  - a) mount
  - b) mount -r
  - c) mount -a
  - d) mount -ro
- 20) Lệnh nào sau đây được sử dụng để hiển thị bảng partition?
  - a) fdisk -p
  - b) fdisk -t
  - c) fdisk -d
  - d) fdisk -l
- 21) Những file cấu hình của hệ thống lưu trong thư mục nào?
  - a) /config
  - b) /lib
  - c) /etc
  - d) /var
- 22) RPM viết tắt cho từ nào sau đây?
  - a) RedHat Priority Module
  - b) Reduced Priority Module
  - c) RedHat Package Manager
  - d) RedHat Package Module
- 23) Bạn nghi ngờ rằng có một tiến trình đang tạm dừng. Bạn sử dụng lệnh nào sau đây để kiểm tra điều này?
  - a) Process
  - b) Pc
  - c) Jobs
  - d) Susp
- 24) Lệnh nào sau đây cho phép bạn copy một tập tin đến một vị trí nào đó nhưng đã tồn tại một file giống như vậy( ngoài việc thông báo bạn phải ghi đè)
  - a) mv -u
  - b) mv -f
  - c) mv -e
  - d) mv -r



- 25) Cấu trúc của lệnh pipe nào sau đây đúng? (lệnh lpr dùng để in ấn)
- man ls pipe lpr
  - man ls |
  - man pipe
  - man ls | lpr
- 26) Bạn tạo một account có tên jason. Group mặc định của account này?
- everyone
  - domain users
  - jason
  - superuser
- 27) Bạn muốn liệt kê bảng cron của user1. bạn làm điều này như thế nào?
- Cron –user1
  - Crontab –u user1
  - Cron –l user1
  - Crontab –d user1
- 28) Loại người dùng nào liên quan đến quyền hạn của một file hay thư mục trong Linux?
- group
  - owner
  - others
  - a,b,c đều đúng
- 29) Shell mặc định của Red Hat Linux là gì?
- Ksh
  - Sh
  - Bash
  - Csh
- 30) Câu định nghĩa nào sau đây đúng về access list trong squid?
- acl mydomain srcdomain hcmuns.edu.vn
  - acl mydomain srcdomain 172.29.2.0/24
  - acl mydomain srcdomain 172.29.2.4
  - acl mydomain srcdomain .yahoo.com
- 31) Những partition logic được định nghĩa từ số mấy?
- 1
  - 2
  - 6
  - 5
- 32) Người dùng có thể gửi mail nhưng không thể nào nhận mail thông qua POP. Có thể xảy ra lỗi nào sau đây?
- Sendmail bị lỗi.
  - Sendmail chưa được khởi động.
  - Chưa cài POP server.
  - Do đường mạng.
- 33) Khi lập trình shell script, với a và b là 2 biến số biểu thức so sánh nào sau đây đúng?
- [ \$a –eq \$b ]
  - [ \$a = \$b ]
  - [\$a = \$b]
  - [ \$a=\$b ]
- 34) Muốn cho phép tại một thời điểm chỉ có 100 kết nối đồng thời đến Web server. Bạn cấu hình như sau:
- Maxclient 100
  - Client 100



- c) Clients 100  
 d) Maxclients 100
- 35) Khi cấu hình proxy server, bạn khai báo http\_port là 8081 thì những browser khai báo port nào sau đây để có khả năng truy cập internet thông qua proxy
- a) 8080  
 b) 80  
 c) 8081  
 d) không khai báo port
- 36) Bạn cấu hình chứng thực như sau:
- ```
<Directory /upload>
    AuthType Basic
    AuthName public
    AuthUserFile /etc/httpd/conf/htpasswd
    Require user hv1 hv2
</Directory>
```
- Đúng hay sai?
- a) Đúng
 b) Sai
- 37) Những lệnh hay tiện ích nào sau đây dùng để thay đổi địa chỉ IP của máy tính (chọn 3 câu đúng)
- a) linuxconf
 b) ifconfig
 c) chỉnh sửa tập tin /etc/sysconfig/network-scripts/eth0
 d) setup
- 38) Bạn cấu hình web server nhưng chỉ truy cập được theo địa chỉ IP chứ không truy cập được theo tên www.domain
- a) lỗi do web server
 b) lỗi do dns
 c) lỗi do browser
- 39) Trong Red Hat Linux 9, tập tin cấu hình dịch vụ ftp là tập tin nào sau đây?
- a) Vsftpd.conf
 b) Ftpaccess
 c) Ftpd.conf
 d) Vfstd.conf
- 40) Khi cấu hình web server bạn không cần định nghĩa directive ServerName?
- a) Đúng
 b) Sai
- 41) Bạn đã cấu hình quota trong file fstab đúng, dùng lệnh quotacheck và cấp quota cho user nhưng quota vẫn chưa thực thi. Tại sao?
- a) Chưa khởi động lại máy tính
 b) Quota chưa được bật lên
 c) Kiểm tra lại file fstab
 d) Chạy lại lệnh quotacheck
- 42) DNS cung cấp việc chuyển đổi nào sau đây thành địa chỉ IP?
- a) Tên NETBIOS
 b) Hostname
 c) MAC address
 d) CNAME
- 43) Tập tin /etc/resolv.conf có cấu hình như sau:
 Domain csc.com.



- Nameserver 172.29.8.1
Định nghĩa trên đúng sai
- a) Đúng
 - b) Sai
- 44) Apache hỗ trợ virtual host dựa trên
- a) Tên
 - b) địa chỉ ip
 - c) tên và địa chỉ IP
 - d) Không có hỗ trợ virtual host
- 45) Bạn muốn lập lịch một công việc sẽ thực hiện vào 2 giờ ngày 1 tháng 10. Lệnh nào sau đây sẽ hoàn thành điều này?
- a) At 2 4 1
 - b) At 4 1 2
 - c) At 2am April 1
 - d) At April 1 2am



BẢNG TRẢ LỜI

Họ tên học viên:

Lớp :

HỌC VIÊN CHỌN MỤC ĐÚNG CHO MỖI CÂU VÀ ĐÁNH DẤU VÀO BẢNG TRẢ LỜI

Chọn lần đầu:

Bỏ ô đã chọn

Chọn lại ô đã bỏ

	a	b	c	d
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				

	a	b	c	d
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				
36				
37				
38				
39				
40				
41				
42				
43				
44				
45				



II. Mẫu đề thi thực hành

**ĐỀ THI CUỐI HỌC PHẦN IV
PHẦN THỰC HÀNH
Thời gian: 120 phút
(Học viên không được sử dụng tài liệu.)**

Câu 1 (0,5 điểm)

- a) Tìm xem tập tin hosts nằm ở đâu nhưng kết quả không xuất ra màn hình mà xuất vào tập tin /home/hosts.
- b) Chèn nội dung tập tin /etc/sysconfig/network, /etc/sysconfig/network-scripts/ifcfg-eth0 vào sau nội dung tập tin /home/hosts
- c) Copy các tập tin /etc/passwd, /etc/shadow, /etc/sysconfig/network, /etc/sysconfig/network-scripts/ifcfg-eth0 vào thư mục /home
- d) Backup tất cả các tập tin vừa copy trong thư mục /home và tập tin /home/hosts thành tập tin có tên data.tar.gzip

Câu 2 (0,5 điểm)

- a) Tạo user và group theo yêu cầu :
 - o Group **admin** gồm các user **admin1, admin2.**
 - o Group **hocvien** gồm các user **hv1, hv2.**
- b) Cấp quyền cho những user trong group admin có quyền quản trị hệ thống tương đương với user **root**.

Câu 3 (1,5 điểm)

- a) Tạo thư mục /home/pub. Sau đó cấp quyền cho tất cả những user chỉ có quyền read trên thư mục này, riêng những user trong group admin có quyền read, write và execute.
- b) Cấu hình sao cho các user có thể truy cập đến thư mục /home/pub từ Linux hay windows.

Câu 5 (1 điểm) Viết chương trình cho phép tạo user như sau thay vì dùng lệnh useradd, passwd.

Ví dụ: ./taouser nvnguyen

New passwd :

Confirm passwd

Câu 6 (2,5 điểm)

Giả sử bạn có một domain cscXX.edu (*xx là số thứ tự của máy* . Bạn có kế hoạch cấu hình Web server.

- a) Cấu hình DNS Server (Primary Name Server) cho domain “cscXX.edu” sao cho đảm bảo những yêu cầu các dịch vụ đề ra.
- b) Cấu hình Web Server cho domain name cscXX.edu với yêu cầu sau:
 - Thiết kế trang web chủ cho domain cscXX.edu với nội dung giới thiệu về cá nhân của mình như : Họ và Tên, Lop, ...
 - Vị trí lưu trữ website là /home/webdata
 - Thiết kế một trang web có tên “index.html” với nội dung tùy ý lưu trong thư mục /home/www/data.
 - Cấu hình web server sao cho người dùng có thể truy cập những trang web lưu trong thư mục /home/www/data theo đường dẫn http://www.cscxx.edu/data với sự chứng thực của user có username là **local** và password là **local**
 - Cấu hình website cá nhân cho 2 user nvbinh và natan



- c) Ngoài Web Site cho miền chính cscXX.edu. Ta có yêu cầu muốn tổ chức một web hosting cho <http://psv.cscXX.edu> và <http://fpt.cscXX.edu>. Hãy cấu hình theo yêu cầu trên.



ĐỀ THI KIỂM TRA CHUYÊN MÔN GIÁO VIÊN

ĐỀ THI KIỂM TRA CHUYÊN MÔN GIÁO VIÊN

Thời gian:

1. Cách định nghĩa một đĩa cứng logic trong Linux có khác gì so với Windows.
2. Trong khi cài đặt Linux bạn có thể chia 2 primary partition được không ? (Máy không có hệ điều hành nào trước). Những partition logic trong Linux được đánh số thứ tự từ mấy?
3. Cho biết cây thư mục của Linux. Thư mục /etc dùng để làm gì?
4. Có mấy mức để khởi động một hệ điều hành Linux?
5. Liệt kê những lệnh liên quan đến thư mục và tập tin như xóa thư mục, xóa tập tin
...
6. Giả sử thư mục /var là một partition đã hết dung lượng. Nêu tóm tắt các bước để tăng dung lượng của thư mục /var mà không làm mất những dữ liệu đã có.
7. Cho một ví dụ về cách sử dụng của dấu chuyển hướng, dấu đường ống (pipe).
8. Những user thường không phải là root có thể đọc tập tin /etc/shadow hay không ?
9. Cho biết những lệnh thường được sử dụng để quản lý user và group
10. Nêu những bước cơ bản để cấu hình quota
11. Bạn có thể chỉnh sửa địa chỉ IP hay cấu hình mạng theo những cách nào?
12. Dùng lệnh hostname để thay đổi tên máy tính. Khi khởi động lại máy thì tên máy như thế nào? Ngoài cách dùng lệnh bạn còn cách nào để thay đổi tên máy tính?
13. Trình bày các bước cấu hình một DHCP Server.
14. Trong Linux có mấy loại tiến trình? Để đưa một tiến trình từ hậu cảnh sang tiền cảnh bạn dùng lệnh gì?
15. Bạn muốn hủy một tiến trình không có điều kiện, dùng lệnh gì?
16. Họ samba gồm những gì? File cấu hình samba chia làm mấy phần. Muốn chia sẻ một thư mục /public cho mọi user chỉ có quyền read bạn làm như thế nào?
17. Dịch vụ DNS dùng để làm gì? Có mấy loại DNS server
18. FQDN viết tắt cho từ nào. Cho ví dụ.
19. Muốn cấu hình một Primary Name Server bạn cần làm những bước nào?
20. Cho biết trình từ phân giải tên webserver.csc.hcmuns.edu.vn.
21. Thuộc tính forward dùng để làm gì?
22. Cơ sở dữ liệu của Secondary Name Server có từ đâu. Dựa vào đâu mà Secondary cập nhật thông tin khi Primary có sự thay đổi. Và nó cập nhật bao lâu một lần.
23. Bạn khởi động dịch vụ DNS và bị báo lỗi. Bạn phải làm gì để sửa lỗi này để dịch vụ DNS khởi động được.
24. Từ một máy client bạn dùng tiện ích nslookup để kiểm tra việc phân giải tên máy tính thành địa chỉ IP và ngược lại, nhưng không phân giải được. Bạn đoán xem có thể xảy ra những lỗi nào?
25. Bạn có biết trong RedHat Linux 9, cấu hình ftp server bạn dùng package có tên là gì?
26. Muốn cho user anonymous được truy cập đến ftp server và được upload và tạo thư mục trên server. Những thuộc tính chính nào trong file cấu hình giúp bạn thực hiện điều này?



27. Để cấu hình một apache web server hoạt động ở mức cơ bản nhất, bạn quan tâm đến những directive nào?
28. Để cho phép mỗi user trong mạng có thể tạo ra website cá nhân của mình bạn cần phải làm gì để đảm bảo user truy cập được.
29. Có một trang web nào đó mà khi user truy cập đến đòi hỏi phải nhập vào username và password hợp lệ mới được xem nội dung. Trong apache bạn làm cách nào để thực hiện yêu cầu này
30. Bạn có một domain là csc.hcmuns.edu.vn. Bạn muốn tổ chức web server cho domain này. Và ngoài ra bạn muốn tổ chức thêm 2 domain tata.csc.hcmuns.edu.vn và gnt.csc.hcmuns.edu.vn. Bạn phải làm gì để đáp ứng nhu cầu này (Nêu tóm tắt)
31. Trình bày các bước cấu hình một mail server cục bộ bằng sendmail sao cho các user trong mạng có thể trao đổi mail qua lại với nhau.
32. Máy tính của bạn không có kết nối ra ngoài Internet nhưng có kết nối đến một máy tính khác có kết nối Internet. Bạn làm cách nào để cấu hình máy tính của mình là một squid proxy server. (Liệt kê những thuộc tính trong file cấu hình mà mình sẽ sử dụng)
33. Viết một shell script để tính diện tích của một hình chữ nhật



Phần làm bài của giáo viên:

A series of horizontal dashed lines intended for the teacher's notes or answers.