I. Tổng quan hệ điều hành UNIX

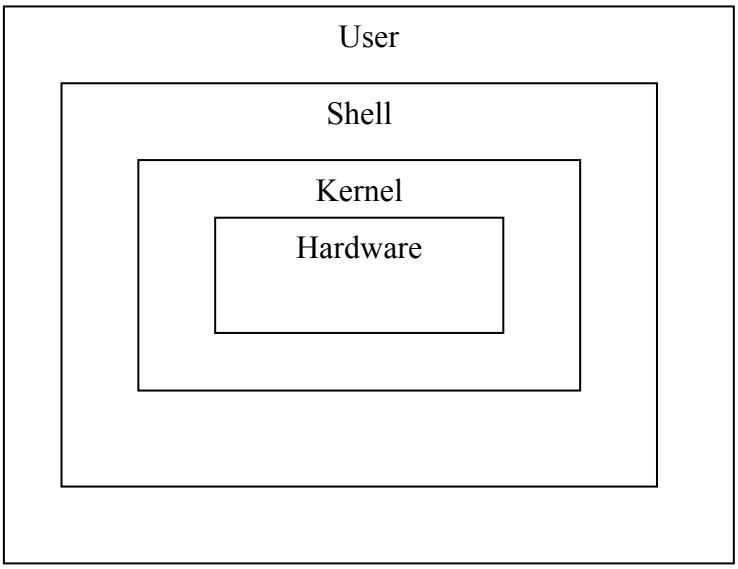
1. Hệ điều hành Unix

UNIX là một hệ điều hành phổ biến, trước đây chúng được sử dụng trong các minicomputer và các workstation trong các công sở nghiên cứu khoa học. Ngày nay UNIX đã trở thành hệ điều hành được dùng cho cả máy tính cá nhân và phục vụ công việc kinh doanh nhờ khả năng mở của nó.

UNIX giống như các hệ điều hành khác nó là lớp nằm giữa phần cứng và ứng dụng. Nó có chức năng quản lý phần cứng và quản lý các ứng dụng thực thi. Điều khác nhau cơ bản giữa UNIX và bất kỳ hệ điều hành khác là sự thực thi bên trong và giao diện.

Hệ điều hành UNIX thực sự là một hệ điều hành. Nó bao gồm các thành phần trước đây (Phần cơ bản vốn có của hệ điều hành Unix) và các thành phần mới bổ sung, nó là lớp nằm giữa phần cứng và các ứng dụng.

Cấu trúc cơ bản của hệ điều hành Unix như sau:



Kernel

Phần quản lý phần cứng và các ứng dụng thực thi gọi là kernel. Trong quản lý các thiết bị phần cứng UNIX xem mỗi thiết bị này như một file (được gọi là device file). Điều này cho phép việc truy nhập các thiết bị giống như việc đọc và ghi trên file. Việc quản lý quyền truy nhập trên các thiết bị thông qua hệ thống kiểm soát bảo mật quyền hạn. Các process đang thực thi được UNIX phân chi tài nguyên bao gồm CPU và các truy nhập tới phần cứng.

Khi khởi động máy tính thì một chương trình unix được nạp vào trong bộ nhớ chính, và nó hoạt động cho đến khi shutdown hoặc khi tắt máy. Chương trình này được gọi là kernel, thực hiện chức năng mức thấp và chức năng mức hệ thống. Kernel chịu trách nhiệm thông dịch và gửi các chỉ thị tới bộ vi xử lý máy tính. Kernel cũng chịu trách nhiệm về các tiến trình và cung cấp các đầu vào và ra cho các tiến trình. Kernel là trái tim của hệ điều hành UNIX.

Khi kernel được nạp vào trong bộ nhớ lúc đó nó đã sẵn sàng nhận các yêu cầu từ người sử dụng. Đầu tiên người sử dụng phải login và đưa ra yêu cầu. Việc login là để kernel biết ai đã vào hệ thống và cách truyền thông với chúng. Để làm điều này kernel gọi chạy hai chương trình đặc biệt là getty và login. Đầu tiên kernel gọi chạy getty. Getty hiển thị dấu nhắc và yêu cầu người sử dụng nhập vào.

Khi nhận được thông tin đầu vào getty gọi chương trình login. Chương trình login thiết lập định danh cho user và xác định quyền của user login. Chương trình login kiểm tra mật khẩu trong file mật khẩu. Nếu mật khẩu không đúng cổng vào sẽ không được thiết lập và bị trả lại điều khiển cho getty. Nếu user nhập đúng mật khẩu chương trình login gửi điều khiển tới chương trình mà có tên nằm trong password file. Thông thường chương trình này là shell.

Shell

Việc thao tác trực tiếp tới kernel là rất phức tạp và đòi hỏi kỹ thuật cao Để tránh sự phức tạp cho người sử dụng và để bảo vệ kernel từ những sai sót của người sử dụng shell đã được xây dựng thành lớp bao quanh kernel. Người sử dụng gửi yêu cầu tới shell, shell biên dịch chúng và sau đó gửi tới kernel.

Chức năng của shell

Thường với UNIX có ba loại shell được dùng phổ biến. Cả ba đều nhằm một mục đích cung cấp các chức năng sau:

- Thông dịch lệnh
- Khởi tạo chương trình
- Định hướng vào ra
- Kết nối đường ống
- Thao tác trên file
- Duy trì các biến
- Điều khiển môi trường
- Lập trình shell

Hiện nay trên hệ điều hành Unix người ta đang sử dụng chủ yếu ba loại shell sau: Bourne shell, Korn shell, C shell. Bảng sau so sánh giữa 3 loại shell (Theo tài liệu UNIX UNLEASHED - Sams Development Team - SAMS Publishibng)

1 tốt nhất, 2 trung bình, 3 yếu.

Shell	*Learning*	*Editing*	*Shortcuts*	*Portability*	*Experience*
Bourne	1	3	3	1	3
C	2	2	1	3	2
Korn	3	1	2	2	1

User

Gồm các tiện tích, các ứng dụng giao tiếp với người sử dụng.

2. Các đặc điểm cơ bản

Hệ điều hành UNIX có một số đặc điểm sau:

- Đa chương

- Nhiều người sử dụng
- Bảo mật
- Độc lập phần cứng
- Hệ mở
- Dùng chung thiết bị
- Tổ chức tập tin phân cấp

Bảng so sánh giữa UNIX, NetWare và Windows NT (Theo tài liệu Upgrading and Repairing Networks –QUE)

Network Goals	**UNIX**	**NetWare**	**Windows NT**
Interoperability	Excellent	Good	Fair
Transparency	Good	Good	Fair
Security	Good	Good	Good
Efficiency	Excellent	Good	Fair
Reliability	Excellent	Good	Good
Accessibility	Good	Excellent	Fair
Cost	Depends	Fair	Fair
Scalability	Excellent	Good	Fair
Third-party utilities available	Excellent	Good	Fair
Directory services	Excellent	Good	Fair
Flexibility	Excellent	Good	Fair
Performance	Excellent	Good	Fair
Print support	Good	Good	Fair
Years of experience	>25	>10	<10

II. Lệnh và tiện ích cơ bản

Các lệnh và tiện ích của Unix rất đa dạng.

Một lệnh UNIX có dạng: $lệnh [các chọn lựa] [các đối số] lệnh thường là chữ nhỏ. Unix phân biệt chữ lớn, nhỏ với chữ lớn.

Ví dụ: $ls -c /dev

Với người sử dụng hệ thống, ta có thể chia lệnh thành các nhóm sau:

1. Các lệnh khởi tạo

login	Thực hiện login vào một người sử dụng nào đó
su	Chuyển sang người sử dụng từ một người sử dụng nào đó
uname	Xem một số thông tin về hệ thống.
who	Hiện lên người đang thâm nhập hệ thống
who am i	xem ai đang làm việc tại terminal
exit	Thoát khỏi hệ thống
env	Xem thông tin tất cả các biến môi trường.
man	Gọi trình trợ giúp

2. Các lệnh hiển thị

echo	Hiển thị dòng ký tự hay biến lên màn hình
setcolor	Đặt màu nền và chữ của màn hình

3. Định hướng vào ra

cmd > File	Chuyển nội dung hiển thị ra file
cmd < file	Lấy đầu vào từ file
cmd>>file	Nội dung hiển thị được thêm vào file
cmd1 | cmd2	Đầu ra của lệnh cmd1 thành đầu vào của lệnh cmd2

4. Desktop:

bc	Dùng để tính toán các biểu thức số học
cal	Hiện lịch
date	Hiển thị và đặt ngày

mesg	Cấm/ cho phép hiển thị thông báo trên màn hình (bởi write/ hello)
spell	Kiểm tra lỗi chính tả
vi	Soạn thảo văn bản
write/ hello	Cho phép gửi dòng thông báo đến những người đang sử dụng trong hệ thống
wall	Gửi thông báo đến màn hình người sử dụng hệ thống

5. Các lệnh thao thư mục và tác file

cd	Thay đổi thư mục
cp	Sao chép một hay nhiều tập tin
find	Tìm vị trí của tập tin
mkdir	Tạo thư mục
rmdir	Xoá thư mục
mv	Chuyển/ đổi tên một tập tin
pwd	Hiện vị trí thư mục hiện thời
ls	Hiện tên file và thuộc tính của nó
ln	Tạo liên kết file (link)
sort	Sắp xếp thứ tự tập tin hiển thị
cat	Xem nội dung của file
tail	Xem nội dung file tại cuối của file
more	Hiện nội dung tập tin trình bày dưới dạng nhiều trang
grep	Tìm vị trí của chuỗi ký tự
wc	Đếm số từ trong tập tin
compress	Nén file.
uncompress	Mở nén.

6. In ấn

cancel	Huỷ bỏ việc In
lp	In tài liệu ra máy in
lpstat	Hiện trạng thái hàng chờ in

7. Thư tín

mail	Gửi - nhận thư tín điện tử
mailx	

8. Quản lý tiến trình

kill	Hủy bỏ một quá trình đang hoạt động
ps	Hiện các tiến trình đang hoạt động và trạng thái của các tiến trình
sleep	Ngưng hoạt động của tiến trình trong một khoảng thời gian

9. Kiểm soát quyền hạn và bảo mật

passwd	thay đổi password hoặc các tham số đối với người sử dụng
chgrp	Thay đổi quyền chủ sở hữu file hoặc thư mục
chmod	Thay đổi quyền hạn trên file hoặc thư mục
chown	Thay đổi người sở hữu tập tin hay thư mục

10. Lưu trữ và hồi phục dữ liệu

cpio	Lưu trữ và hồi phục dữ liệu ra các thiết bị lưu trữ
tar	Lưu trữ dữ liệu ra tape hoặc các file tar

11. Các thao tác trên mạng

ping	Kiểm tra sự tham gia của các nút trên mạng
netstat	Kiểm tra trạng thái của mạng hiện thời
ftp	Thực hiện dịch vụ truyền nhận file

telnet	Thực hiển kết nối với một hệ thống
Uutry	Kết nối UUCP
rcp	Sao chép file ở xa

III.Thâm nhập hệ thống - Các lệnh căn bản

1. Bắt đầu và kết thúc phiên làm việc-Xác lập môi trường hệ thống

Khi bắt đầu làm việc trên hệ thống bạn phải login. Việc login báo cho hệ thống biết bạn là ai và các chủ quyền làm việc của bạn, khi kết thúc phiên làm việc phải logout. Khi đó không có một ai khác có thể truy xuất tập tin của bạn nếu không được phép. Trong một hệ thống có nhiều người sử dụng, mỗi người có một tên và một mật khẩu duy nhất. Quy định tên của người sử dụng không được nhỏ hơn 2 ký tự nếu lớn hơn 8 ký tự thì Unix chỉ lấy 8 ký tự đầu.

Lệnh env: Hiện các thông tin về biến môi trường.

Lệnh su: Chuyển sang người sử dụng từ một người sử dụng nào đó.

su <user>

Ví dụ: Đang ở người sử dụng *anh* muốn tạm chuyển sang người sử dụng root

$su root

Hệ thống sẽ yêu cầu nhập mật khẩu của người sử dụng root

su - <user> - c <command arg>

Chuyển sang user và gọi chạy lệnh command

Lệnh uname: Xem một số thông tin hệ thống hiện thời.

uname <option>

-a	Hiện tất cả các thông tin.
-A	Hiện các thông tin liên quan đến license.
-m	Hiện tên phần cứng của hệ thống
-r	Hiện lên version hệ điều hành.

-s Hiện tên hệ thống

Lệnh who: Hiện một số thông tin sử dụng hệ thống.

who <option>

-r Chỉ ra hệ thống đang chạy tại level nào.

-n Hiện thông tin user đang login vào hệ thống

Ví du: $ who

juucp tty00 Sep 28 11:13

pjh slan05 Sep 28 12:08

Lệnh who am i: Xem ai đang làm việc tại terminal

Lệnh exit: Thoát khỏi shell

Lệnh man: Gọi trình trợ giúp

man <command>

Ví dụ: Muốn xem trợ giúp lệnh cp gõ $man cp

2. Các lệnh hiển thị

Lệnh echo: Hiển thị dòng ký tự hay trị của biến lên màn hình.

echo [-n] [arg]

-n In ra chuỗi kỹ tự mà không tạo dòng mới

Các ký tự qui định khác được quy định giống như qui định trong lệnh printf trong C

(\c In dòng không tạo dòng mới, \t tab, \n in dòng và tạo dòng mới ...)

Ví dụ: $echo ‘Hien len man hinh’

$echo $PATH

Lệnh setcolor: Đặt màu nền và chữ của màn hình.

setcolor <option>

-b <color> Đặt màu nền.

-f <color> Đặt màu chữ.

3. Định hướng vào ra và đường ống:

Định hướng vào ra

Các chương trình nhận dữ liệu nhập và tạo xuất đều có các kênh liên lạc để chuyển các thông tin đó. Đôi khi công việc này được thực hiện tường minh bởi chương trình "mở" một tập tin cụ thể.

Trong UNIX các thiết bị được xử lý như tập tin -> các thao tác vào ra dễ dàng có thể đổi hướng vào ra.

UNIX cung cấp một số phương tiện giúp cho các thao tác định hướng vào ra:

cmd > File Chuyển nội dung hiển thị ra file

cmd < file Lấy đầu vào từ file

cmd>>file Nội dung hiển thị được thêm vào file

Ví dụ: $cal > cde

Hiện nội dung lịch ra file abc

Các đổi thao tác đổi hướng vào ra có thể kết hợp với nhau trong một lệnh.

Ví dụ: cat <file1 > file2

Đường ống

cmd1 | cmd2 Đầu ra của lệnh cmd1 thành đầu vào của lệnh cmd2

Đặc điểm đường ống của UNIX nối kết 1 lệnh này với 1 lệnh khác. Đặc biệt hơn nó tạo xuất chuẩn của 1 lệnh thành nhập chuẩn của 1 lệnh khác. Ký hiệu đường ống (|) được sử dụng để thiết lập đường ống.

Ví dụ: $ls | sort

Nhận xuất của ls và gửi nó đến lệnh sort để sắp thứ tự.

Tổ hợp các tập tin với nhập chuẩn

Trong Unix các lệnh đường ống có thể kết hợp với đổi hướng.

Ví dụ: wc baocao* | sort -n > rep-count kết quả sẽ đưa ra tập tin rep-count.

Các ký kiệu vào ra chuẩn trong lệnh (0: nhập chuẩn, 1: xuất chuẩn, 2: sai chuẩn)

Ví dụ:

spell baocao > baocaodung 2> baocaosai &

Trong lệnh trên các từ sai sẽ được đưa ra file baocaosai.

Các lệnh Desktop

4. Desktop:

Lệnh bc: Dùng để tính toán các biểu thức số học

Ví dụ:

$ **bc**

x=5

10*x

50

^d

Lệnh cal: Hiện lịch dưới dạng sau:

$ cal

February 1994

S	M	Tu	W	Th	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28					

Lệnh date: Hiển thị ngày giờ theo khuôn dạng mong muốn và cho phép thay đổi

ngày hệ thống.

Ví dụ:

$date

Sat Sep 28 1:45:58 EDT 1998

$date +%D

09/28/98

Lệnh mesg: *C*ho phép hoặc cấm hiển thị thông báo trên màn hình (bởi write/ hello)

mesg [n] [y]

n Cấn không cho hiển thị.

y Cho phép hiển thị.

Lệnh spell: Kiểm tra lỗi chính tả xem có lỗi hay không nếu có thì hiện các lỗi sai

Lệnh write/ hello:Cho phép gửi dòng thông báo đến những người đang sử dụng trong hệ thống và thực hiện trao đổi thông tin trực tiếp qua màn hình terminal

write <user>

Hello <user>

Ví dụ:

$write username

< Câu thông báo cần gởi >

^d

Lệnh wall: Gửi thông báo đến tất cả màn hình người sử dụng hệ thống terminal.

Ví dụ:

$wall

Thong bao

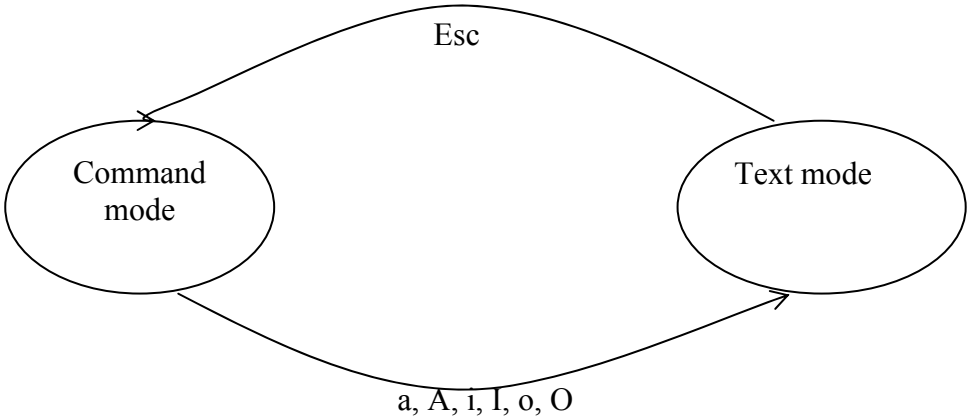
^d

Lệnh vi: Soạn thảo văn bản dạng đơn giản trên Unix

Để thực hiện soạn thảo văn bản dạng text đơn giản, trong Unix hỗ trợ chương trình soạn thảo vi. Trong soạn thảo phân ra hai chế độ là chế độ lệnh và chế độ soạn thảo

Chế độ lệnh: cho phép chèn, xoá, thay thế ...

Chế độ soạn thảo: cho phép soạn thảo văn bản.



Để vào trình soạn thảo vi ta đánh: vi <tên file>

Khởi đầu vi đặt ở chế độ lệnh. Để vào chế độ soạn thảo đánh (a, A, i, I, o, O) thoát khỏi chế độ này đánh ESC, thoát khỏi vi nhấn: x

Một số tuỳ chọn của vi

vi <file> Bắt đầu soạn thảo tại dòng 1

vi +n <file> Bắt đầu ở dòng n

vi +/pattern Bắt đầu ở pattern

vi -r tập tin Phục hồi tập tin sau khi hệ thống treo

Một số lệnh trong command mode của lệnh vi

0	Chuyển con trỏ tới đầu dòng
$	Chuyển con trỏ tới cuối dòng.
/pattern	Tìm xâu văn bản bắt đầu từ dòng kế tiếp.
?pattern	Tìm xâu văn bản từ dòng trước đó về đầu.
a	Thêm text vào sau ký tự hiện thời.
^b	Back up one screen of text.

B	Back up one space-delimited word.
b	Back up one word.
Backspace	Move left one character.
^d	Chuyển xuống dưới nửa trang màn hình
D	Xoá đến cuối dòng
d	dw = Xoá 1 từ, dd = Xoá một dòng
Esc	Chuyển từ insert mode sang command mode
^f	Chuyển xuống một trang màn hình
G	Chuyển con trỏ tới dòng cuối cùng của file
nG	Chuyển con trỏ tới dòng thứ n
h	Chuyển sang trái một ký tự.
i	Chèn text (chuyển sang insert mode) sau ký tự hiện thời.
j	Chuyển con trỏ xuống một dòng.
k	Chuyển con trỏ lên một dòng.
l	Chuyển con trỏ sang phải một ký tự.
n	Lặp lại tìm kiếm.
O	Thêm một dòng mới trên dòng hiện thời.
o	Thêm một dòng mới dưới dòng hiện thời.
Return	Bắt đầu một dòng mới
^u	Chuyển lên nửa trang màn hình.
U	Undo—Thay thế lại dòng hiện thời nếu có thay đổi.
u	Undo thay đổi cuối cùng trên file đang soạn thảo.
W	Move forward one space-delimited word.
w	Di chuyển tiếp một từ.
x	Xoá một ký tự.
:e *file*	Soạn thảo file mới mà không thoát khỏi vi.
:n	Chuyển tới file tiếp trong danh sách file dang soạn thảo.
:q	Thoát khỏi lệnh vi và quay trở lại dấu nhắc của UNIX.
:q!	Thoát khỏi lệnh vi và quay trở lại hệ thống không ghi bất cứ thay đổi nào.

:r file Đọc nội dung file chỉ định và đưa nó vào trong bộ đệm hiện thời của lệnh vi.

:w file Ghi nội dung trong bộ đệm ra file chỉ định.

:w Ghi bộ đệm ra đĩa.

5. Các lệnh thao tác trên thư mục, file

Các thao tác trên thư mục

Lệnh cd: Sử dụng lệnh để thay đổi thư mục làm việc hiện hành.

cd [directory]

Ví dụ: Chuyển đến thư mục /usr/include: $cd /usr/include

Chuyển trở lại thư mục "home": $cd

Chuyển đến một thư mục con của thư mục hiện hành: $cd ccs

Chuyển đến thư mục cha: $cd..

Lệnh ls: Sử dụng lệnh ls để trình bày nội dung của thư mục ls [option] [directory]

option:

-l Hiện chi tiết thông tin một file

-c Hiện danh sách các tập tin theo thứ tự

-a Hiện lên các file bao gồm cả file .<tên file> (vd: .profile)

-F Hiện phân biệt giữa directory (/), executable files (*) với các file thông thường.

-u Sử dụng với –l hiện thay vì hiện last access time sẽ hiện last modification time.

-s Sử dụng với –l hiện file size dưới dạng blocks thay cho dạng bytes.

-t Sử dụng với –u sắp xếp đầu ra theo time thay cho tên.

-r Đảo ngược trình tự sắp xếp

-x Cho phép hiển thị dạng nhiều cột

Ví dụ: $ls -F

bin/ chmod*

etv/ temp

$ls -l

drwx------ 2 sshah admin 512 May 12 13:08 public_html

Lệnh mkdir: Sử dụng mkdir để tạo thư mục

mkdir [-p] [directory]

-p Tạo các thư mục gián tiếp nếu tên thư mục gián tiếp trong đường dẫn là không tồn tại sự tạo.

Ví dụ: tạo các thư mục gián tiếp:

$mkdir -p /usr/tam2/duc

thư mục tam2 không tồn tại do đó tạo cả thư mục tam2 và duc.

Lệnh pwd: Sử dụng lệnh pwd hiện toàn bộ đường dẫn của thư mục hiện hành

$pwd

/usr/tam1/dung

Lệnh rmdir: Sử dụng rmdir để xoá thư mục

rmdir [-ps] <directory>

-p Đòi hỏi chấp nhận các thư mục bị xoá.

-s Không hiện thông báo.

Ví dụ: xoá 1 thư mục rỗng:

$rmdir dung

Các thao tác file

File System

File system là file được lưu trên UNIX. Mỗi file system lưu trong thư mục trong hệ thống cây thưc mục UNIX. Mức đỉnh của cây thư mục là thư mục gốc (root

directory) bắt đầu bằng /. tiếp sau là hệ thống các thư mục con giá trị dài nhất có thể của một thư mục là 1,024 ký tự.

Thông thường ít quan tâm đến mức thấp của một file lưu trên hệ thống UNIX nhưng để hiểu kỹ ta cần quan tâm đến hai khái niệm inodes và superblock. Một khi đã hiểu nó sẽ giúp bạn thuận lợi trong việc quản trị hệ thống file.

inodes

Inode duy trì thông tin về mỗi file và phụ thuộc vào kiểu file, Inode có thể có chứa hơn 40 phần thông tin. Tuy nhiên hầu như chỉ có tác dụng đối với kernel và không liên quan đến người sử dụng. Phần liên quan chủ yếu đến người sử dụng là:

mode: Đánh dấu quyền truy nhập và kiểu file.

link count: Số liên kết có chứa inode này.

user ID : ID của người chủ sở hữu file.

group ID: ID Group của file.

size Number: Kích thức file.

access time: Thời điểm truy nhập gần nhất.

mod time: Thời điểm sửa đổi gần nhất.

inode time: Thời điểm mà cấu trúc inode thay đổi gần nhất.

block list: Danh sách số block đĩa mà có chứa segment đầu của file.

Superblocks

Là thông tin đặc biệt quan trọng lưu trên dĩa. Nó có chứa thông tin định hình của đĩa (số head, cylinders ...), phần đầu của danh sách inode, và danh sách block tự do. Bởi vì thông tin này là quan trọng cho nên hệ thống tự động giữa một bản sao trên đĩa tránh việc rủi ro. Nó chỉ liên quan đến khi mà file system bị hỏng nặng.

Các kiểu File

Có 8 kiểu file là: Normal Files, Directories, Hard Links, Symbolic links, Sockets, Named Pipes, Character Devices, Block Devices.

Normal Files: Là loại file sử dụng thông thwngf nhất, chúng có thể là text hoặc

binary file tuy nhiên cấu trúc bên trong không liên quan đến quan điểm quản trị hệ thống. Đặc tính của file được xác định bởi inode trong file system mà mô tả nó. Lệnh ls -l chỉ ra Normal Files như sau:

-rw------- 1 sshah admin 42 May 12 13:09 hello

Directories: Là loại files đặc biệt mà có chứa các file khác. Chỉ có một ánh xạ từ inode tới disk blocks, có thể có nhiều ánh xạ tới một từ một mục của thư mục tới inode. Khi dùng lệnh ls –l một Directorie hiện như sau:

drwx------ 2 sshah admin 512 May 12 13:08 public_html

Hard Links: Hard link là một directory entry ngoại trừ việc thay vì trỏ tới file duy nhất nó trở tới file đã tồn tại. Điều này tạo ra có hai file giống hệt nhau khi liệt kê danh sách file. dùng lệnh ls -l:

-rw------- 1 sshah admin 42 May 12 13:04 hello

sau khi thực hiện Hard link dùng lệnh ls –l sẽ hiển thị như sau:

-rw------- 2 sshah admin 42 May 12 13:04 goodbye

-rw------- 2 sshah admin 42 May 12 13:04 hello

Symbolic Links: Symbolic link khác với hard link là nó không trỏ tới một inode khác nhưng trỏ tới một filename khác. Điều này cho phép symbolic links thực hiện liên kết các file systems một cách thuận lợi sử dụng lệnh ln -s ta thấy file www hiện như sau:

drwx------ 2 sshah admin 512 May 12 13:08 public_html

lrwx------ 1 sshah admin 11 May 12 13:08 www -> public_html

Sockets: Sockets dùng cho UNIX liên kết mạng với máy khác. Điều này được sử dụng nhờ network ports. Dùng lệnh ls -l socket file hiện như sau:

srwxrwxrwx 1 root admin 0 May 10 14:38 X0

Named Pipes: Giống như socket named pipe cho phép chương trình liên lạc với nhau qua file system. Bạn có thể sử dụng lệnh mknod để tạo named Pipe. Dùng lệnh ls -l named pipe hiện như sau:

prw------- 1 sshah admin 0 May 12 22:02 mypipe

Character Devices: Là kiểu file đặc biệt dùng để liên lạc với các system Device driver. Dùng lệnh ls -l character device hiện như sau:

crw-rw-rw- 1 root wheel 21, 4 May 12 13:40 ptyp4

Block Devices: Block devices hầu như chia sẻ các đặc tính với các character devices trong thư mục /dev, được sử dụng để liên lạc với các device drivers. Điểm khác của block devices là khả năng truyền một khối lượng lớn dữ liệu tại một thời điểm. Dùng lệnh ls –l hiện như sau:

brw------- 2 root staff 16, 2 Jul 29 1992 fd0c

UNIX tổ chức hệ thống tập tin bao gồm chỉ một thư mục gốc (/) mà từ đó các thư mục con của nó được gắn vào một cách trực tiếp hay gián tiếp. Có một vài thư mục con chuẩn /bin, /usr, /etc, v.v... Mỗi thư mục này lại chứa các tập tin hay thư mục con.

Ta có thể sử dụng đường dẫn đầy đủ để xác định một tập tin, ví dụ: /usr/NVA/chuong1. Bạn cũng có thể sử dụng chỉ tên tập tin nếu tập tin được chứa trong thư mục hiện hành. Thường khi login, thư mục hiện hành sẽ được đặt đến là thư mục "home". Đây là thư mục được thiết lập bởi người quản trị hệ thống dành cho người sử dụng.

Tên tập tin trong UNIX có thể dài 256 ký tự, ngoại trừ các ký tự đặc biệt sau: ! " ' ; / $ < > ( ) [ ]. { }. Ngoài ra ta cũng có thể sử dụng các ký tự sau:

Các ký hiệu đại diện:

Dấu (*) đại diện cho một, nhiều hoặc không ký tự nào.

Dấu (?) đại diện cho một ký tự đơn

[...] đại diện cho một dãy ký tự có thứ tự trong bảng Alphabet. Ví dụ: liệt kê tất cả các thư mục bắtt đầu bằng chữ c, d, e: lc [cde]*

Lệnh file: Nhận biết dạng file. Thông thường lệnh file phân tích nội dung của 1 file và hiển thị tính chất của thông tin chứa trong file:

Ví dụ: $ file /etc/passwd => /etc/hosts: ascii text

Lệnh cat: Xem nội dung 1 tập tin và nối kết các tập tin cat [option] [files]

Lệnh more, pg: Dùng lệnh more hoặc pg để hiện nội dung file trên từng trang màn hình:

Ví dụ: more thu

Lệnh mv: Đổi tên tập tin.

mv <option> <old><new>

-i Nếu file đã có thì lệnh sẽ hỏi có ghi đè lên file hay không.

-f Thực hiện lệnh mà không hỏi gì.

Ví dụ: Thay đổi tên của tập tin ở thư mục hiện hành:

$mv a.out test

Lệnh ln: Sử dụng lệnh ln để gán thêm 1 tên mới cho 1 tập tin

Cú pháp: ln [-s] <tên> <file, directory>

ln Không tham số tạo hard link.

ln -s Tạo symbolic link.

Lệnh rm: Xoá tập tin rm tập tin.

rm <option> <files>

-f Xoá các tập tin mà không hỏi, thậm chí chủ quyền ghi là không cho phép

-r Cho phép xoá cả thư mục bao gồm cả file và thư mục.

-i: Trước khi xoá tập tin sẽ hỏi xác nhận việc xoá tập tin

Ví dụ: Xoá tập tin thu:

$rm thu

Lệnh cp: Sao chép tập tin.

cp <option> <sou> <des>

-i Nếu file trên đích đã có thì sẽ được hỏi có ghi đè hay không

-r Copy cả thư mục

Ví dụ: cp /etc/passwd /usr/dung/passwdold

Lệnh find: Tìm kiếm 1 tập tin hoặc một số tập tin thoả mãn điều kiện nào *đó*

find <path> <expression>

-atime <n> Đúng nếu file bị truy nhập n ngày trước đây.

-mtime <n> Đúng nếu file bị thay đổi n ngày trước đây.

-user <un> Đúng nếu chủ của files là un. Nếu giá trị là số nó sẽ so sánh với userID.

-group <gn> Đúng nếu files thuộc thánh viên của nhóm gn. Nếu gn là số thì nó sẽ so sánh

với groupID.

-perm <on> Tìm files có quyền truy nhập files đúng với giá trị on.

-links <n> Tìm files có n links.

-type <*x*> Tìm file có kiểu x.

-newer <fn> Tìm file bị thay đổi gần hơn so với fn.

-local Chỉ tìm tại local.

-size <n> [c] Tìm file có kích thước n blocks (c chỉ ra character –byte)

-print Hiện đầy đủ đường dẫn của files.

-depth Luôn đúng (cho phép tìm tất cả các files trên directory).

-name <pt> Tìm files thoả mãn mẫu tìm pt.

Ví dụ: Tìm tập tin thu:

$find . -name thu -print

/usr/tam/thu

Lệnh grep: Tìm kiếm chuỗi văn bản bên trong tập tin

grep <option> <Chuỗi cần tìm> <Files>

Sử dụng lệnh grep để tìm kiếm một chuỗi văn bản bên trong các tập tin được chỉ định. Nếu chuỗi văn bản dài hơn 1 ký tự thì phải để trong hai dấu nháy.

-c In ra tổng số dòng có chứa mẫu cần tìm.

-h Bỏ tên file không hiện lên tại dòng có chứa mẫu tìm thấy.

-i Bỏ qua phân biệt chữ hoá và chữ thường.

-n Hiện lên dòng chứa mẫu tìm thấy và trước đó là số của dòng trong file.

-v In tất cả các dòng có chứa mẫu tìm kiếm.

Dấu * đại diện cho một hoặc bất kỳ ký tự nào.

[character] Đại diện bởi một mảng các ký tự.

Ví dụ: Tìm chuỗi ký tự "mail" trong tập tin thu: $grep ‘ngan hang’ thu

Lệnh tail: Hiện các dòng cuối của files

tail [-f] <file>

-f Hiện 10 dòng cuối và mỗi dòng được thêm vào từ khi gọi chạy lệnh tail cho đến khi

kết thúc nó.

Lệnh compress: Thực hiện việc nén dữ liệu. File tạo ra tự động có đuôi *.Z*

compress [-cfv] file(s)

-c Writes to stdout instead of changing the file.

-f Cho phép nén cả file đã được nén.

-v Hiển thị phần trăm giảm mỗi lần nén.

Lệnh uncompress: Thực hiện việc cởi nén file dữ liệu.

uncompress [-cv] [file(s)]

6. In ấn

Lệnh lp: Thực hiện việc in file máy in.

lp <option> <files>

-c Khi lệnh được gọi lập tức tạo bản sao của file và thực hiện in trên bản sao này.

-d <des> Đích cần in tới.

-n <num> Số bản in.

-o nobaner Không in phần trang tiêu đề đầu tiên.

cpi=<n> Số character được in trên 1 inch (10/12 ...)

-q <pri> Mức độ ưu tiên in (0 cao nhất ->39 thấp nhất>

Ví dụ: $lp -d epson thu.txt

Lệnh lpstat: Hiện trạng thái hàng chờ in của máy in

lpstat <option>

-a Hiện danh sách các printer.

-d Hiện máy in ngầm định.

-o <pr>[-l] Hiện trạng thái yêu cầu đầu ra của máy in pr. Nếu có -l hiện chi tiết trạng thái.

-p <pr> Hiện trạng thái máy in.

-t Hiện tất cả thông tin về trạng thái.

-v <pr> Hiện tên máy in và tên đường dẫn tới thiết bị tương ứng.

Lệnh cancel: Huỷ bỏ việc In ấn

cancel <request id> <printer>

7. Thư tín điện tử

Trong Unix hỗ trợ chương trình mail và mailx cho phép người sử dụng thực hiện việc gửi và nhân mail theo chuẩn SMTP

Lệnh mail:

Thư điện tử cung cấp các khả năng cơ bản để gửi và nhận thông báo. Mail sẽ hiển thị thông báo theo thứ tự vào trước ra sau.Sau khi hiển thị mỗi thông báo mail sẽ hiện lên dấu ? để chờ lệnh của người sử dụng.

Gồm một số lệnh sau:

+	Hiện message tiếp theo
-	Hiện mesage trước đó
d	Xoá message hiện thời.
h	Hiện header của message.
m <user>	Gửi message tới user.
p	Hiện lại nội dung message.
s <file>	Ghi message ra tập tin hoặc mbox
w<file>	Ghi message ra file nhưng không ghi phần header.
q	Thoát khỏi mail
x	Thoát khỏi mail mà không thay đổi thông báo
! lệnh	Thực hiện lệnh Unix

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

Ví dụ: $mail dung@sysdomain

<thông báo>

ctrl-d

Nhận thư: Khi login vào hệ thống nếu có thư hệ thống sẽ thông báo " You have mail" khi đó có thể đánh $mail để thực hiện các thao tác trên message.

Lệnh mailx

Mailx bao gồm các lệnh để chuyển và nhận thư.: mailx <option> <user>

option -e soạn message

-v soạn message bằng lệnh vi

-r <file> đọc file vào message

!<lệnh>		Cho phép thực hiện các lệnh shell.
=		Hiện số của message
delete	d	Xoá message hiện thời
dp		Xoá message và chuyển đến message kế tiếp
edit	e	Soạn thảo message.
exit		Thoát khỏi mailx và không lưu lại thay đổi.
headers	h	Hiện header của message chỉ định.
mail <user>	m	Gửi message tới người chỉ định
next	n	Hiện message tiếp theo
print	p	Hiện message lên màn hình.
quit	q	Thoát khỏi mailx và hoàn thành các thao tác đã thực hiện.
reply	r	Trả lời mail tới người gửi và các người nhận.
Reply	R	Chỉ trả lời mail tới người gửi.
save <file>	s	Ghi message tới file chỉ định.
Save	S	Ghi message sau khi gửi nó.
undelete	u	Bỏ xoá bởi lệnh delete.
visual	v	Dùng lệnh vi để soạn thảo message.
write <file>	w	Ghi message mà không ghi header.

8. Quản lý tiến trình

Hệ thống Unix dùng các tiến trình (process) như là phương tiện để quản lý các chương trình đang thực hiện. Mỗi lệnh mà người dùng gọi thực hiện đều gọi là một tiến trình. Mỗi tiến trình đang hoạt động đều bao hàm một số thông tin liên quan đặc biệt có một giá trị ID để nhận dạng. Ngoài ra còn các thông tin khác như TTY, thời gian, lệnh ...

Tiến trình tạo ra một tiến trình khác được gọi là tiến trình cha. Các tiến trình con nhận biết tiến trình cha của nó qua ID của quá trình cha.

Lệnh kill: Hủy bỏ một quá trình đang hoạt động.

kill <signal> PID

-0 Kết thúc tất cả các tiến trình trong process group.

-9 Kết thúc tiến trình không điều kiện.(Unconditional kill signal)

-15 Kết thúc tiến trình- ngầm định (software termination signal)

Lệnh ps: Hiện các tiến trình đang hoạt động và trạng thái của các tiến trình bao gồm các trường thể hiện các thông tin sau:

ps <option>

-e Hiện thông tin về tất cả các process đang hoạt động.

-d In thông tin về tất cả các tiến trình trừ phần leader

-a In tất cả thông tin về process trừ các process không tương ứng với terminal.

-f In đầy đủ tất cả các thông tin.

Lệnh sleep: Ngưng hoạt động của tiến trình trong một khoảng thời gian.

sleep <time>

Thời gian được tính bằng giây 1-65536.

Lệnh wait: Cho phép chờ các tiến trình chạy chế độ background kết thúc. Thường dùng trong các script để đồng bộ tương tác.

Ví dụ:

cmd1 > file1&

cmd2>file2&

wait

sort file1 file2

9. Các lệnh liên quan bảo mật và quyền hạn

a) Khái niệm:

Khi người sử dụng được tạo thì các thông tin sau yêu cầu được được đưa vào:

Tên người sử dụng

Mật khẩu

Số nhận dạng (uid: user identify number)

Số của nhóm (gid: group identify number)

Chú thích

Thư mục xâm nhập (home directory)

Tên chương trình cho chạy lúc bắt đầu làm việc

Các thông tin này được chứa trong file /etc/passwd

Nhóm người sử dụng: 1 nhóm người sử dụng là tập hợp của 1 số người sử dụng có thể dùng chung các file của nhau, được mô tả bằng những thông tin sau:

Tên của nhóm

Mật mã (có thể có hoặc không có)

Số của nhóm (gid)

Danh sách những người sử dụng thuộc nhóm

Các thông tin này được lưu trong tập tin /etc/group

Trong một file được tạo trong đó có các thông tin sau:

-rwxr-xr-- 2 sshah admin 42 May 12 13:04 hello

Nhóm đầu -rwxr-xr—

Dấu gạch đầu tiên thể hiện loại file.

Nhóm tiếp theo gồm 3 nhóm nhỏ **rwx**rwx**rwx** nhóm đầu thể hiện quyền hạn của người chủ sở hữu tập tin, nhóm 2 thể hiện quyền hạn của nhóm có quyền truy nhập, nhóm 3 là quyền hạn của các người sử dụng khác. Mỗi nhóm gồm 3 giá trị r (đọc), w (ghi, thay đổi), x (thực hiện).

Permission	Owner	Group	Other
Read (r)	4	4	4
Write (w)	2	2	2
Execute (x)	1	1	1
Total	7	7	7

Tiếp theo chỉ ra số liên kết trên file (2)

Người chủ sở hữu của file (sshah)

Nhóm người sử dụng có quyền trên file (admin)

Kích thước file (42)

Thời gian (May 12 13:04)

Tên file (Hello)

b) Các lệnh

Lệnh chgrp: Thay đổi nhóm truy xuất của tập tin. Chỉ có superuser hay người sở hữu mới được quyền thay đổi quyền sở hữu file.

chgrp <group> <files>

Ví dụ: $ls -a test

-rwx--x--x 1 bin bin 13023 Jun 21 94 test

$chgrp data test

$ls -a test

-rwx--x--x 1 bin data 13023 Jun 21 94 test

Lệnh chown: Thay đổi người sở hữu tập tin. Chỉ có superuser hay người sở hữu mới được quyền thay đổi.

chown <owner> <files>.

Ví dụ: $ls -a test

-rwx--x--x 1 bin data 13023 Jun 21 94 test

$chown dung test

$ls -a test

-rwx--x--x 1 dung data 13023 Jun 21 94 test

Lệnh umask: Đặt quyền truy xuất ngầm định đối với 1 file hay thư mục tạo. Sau khi đặt umask tất cả các tập tin và thư mục tạo sẽ nhận quyền truy nhập. (giá trị ngầm định là 022)

umask <mask>

0	read and write (and execute for directories)
1	read and write (not execute for directories)
2	read (and execute for directories)
3	read
4	write (and execute for directories)
5	write
6	execute
7	no permissions

Ví dụ:

$umask 177

$cat test

...

^d

$ls -l test

-rw------ 1 dung adm 1 Mar 11 10:11 test

Lệnh chmod: Thay đổi quyền hạn truy nhập tập tin

Để thực hiện việc thay đổi quyền hạn trên tập tin thì đòi hỏi người thực hiện lệnh phải là người sở hữu hay superuser (root).

chmod <mode> <files>

Ví dụ:

$ls -l test

-rw------ 1 dung adm 1 Mar 11 10:11 test

$chmode 754 test

$ls -l test

-rwxr-xr-- .....

Các tham số dùng với lệnh chmode <option> <files>

u: người sử dụng (user)

g: nhóm (group)

o: chung (other)

a: tất cả (all)

Toán tử:

+: thêm quyền

-: bớt quyền

=: gán giá trị khác

Quyền:

r: đọc (reading)

w: ghi (writting)

x: thực hiện (execution)

Ví dụ: $ls -l test

-rwxr-xr-- .....

$chmode g +w test

$ls -l test

-rwxrwxr-- ..... test

10. Lưu trữ và hồi phục dữ liệu

Các tập tin của những hệ thống thông tin ngày càng lớn, sự cần thiết và mức độ quan trọng của các tập tin này vô cùng quan trọng.

Các thiết bị phần cứng không thể đảm bảo rằng không bao giờ có sự cố như hỏng đĩa, hỏng thiết bị lưu dữ liệu ... Các hệ thống phần mềm cũng không phải là chạy hoàn toàn không xảy ra sự cố gì.

Dữ liệu của các hệ thống thông tin xử lý nghiệp vụ tức thời đòi hỏi khi có bất cứ sự cố nào xuất hiện làm hỏng dữ liệu của hệ thống, thì ngay sau đó dữ liệu phải được khôi phục ngay.

Tùy theo các nhà cung cấp phần mềm và phần cứng khác nhau mà hệ thống lưu trữ có những chức năng và tiện ích khác nhau nhưng đều có chung một mục đích là lưu dữ liệu.

Thông thường những hệ điều hành lớn hoặc nhỏ đều hỗ trợ các tiện ích giúp cho việc sao lưu và hồi phục dữ liệu nhằm giảm tối thiểu các ảnh hưởng đến hệ thốngdữ liệu.

Trong các loại hệ điều hành Unix khác nhau có thể có các công cụ và các tiện ích giúp cho việc thực hiện lưu trữ tuy nhiên các lệnh hầu hết được hỗ trợ bởi các loại

hệ điều hành Unix.

Lệnh tar: Lưu trữ hoặc hồi phục files từ các thiết bị lưu trữ.

tar <options> <tarfile name> <filenames to backup or restore>

Một số các option hay dùng.

option =[key]<sub>

Key c Tạo lưu trữ mới và thực hiện bắt đầu ghi từ đầu.

r File lưu trữ được ghi vào vih trí cuối của thiết bị lưu trữ.

t Hiện danh sách các file lưu trữ.

u Cập nhập thêm nếu files chưa có, hoặc đã thay đổi so với lần sao chép trước.

x Lấy thông tin từ thiết bị lưu trữ.

<sub> e Cho phép ghi trên nhiều volume.

f Sử dụng các đối số thay cho các giá trị ngầm định.

n Chỉ thiết bị lưu trữ không phaie là tape

v Hiển thị tên file sao lưu hoặc hồi phục

Ví dụ: tar cvf /dev/rmt/0hc /usr/local/datafiles

tar tvf /dev/rmt/0hc > tarlist.txt

cd /usr/contrib

tar xvf /dev/rmt/0hc

Lệnh cpio: Lưu trữ và hồi phục dữ liệu ra các thiết bị lưu trữ chuẩn

cpio -o [acv] Ghi dữ liệu ra đầu ra thiết bị lưu trữ

-i [cdmnv] Đọc dữ liệu từ thiết bị lưu trữ.

c Ghi thêm phần header phòng trường hợp dùng trên các máy khác.

d Thư mục sẽ được tạo nếu cần.

m Dữ lại thời gian thay đổi trước đây.

a Thay đổi thời gian truy nhập.

u Sao chép không điều kiện.

v Hiện danh sách tên file.

Ví dụ: ls | cpio -oc > /dev/rst0

cpio -icd < /dev/rst0

Lệnh dd: Sao lưu và hồi phục dữ liệu theo đúng trạng thái trên hệ thống file (block copy)

dd [if=][of=]

if= Đầu vào chuẩn.

of= Đầu ra chuẩn.

Ví dụ:

dd if=/dev/diskette0 of=/mnt/abc.xx

dd if=/mnt/abc.xx of=/dev/diskette0

11. Các thao tác trên mạng

Lệnh ping: Kiểm tra sự tham gia của các nút trên mạng

Lệnh netstat: Kiểm tra trạng thái của mạng hiện thời của hệ thống local. Nó thể hiện các thông tin về giao diện mạng, thông tin routing table, thông tin về Protocol.

netstat <option>

-a Hiển thị thông tin tất cả các interface

-c Tiếp tục hiển thị và tự update sau một vài giây.

-i Chỉ hiển thị thông tin về interface

-n Hiển thị địa chỉ thay cho tên.

-r Hiển thị thông tin về kernel routing table

-t Chỉ hiển thị thông tin về TCP socket

-u Chỉ hiển thị thông tin về UDP socket.

-x Hiển thị thông tin về socket

Ví dụ:

netstat –r

Routing Table:

Destination	Gateway	Flags	Ref	Use	Interface
localhost	localhost	UH	0	109761	lo0
rmtnet	ws2	UG	0	20086	
rmtppp	ws2	UG	0	1096	
subnet1	ws1	U	3	1955	le0
224.0.0.0	ws1	U	3	0	le0
default	gateway	UG	0	16100	

Lệnh telnet: Thực hiện kết nối với một hệ thống cho phép trở thành terminal của hệ thống mà nó kết nối tới.

telnet <dest>

Lệnh ftp: Thực hiện dịch vụ truyền nhận file.

ftp <dest>

Muốn debuge lệnh ftp dùng thêm option -d (ví dụ: ftp -d ftp.ha.com)

Một số lệnh trong ftp:

ascii Chuyển sang ASCII transfer mode.

binary Chuyển sang binary transfer mode.

cd	Thay đổi thư mục trên ftp server.
close	Kết thúc kết nối.
del	Xoá file trên ftp server
pwd	Hiện thư mục hiện thời trên ftp server
get	Lấy file từ ftp server
help	Trợ giúp
lcd	Thay đổi thư mục trên client
mget	Lấy một số file trên ftp server
mput	Truyền một vài file lên ftp server
open	Mở kết nối với ftp a server
put	Truyền file tới ftp server
quit	Thoát khỏi FTP.

Lệnh rlogin: Thực hiện login tới máy ở xa, cho phép truy nhập tới máy tính trên mạng giống như lệnh telnet.

rlogin <hosts name>

Trong trường hợp này user ID trên remote host phải giống ới user ID trên local host. ví dụ như nếu testuser login vào box1, rlogin dùng testuser login vào box2.

Tuy nhiên nếu muôn slogin vào user ID khác dùng option sau:

rlogin <hosts name> -l <user>

Lệnh rcp: Sao chép file ở xa

Trước khi sử dụng lệnh rcp người sử dụng đã phải được sẵn sàng trên remote machine. Bởi vì rcp không sử dụng authentication (không giống như rlogin).

rcp <option> <sour> <dest>

-r Chỉ sử dụng trong trường hợp copy thư mục.

<sour> <dest> được viết theo quy định

hostname:filepath

user@hostname:filepath

user@hostname.domain:filepath

Ví dụ: rcp ha@box1:/export/home/ha/abc.txt box2:/export/home/cc.m

IV.Lập trình Shell

1. Các đặc tính cơ bản.

Lệnh đơn giản

Là lệnh được gọi thực hiện có tính chất đơn như dạng sau:

command <option> <arg>

Nhiều lệnh trên một dòng

Thông thường shell thông dịch từ đầu tiên như là lệnh còn các phần sau trở thành các đối số của lệnh. Có 3 ký tự đặc biệt mà khi shell thông dịch mà gặp phải sẽ hiểu sau đó là có một lệnh tiếp theo cần thực hiện đó là (;), (&), (|).

; Đợi lệnh trước hoàn thành mới thực hiện đến lệnh tiếp sau (tương đương với thực

hiện các lệnh riêng rẽ).

Ví dụ: $ who -H; df -v; ps -e

& Lệnh sau không cần phải đợi lệnh trước kết thúc thực hiện.

Ví dụ: $who -H & df -v & ps -e

| Sẽ lấy đầu ra của lệnh trước thành đầu vào của lệnh sau:

Định hướng vào ra

Khi shell thông dịch lệnh mà nhìn thấy các ký hiệu đổi hướng vào ra (<), (>). Các định hướng vào ra này được gửi tới subshell để điều khiển việc thực hiện lệnh.

Dòng lệnh dài

Trong trường hợp dòng lệnh dài muốn chia thành nhiều dòng thì kết thúc dòng phải

đặt ký tự (\). Khi gặp ký tự này shell không coi dòng mới là kết thúc của đầu vào.

Ví dụ: $ echo Now is the time for all good men _

to come to the aid of the party.

Biến trong shell

Khi shell gặp ký tự $ thì nó hiểu từ sau đó là tên biến. Shell sẽ tìm biến đã được định nghĩa và lấy giá trị của nó. Nếu biến chưa định nghĩa thì một null string sẽ được trả lại.

Để đặt giá trị cho biến chỉ cần gán <tên biến>=giá trị

Ví dụ:

$ dir=ls

$ $dir f*

file1

file1a

Có thể thực hiện gán nhiều hơn một biến trên một dòng lệnh. Biến sẽ được gán từ phải sang trái.

Ví dụ: X=$Y Y=y

echo $X

y

Nếu muốn lấy đầu ra hoặc kết quả thực hiện một lệnh làm đối số của một lệnh khác thì có thể dùng dấu (`) để bao lấy lệnh cần thực hiện.

{ variable:-value}	Gán giá trị ngầm định cho biến.
{ variable:+value}	Nếu biến khác null thì sẽ lấy value.
${variable:?message} ra lỗi	Nếu biến không đặt giá trị thì message sẽ được in ra đầu
	chuẩn.

Ví dụ:

$ echo Hello $UNA

Hello

$ echo Hello ${UNA:-there}

Nếu không gán trị cho UNA thì sẽ hiện

Hello there

Nếu gán UNA=John

Sẽ hiện

Hello John

Khi shell gọi thực hiện một lệnh từ đầu tiên được hiểu là lệnh, các thông tin sau sẽ được hiểu là các đối số của lệnh. Đối số đầu tiên được gán tới biến $1, đối số thứ hai sẽ được gán tới biến $2 ... $9. Vị trí biến $0 luôn có chứa lệnh.

$# Chứa số đối số được gửi tới lệnh qua vị trí của biến.

$$ Chứa process ID của process hiện thời.

$? Có chứa trạng thái của lệnh cuối cùng. Mang giá trị 0 nếu lệnh thực hiện thành công,

khác không nếu có lỗi xuất hiện.

$* Có chứa tất cả positional argument được gửi tới chương trình.

Ví dụ: có một script file như sau:

restoreany

cd $WORKDIR

cpio -i $* </dev/rmt0

$ restoreany file1 file2 file3

Các file file1 file2 file3 sẽ được restore từ thiết bị lưu trữ

Lệnh shift: Dùng để dịch biến vị trí

Ví dụ:

$1 = -r $2 = file1 $3 = file2

shift

Kết quả là:

$1 = file1 $2 = file2

Biến môi trường

Là các biến mà shell hoặc bất kỳ một chương trình nào có thể lấy và truy nhập nó. Có một số biến môi trường ngầm định trong shell như HOME, MAIL, PATH, PS1, PS2 ...

2. Lập trình shell

Trong Shell các biến thông thường được lưu dưới dạng character để thực hiện các tính toán toán học phải dùng lệnh expr.

expr <integer operator integer>

Các toán tử là (+), (-), (*), (/).

Các phép toán trả về phần nguyên.

Ví dụ:

$ expr 5 + 7 / 3

7

Null command đại diện bởi dấu (:)

Để lấy dữ liệu trực tiếp từ người sử dụng dùng lệnh read

$ read var1 var2 var3

Hello my friend

$echo $var1 $var2 $var3

Hello my friend

a) Lệnh điều kiện

Lệnh true và false: True luôn trả giá trị 0, false luôn trả giá trị 1

Lệnh test: Kiểm tra điều kiện xem đúng hay sai

test condition

Testing Character Data

str1 = str2 Đúng nếu str1 giống hệt str2 (về độ dài và ký tự)

str1 != str2 Đúng nếu str1 khác str2

-n str1 Đúng nếu chiều dài str1 lớn hơn 0 (is not null)

-z str1 Đúng nếu str1 là null (chiều dài =0)

str1 Đúng nếu *str1* khác null

Testing Numeric Data

int1 -eq int2 Đúng nếu int1 bằng int2

int1 -ne int2 Đúng nếu int1 khác int2

int1 -gt int2 Đúng nếu int1 lớn hơn int2

int1 -ge int2 Đúng nếu int1 lớn hơn hoặc bằng int2

int1 -lt int2 Đúng nếu int1 nhỏ hơn int2

int1 -le int2 Đúng nếu int1 nhỏ hơn hoặc bằng int2

Testing for Files

-r filenm Đúng nếu user có quyền đọc filenm

-w filenm Đúng nếu user có quyền ghi trên filenm

-x filenm Đúng nếu user có quyền thực hiện filenm

-f filenm Đúng nếu filenm là regular file

-d filenm Đúng nếu filenm là thư mục

-c filenm Đúng nếu filenm là character special file

-b filenm Đúng nếu filenm là block special file

-s *filenm* Đúng nếu kích thước filenm khác 0

-t fnumb Đúng nếu fnumb (1 by default) là terminal device

Shorthand Method of Doing Tests

Bởi vì lệnh test là một trong những lệnh quan trọng bậc nhất trong shell để cho shell gần với các ngôn ngữ lập trinhf khác người ta đã thay test bằng bao đóng ([]).

Ví dụ:

```
$ int1=4
$ [ $int1 -gt 2 ]
$ echo $?
0
```

if-then

```
if command_1
then
        command_2
        command_3
fi
```

Nếu command_1 Thực hiện thành công thì command_2, command_3 mới được thực hiện tiếp theo.

if-then -else

```
if command_1
then
  command_2
  command_3
else
```

```
  command_4
  command_5
fi
```

if -then-elif

```
if command
then
  command
elif command
then
  command
elif command
then
  command
fi
```

Lệnh case

```
case value in
  pattern1)
    command
    command;;
  pattern2)
    command
    command;;
  ...
  patternn)
    command;
esac
```

Lệnh case chỉ thực hiện một lệnh tại một thời điểm nếu giá trị phù hợp với pattern, Các lệnh tiếp sau đó sẽ được thực hiện cho đến khi gặp (;;).

b) Lệnh lặp

Lệnh while : Thực hiện khi điều kiện còn đúng

```
while command
do
  command
  ...
  command
done
```

Lệnh until Thực hiện cho đến chi điều kiện đúng

```
until command
do
  command
  ...
  command
done
```

Lệnh for: Thực hiện lần lượt ứng các giá trị trong arg

```
for variable in arg1 arg2  ... argn
do
  command
  ...
  command
done
```

Lệnh break: Cho phép thoát khỏi vòng lặp.

c) Shell Functions

```
funcname ()
{
  command
  ...  _
```

```
  command;
}
```

d) Lệnh trap

Trong quá trình thực hiện các shell script có thể tạo ra nhiều các file tạm dùng trong quá trình thao tác dữ liệu. Tuy nhiên trong quá trình chạy không tránh khỏi các sự cố, hoặc các thao tác từ phía người sử dụng nhằm ngừng thực hiện tiến trình giữa chừng. Để có thể thực hiện việc xoá các file tạm này hoặc thực hiện các thao tác nào đó khi tiến trình bị ngừng thực hiện dùng lệnh trap.

trap command_string signals

Signal	Description
0	Shell exit
1	Hangup
2	Operator Interrupt
3	Quit
9	Kill
15	Software Termination (kill signal)

Ví dụ:

```
trap "rm $TEMPDIR/*$$; exit" 1 2 15
```

e) Thực hiện lệnh điều kiện với cấu trúc AND(&&) và OR (||)

Thông thường để thực hiện các lệnh theo điều kiện ta có thể sử dụng các lệnh trong lập trình shell để thực hiện. Tuy nhiên Shell cung cấp tổ hợp lệnh thực hiện điều kiện là && và ||.

command1&&command2

Trong tổ hợp lệnh này thì lệnh đầu tiên được thực hiện trước nếu quá trình thực hiện kết thúc hoàn thành (trả giá trị 0) thì lệnh tiếp sau đó mới được thực hiện. Tổ hợp trả giá trị đúng (0) khi các lệnh đều trả giá trị đúng (0)

command1||command2

Trong tổ hợp lệnh này thì lệnh đầu được thực hiện trước và nếu nó kết thúc có lỗi (khác 0) thì lệnh tiếp sau đó mới được thực hiện. Tổ hợp trả giá trị sai khi tất cả các lệnh đều trả giá trị sai (khác 0)

Debugging Shell Programs

Để lần bước theo các lệnh trong chường trình shell dùng lệnh.

sh -x <shell file>

Lệnh sẽ thực hiện từng lệnh trong file và hiện nó lên màn hình.

V. Starting Up and Shutting Down

1. Booting the System

Trước khi bạn có thể sử dụng máy tính của bạn, phải khởi động hệ điều hành. Quá trình khởi động hệ điều hành được gọi là booting. Khi hệ thống đã được khởi động thì các device, application, và service trên máy tính đã sẵn sàng cho việc sử dụng.

Bởi vì UNIX là một hệ điều hành đa nhiệm và đa người sử dụng, nên nhiều tiến trình được gọi thực hiện ngay ban đầu. Đầu tiên UNIX chạy phần khởi động hệ thống để đặt đồng hồ, cấu hình thiết bị và tạo UNIX kernel mới (nếu cần thiết). Sau đó hệ thống bắt đầu chạy các tiến trình tương ứng với các trang thái khởi động riêng biệt được phân ra trên hệ thống.

Trong hầu hết các hệ điều hành Unix việc khởi động hệ thống thường theo các trình tự sau:

- Xác định thiết bị boot.
- Nạp kernel từ thiết bị boot.
- Tìm và khởi động các thiết bị ngoại vi.
- Khởi tạo các tác vụ hệ thống cơ bản.
- Chạy các script mà các chương trình tạo ra để cung cấp các dịc vụ.

- Bắt đầu các ứng dụng khác.

Thông thường hệ thống chạy ngầm định là ở trạng thái 3. Trạng thái này sẵn sàng cho nhiều người sử dụng trong môi trường mạng (Bao gồm cả file sharing). Điều này có nghĩa là tất cả các hệ thống file được liên kết (mounted) trên hệ thống các tiến trình nền (daemon) được bắt đầu cho phép người sử dụng login, và quá trình điều khiển vào ra của mạng cũng được bắt đầu.

Ta có thể thực hiện các mức hoạt động khác nhau và các mức truy nhập khác nhau bằng các đặt trạng thái boot hoặc bằng các thay đổi trạng thái trong khi hệ thống đang chạy. Trạng thái 3 là full network/multiuser, trạng thái 1 và 2 là single-user/limited access.

Khi ta khởi động máy tình mà cài hệ điều hành UNIX, Phần khởi động ban đầu được quản lý bởi một tiến trình gọi là init. Init xử lý các tiến trình trong cách thức trên cơ sở trạng thái định nghĩa trong file /etc/inittab.

Tiến trình init khiểm tra và khởi động các tiến trình trong file /etc/inittab. Các tiến trình được coi như sysinit processe. Sysinit processes là các tiến trình đảm bảo cho hệ thống hoạt động một cách đùng đắn, chính xác.

Ví dụ:/etc/inittab:

cr::sysinit:/sbin/ckroot >/dev/sysmsg 2>&1

ck::sysinit:/sbin/setclk >/dev/sysmsg 2>&1

mm::sysinit:/etc/conf/bin/idmodreg >/dev/sysmsg 2>&1

ldmd::sysinit:/etc/conf/bin/idmodload >/dev/sysmsg 2>&1

ap::sysinit:/sbin/autopush f /etc/ap/chan.ap

bchk::sysinit:/sbin/bcheckrc </dev/console >/dev/sysmsg 2>&1

bu::sysinit:/etc/conf/bin/idrebuild reboot </dev/console >/dev/sysmsg 2>&1

ia::sysinit:/sbin/creatiadb </dev/console >/dev/sysmsg 2>&1

Chi tiết các lệnh như sau:

- ckroot — Đọc các tham số của mount cho root file system trong file /etc/vfstab. Các tham số bao gồm các kiểu file hệ thống, điều này là cần thiết tạo root file system sẵn sàng cho hệ thống. Ckroot cũng thực hiện việc kiểm tra hệ thống file (với lệnh fsck) nếu như nó xác định được là có vấn đề trong hệ thống file. Ví dụ như: Nếu ta tắt máy mà không thực hiện shutdown, hệ thống đưa ra thông báo kiểm tra hệ thống khi nó thực hiện việc giải quyết các sự cố mà nó tìm thấy.
- setclk— Đặt đồng hồ cho hệ thống UNIX.
- idmodreg— Nạp danh sách các kernel modules trong file /etc/mod_register.
- idmodload—Nạp danh sách các kernel modules trong file /etc/loadmods.
- autopush—Cấu hình một danh sách các modules được tự động đẩy vào các Streams device khi các device được mở. Danh sách các modules trong file /etc/ap/chan.ap được đẩy trên đỉnh của console monitor device để cung cấp các thông tin theo các dòng với kiểu quy định.
- bcheckrc—Khởi động một vài tác vụ bao gồm đặt tên hệ thống mounting /proc (processes) và /dev/fd (floppy disk) devices, thực hiện kiểm tra và liên kết các thiết bị thêm vào có liên quan đến floppy disks.
- idrebuild—Kiểm tra xem kernel có phải rebuilt hay không nếu có thì chạy lệnh idbuild để tạo lại nó. Kernel cần phải được tạo lại mỗi khi thêm vào thiết bị mới hoặc khi thayđổi các tham số.
- creatiadb—Thiết lập hệ thống bảo mật.

Khi mà chức năng khởi động hệ thống được thiết lập, init kiểm tra các mục khởi động ngầm định trong inittab để xác định mức chạy mà hệ thống.

Khái niệm System States

Trong Unix người quản trị hệ thống có thể thiết lập hệ thống máy tính lớn trong single-user mode, mà không có phần mạng hoặc terminals login. Trong trường hợp này người quản trị có thể kiểm tra xem xét hệ thống trước khi có yêu cầu khác ví dụ như cài đặt lại hệ thống. Hệ thống gồm các level sau:

- 0 - Là mức shutdown. Khi thay đổi sang mức 0 thì tất cả các tiến trình đều bị ngừng hoạt động.
- 1 (s or S)—Ứng với mức single-user. Có 3 trạng thái có thể thay đổi tới single-user là: 1, s, và S. Ta đặt hệ thống trong mức single-user nếu ta muốn không cho các người sử dụng khác truy nhập hệ thống. Điểm khác giữa mức 1, s, và S là: 1—Tất cả các hệ thống file vẫn được mount, tất cả các kết nối mạng bị ngắt bỏ, tất cả các tiến trình terminal đều bị ngắt bỏ. Mức s hoặc S—Đây là mức bắt đầu hệ thống Nếu không có file /etc/inittab. Nếu thay đổi đến trạng thái này, terminal của người sử dụng sẽ là system console, các terminal đều bị ngắt bỏ và các hệ thống file vẫn được mount. Khi hệ thống chuyển sang mức này chỉ có một số các các hệ thống file được mount ví dụ: /, /var, /proc...
- 2—Là mức nhiều người sử dụng. Mức này khởi động tất cả các script trong thư mục /etc/rc2.d, gồm nhiều tiến trình cho phép nhiều người sử dụng. Nếu muốn sử dụng hệ thống với hiệu lực mạng và môi trường nhiều người sử dụng thì phải chạy ở mức 2 (hoặc 3).
- 3—Là mức cho phép chia sẻ dữ liệu với các hệ thống ở xa. Nếu cài đặt NFS Hệ thống tự động thông báo và mount cá hệ thống file ở xa bằng NFS..
- 6—Là mức khởi động lại hệ thống. khi thay đổi sang mức 6 hệ thống shutdown và khởi động lại.
- Ngoài ra còn một số mức khởi động khác tuỳ theo loạ hệ điều hành Unix mà có các hỗ trợ với mục đivchs khác nhau.

Hệ thống có thể đặt ở các mức 1, s, S, 2, hoặc 3 làm mức ngầm định để chạy. Thông thường là mức 2 hoặc 3 trên các hệ thống Unix. Để thay đổi trạng thái của hệ thống sử dụng lệnh init (hoặc telinit).

Initialization Table (inittab)

File /etc/inittab có chứa các tiến trình mà được khởi động khi init thực hiện khởi động hệ thống hoặc khi thay đổi trạng thái. Một số thành phần trong inittab là chạy

dưới chế độ nền một số khác như /etc/rc2, được sử dụng để thiết lập các tiến trình khác cho mức chạy riêng.

Mỗi một thành phần trong file inittab bao gồm các trường sau:

idtag:runstate:action:process

- *Idtag* là một thẻ nào đó (từ 1–4 ký tự) xác định một mục. R*unstate* là trạng thái hệ thống mà thành phần sẽ chạy trong nó. Ta có thể có một vài trạng thái hệ thống gán cho một thành phần.
- *Action* là từ khoá tương ứng với một trong các : respawn (Nếu tiến trình bị mất thì tự bắt đầu lại), wait (Đợi tiến trình kết thúc trước khi thực hiện tiếp thành phần bên dưới, once (Chạy một tiến trình, đợi cho nó kết thúc và không khởi tạo lại), boot (Chạy tiến trình lần đầu chuyển sang trạng thái nhiều người sử dụng và không đợi tiến trình kết thúc), bootwait (chạy tiến trình lần đầu, chuyển sang trạng thái nhiều người sử dụng, đợi cho tiến trình kết thúc, và sysinit (chạy tiến trình khi hệ thống bắt đầu).
- *Process* là lệnh thực sự chạy khi các tiêu chuẩn trước đó là *runstate* và *action* được đáp ứng.

Ví dụ:

co:12345:respawn:ttymon g v p "Console Login: " d \

[cc]/dev/console l console

Mục này là co, chạy ở mức 1, 2, 3, 4, và 5. Nếu tiến trình bị mất đi thì nó tự động khởi động lại. Tiến trình này chạy lệnh ttymon(terminal monitor), mà cho phép ta thực hiện việc login từ system console.

Run State Directories (rc?.d)

Các ứng dụng mà cần có các tiến trình chạy chế độ nền hoặc yêu cầu một vài thành phần được khởi động khi hệ thống bắt đầu hoạt động, thông thường có các script trong thư mục xác định trạng thái chạy của hệ thống. Gồm các thư mục sau:

- /etc/rc0.d—có chứa các quan hệ script khởi động ban đầu với trạng thái

shutdown (0) và reboot (5 và 6).

- /etc/rc1.d— có chứa các quan hệ script khởi động ban đầu với trạng thái single-user (1, s, và S).
- /etc/rc2.d— có chứa các quan hệ script khởi động ban đầu với trạng thái multiuser (2 và 3).
- /etc/rc3.d— có chứa các quan hệ script khởi động ban đầu với trạng thái file-sharing (3).
- /etc/shutdown—Tương thích với các hệ preSystem V, Release 3 gồm các script được gọi chạy khi hệ thống shutdown. Thông thường thư mục này là rỗng ngoại trừ các script mà ta thêm vào.
- /etc/rc.d—Tương thích với các hệ preSystem V, Release 3 gồm các script chạy khi hệ thống khở động. Thông thường thư mục này là rỗng ngoại trừ các script mà ta thêm vào.
- /etc/init.d—Thực hiện như là nơi chứa các startup script. Các Script không thực sự chạy từ thư mục này, nhưng nó liên kết với các thư mục rc*?*.d tương ứng.

Startup Scripts

Startup script là lệnh chạy khi khởi động hệ thống, shutdown hệ thống hoặc khi thay đổi trạng thái hệ thống. Nếu thực hiện xem nội dung các file này bằng lệnh cat hoặc pg ta sẽ nhìn thấy hàng loạt các lệnh shell với các tuỳ chọn start hoặc stop.

Khi ứng dụng thêm vào một startup script. thì nó thêm script vào thư mục /etc/init.d. sau đó thực hiện link nó tới một hoặc nhiều thư mục với tên file bắt đầu bằng S (for start) hoặc K (for kill).

Ta hãy xem xét ví dụ về mouse manager:

Khi cài đặt UnixWare, một shell script cho việc khởi động và kết thúc tiến trình quản lý mouse trên ứng dụng giao diện đồ hoạ là file /etc/init.d/mse, được link thành hai file khác là /etc/rc2.d/S02mse và /etc/rc0.d/K02mse.

Các lệnh trong file script như sau:

```
case "$1" in
'start')
    /usr/lib/mousemgr &
    ;;
'stop')
    pid='/usr/bin/ps e | /usr/bin/grep mousemgr |\
        [cc]/usr/bin/sed e 's/^  *//' e 's/ .*//'"
    if [ "${pid}" != "" ]
    then
        /usr/bin/kill ${pid}
    fi
    ;;
*)
    echo "Usage: /etc/init.d/mse { start | stop }"
    ;;
esac
```

Khi khởi động hệ thống tiến trình init kiểm tra file /etc/inittab tìm các mục mà phù hợp với trạng thái chạy ngầm định, thông thường là trạng thái 3. Thực hiện việc tìm r2, chạy lệnh trong /sbin/rc2 và kiểm tra tất cả các script trong thư mục /etc/rc2.d. Sau đó chạy các file bắt đầu bằng K với tuỳ chọn stop và bắt đầu chạy các script mà bắt đầu bằng chữ S với tuỳ chọn start. Trong ví dụ trên thì lệnh chạy S02mse là: S02mse start. ứng với tuỳ chọn start là lệnh /usr/lib/mousemgr được thực hiện và tiến trình tiếp tục chạy cho đến khi có thay đổi lại trạng thái hệ thống.

Khi shutdown hệ thống tiến trình init chạy tiến trình trạng thái 0, cách thức giống

như trên. Nhưng mục chạy là r0, chạy lệnh /etc/rc0, thực hiện kiểm tra trong thư mục /etc/rc0.d. Tất cả các script bắt đầu bằng chữ K được gọi thực hiện với tuỳ chọn stop. Với script K02mse được gọi chạy như sau: K02mse stop. ứng với tuỳ chọn stop thì script chạy lệnh xá định ID của tiến trình mousemgr và huỷ bỏ nó, tương tự với tiến trình khác và sau khi thực hiện hoàn tất thì hệ thống có thể shutdown.

Thay đổi trạng thái với init hoặc telinit

Khi hệ thống đang chạy ta có thể thay đổi trạng thái hoặc mức hoạt động của hệ thống bằng lệnh init hoặc telinit. Nếu ta đang shutdown hệ thống hoặc chuyển sang trạng thái thấp hơn có thể sử dụng lệnh shutdown.

Lệnh init cho phép thay đổi trạng thái một cách dễ dàng bằng cách gõ lệnh init và theo sau là số chỉ trạng thái ví dụ: init 2

Lệnh telinit là link của init. Lệnh telinit được tạo ra cho người sử dụng.

2. Shutting Down the System

Có vài cách để shutdown hệ thống Unix: Bằng cách sử dụng lệnh Shutdown, reboot, bằng lệnh trong giao diện đó hoạ, bằng cách tắt máy ...

Using the *shutdown* Command

Lệnh shutdown có thể được sử dụng thay cho lệnh init để chuyển trạng thái sang (0) và trạng thái reboot (6). Lệnh có thể là phức tạp trong môi trường nhiều người sử dụng. Nếu dùng lệnh init 0 hệ thống bị down ngay. Nếu muốn người sử dụng phải logout ra hết trước khi hệ thống down có thể dùng lệnh:

cd /

shutdown y g60 i0

Tuỳ chọn y cho phép bỏ qua các câu hỏi yêu cầu g60 trong vòng 60 giây người sử dụng phải logout trước khi hệ thống down. i0 gán với trạng thái lệnh init 0.

Khi chạy lệnh này thì tất cả người sử dụng trên mạng sẽ được thông báo phải logout hệ thống sắp shutdown và dành khoảng thời gian cho người sử dụng hoàn thành

công việc cuối cùng trước khi hệ thống down.

VI.Managing processes

1. Processes

Sau khi hệ thống hoàn thành việc khởi động hệ thống ta có thể thực hiện chạy các ứng dụng. Một ứng dụng đang thực thi gọi là một process. Công việc của hệ điều hành là quản lý thực thi ứng dụng. Khi thực thi một chương trình thì hệ điều hành sẽ tạo ra một process mới. Nhiều process có thể cùng đồng thời tồn tại, nhưng chỉ một process có thể được thực hiện thực sự trên CPU tại một thời điểm. Hệ điều hành phân chia việc thực thi các process một các rất nhanh làm cho các process như đang thực hiện đồng thời. Khái niệm này xem như là sự phân chia thời gian xử lý hoặc đa nhiệm.

Khi thoát khỏi chương trình thì process sẽ kết thúc và hệ điều hành sẽ giải phóng các tài nguyên mà chương trình đã sử dụng.

Hầu hết các chương trình đều thực hiện một vài nhiệm vụ từ lúc khởi đầu cho đến lúc kết thúc cho nên để thực thi các tác vụ chương trình yêu cầu hệ điều hành cung cấp các tài nguyên cần thiết cho việc thực thi.

Có một vài loại process có trong các hệ điều hành Unix. Mỗi loại có các đặc điểm riêng gồm: Interactive process là process được khởi động bởi shell nó có thể là foreground hoặc background. Batch process là process mà không tương ứng với terminal. Daemon process là process mà chạy background khi được yêu cầu. Loại này thường thực hiện khi khởi động hệ thống.

Một số lệnh liên quan đến việc quản lý tiến trình:kill, ps

Phần tham số của lệnh xem trong end user.

Ví dụ:

$ ps -f

UID PID PPID C STIME TTY TIME COMD

sartin 1400 1398 80 18:31:32 pts/5 0:01 -sh

sartin 1406 1400 25 18:34:33 pts/5 0:00 ps –f

Lệnh ps cho phép người quản trị xem xét các thông tin về hệ thông báo gồm các thông số sau:

F Chỉ ra trạng thái của process và được tính toán bởi các giá trị hexadecimal gồm:

- 00 Process đã kết thúc.
- 01 Là system process luôn tồn tại trong memory
- 02 Process đang bị kiểm soát bởi tiến trình cha.
- 04 Process đang bị kiểm soát bởi tiến trình cha và nó đã bị dừng.
- 08 Process không thể được kích hoạt bởi các signal
- 10 Process đang trong bộ nhớ và bị lock đang đợi event
- 20 Process không thể bị swapped

S Cũng dùng chỉ trạng thái của process:

- O Process đang chạy trên processor.
- S Process đang sleeping, và đang đợi I/O event để hoàn thành.
- R Process đang sẵn sàng chạy.
- I Process không làm gì.
- Z Process đã bị kết thúc và tiến trình cha không đợi nhưng nó vẫn đang trong process table (zombie process)
- T Process đã bị ngừng bởi tiến trình cha.
- X Process đang đợi để lấy thêm bộ nhớ.

UID User ID của người chủ process

PID Process ID number

PPID Parent process ID number

C Sử dụng CPU theo thời gian biểu.

CLS Lớp thời gian biểu, real-time, time sharing, hoặc system

PRI Mức ưu tiên Process (số càng lớn độ ưu tiên càng nhỏ).

NI Mức độ ưu tiên về chiếm dụng thời gian xử lý CPU. Tăng giá trị -> giảm độ ưu tiên.

SZ Tổng virtual memory yêu cầu bởi process.

Wchan Địa chỉ của process trong process table.

TTY Terminal khởi động process, hoặc cha nó. (A ? chỉ ra không có terminal tồn tại.)

TIME Tổng thời gian process sử dụng CPU từ khi process bắt đầu.

COMD Lệnh tạo ra process

2. Process scheduling

Thông thường các hệ thống lớn hoặc cần cung cấp dữ liệu thường xuyên hệ thống luôn được chạy 24/24. UNIX đưa ra một số lệnh cho phép thực hiện các process theo thời gian định sẵn.

Lệnh at: Lệnh at được sử dụng để để đặt schedule cho một lệnh thực hiện trong thời gian qui định.

at time date < file

Việc đặt lịch theo khoảng thời gian nào đó tuỳ theo người đặt qui định. Có thể là hh:mm, có thể hh:mm(pm,am). Có một số từ về thời gian dùng làm option là noon, midnight, now, next. Có thể đặt ngày, tháng thực hiện May 10 hoặc day of the week .

Ví dụ:

at 20:30 < reorg.data

at 8:30 pm < reorg/data

at 20:30 today < reorg.data

at 8:30 pm Friday < reorg.data

at 0900 Monday next week < reorg.data

Dùng lệnh at -l để hiện danh sách các các process được schedule.

Ví dụ:

$ at -l

user = tparker job 827362.a at Wed Aug 31 06:00:00 EDT 1995

user = tparker job 829283.a at Wed Aug 31 09:30:00 EDT 1995

Dùng lệnh at –r <job> để bỏ một schedule.

Ví dụ:

at -r 2892732.a

Lệnh cron và crontab: cron là một tiện ích cho phép thực hiện các lệnh tại một thời điểm chỉ định mà không cần một ai trực tiếp khởi động nó. Hệ thống UNIX tự động load cron như là một daemon khi nó khởi động. Khi hoạt động cron đọc thời gian và công việc mà nó được định thực hiện trong crontab file. Việc thực hiện các tác vụ đặt trong crontab file là luôn luôn được thực hiện nó chỉ ngừng thực hiện khi kết thúc tiện ích cron hoặc khi thay đổi thông tin trong crontab file.

Trên hầu hết các hệ thống việc truy nhập và tạo các schedule chỉ được thực hiện bởi người quản trị hệ thống.

Ngày nay một số hệ điều hành UNIX còn cho phép người sử dụng tự tạo các crontab của riêng họ.

Để tạo crontab file sử dụng lệnh crontab với tham số -l. Nó sẽ tạo ra một crontab file cho phép tạo các tác vụ mong muốn.

$ crontab -l > new_crontab_file

$ vi new_crontab_file

[edit the file to update your crontab]

$ crontab new_crontab_file

$

Một crontab file được thể hiện dưới khuôn dạng sau:

<minute> <hour> <day-of-month> <month-of-year> <day-of-week> <command>

Giá trị chấp nhận của các trường:

minute (0-59)

hour (0-23)

day of month (1-31)

month of year (1-12)

day of week (0-6, 0 is Sunday)

Command (rest of line)

Ví dụ

0 17 1 * 0 date | mail user

Lệnh sẽ được thực hiện vào 5 p.m. vào ngày đầu tháng và 5 p.m. mỗi ngày chủ nhật.

Các file cron tab được đặt trong thư mục cron để thực hiện. Thư mục này có thể ở một vị trí nào đó tuỳ loại UNIX thông thường là /usr/spool/cron/ hoặc /var/spool/cron/.

3. Process priorities

VII.Security

Việc truy nhập hện thống thông thường được thực hiện trực tiếp tự system console, qua hệ thống mạng, qua modem connection. Việc truy nhập hệ thống UNIX đòi hỏi người sử dụng phải được tạo trên hệ thống với mật khẩu truy nhập và quyền truy nhập tương ứng tới tài nguyên hệ thống.

1. Security datafiles

Để xác định quyền hạn truy nhập hệ thống, Với các hệ thống UNIX thông thường các thông tin liên quan đến nhóm và người sử dụng truy nhập hệ thống được lưu trữ trong các file (security datafiles).

password File

File này dùng lưu trữ thông tin về người sử dụng hệ thống bao gồm các trường riêng rẽ cách nhau bởi dấu (:), khuôn dạng file gồm các dòng có dạng như sau. File được đặt trong thư mục /etc:

username:pswd:uid:gid:uid comments:directory:shell

- Trường *username* là tên user thực hiện khi login tại dấu nhắc Unix login:. Thông thường trường này là gồm các ký tự chữ thường nhỏ hơn hoặc bằng 8, tên duy nhất, không được chứa dấu (:, dấu cách, ký tự đặc biệt). Cách tốt là dùng dấu gạch dưới (_) cho dấu nối.
- Trường *pswd* là phần mật khẩu với các khuôn dạng khác nhau. Nó có thể mang giá trị rỗng chỉ ra rằng không yêu cầu mật khẩu khi login. Giá trị có thể dài tới 13 ký tự và được mã hoá. Các ký tự được gõ vào là các ký tự nằm trong khoảng { . / *0-9 A-Z a-z* } còn các giá trị khác là không được chấp nhận. Đối với một số loại của hệ điều hành UNIX thì có thể được mở rộng hơn hoặc được liên kết với một file khác (shadown).
- Trường *uid* là id của user. Giá trị này là duy nhất và có giá trị từ 0- 65535. Một số loại hệ điều hành có khuyến nghị cách dùng các id trong phạm vi an toàn. Chúng bao gồm:

  0: The superuser

  1-10: Daemons and pseudo users

  11-99: System, reserved and "famous" users

  100+: Normal users

  60001: "nobody" (occasionally 32000 or 65534)

60002: "noaccess" (occasionally 32001)

- Trường *gid* là id của nhóm ngầm định mà user thuộc vào. Giá trị này tương ứng với giá trị có trong */etc/group* file.
- Trường *uid comments field* là trường ghi chú các thông tin thêm vào các phần thông tin có thể cách nhau bằng dấu (,) ví dụ: Homer User,,,800-IAM-HOME.
- Trường *directory* là thư mục home hoặc thư mục làm việc của người sử dụng.
- Trường *shell* là trường chứa bộ thông dịch lệnh hoặc chương trình người sử dụng gọi tới khi login. Thông thường là một trong ba shell sau: sh (Bourne), ksh (Korn), csh. Nếu không là các shell thì nó có thể là một chương trình nào đó

Ví dụ :

root:x:0:0:Superuser:/:

daemon:*:1:5::/:/sbin/sh

bin:*:2:2::/usr/bin:/sbin/sh

sys:*:3:3::/:

adm:*:4:4::/var/adm:/sbin/sh

uucp:*:5:3::/var/spool/uucppublic:/usr/lbin/uucp/uucico

lp:*:9:7::/var/spool/lp:/sbin/sh

nuucp:*:11:11::/var/spool/uucppublic:/usr/lbin/uucp/uucico

hpdb:*:27:1:ALLBASE:/:/sbin/sh

nobody:*:-2:60001::/:

dave:x:100:10:Dave G,13,x3911,unlisted:/usr1/dave:/bin/tcsh

charlene:x:101:10:Charlene G,14,x1800,unlisted:/usr1/charlene:/bin/tcsh

john:x:102:60:John S,2,555-1234,x1400:/usr2/john:/bin/ksh

georgia:x:103:60:Georgia S,11,x143,x143:/usr2/georgia:/bin/csh

Shadow Password File

File **/etc/passwd** thông thường người sử dụng có thể xem được. Để lưu trữ mật khẩu của người sử dụng đã được mã hoá và một số thông tin khác, một số loại UNIX sử dụng file */etc/shadow*. Thông thường một file shadow có dạng sau:

username:pswd:lastchg:min:max:warn:inactive:expire:flag

- Trường *username* là tên user lấy từ file /etc/passwd.
- Trường *pswd* có chứa 13 ký tự mã hoá của password, null chỉ ra không có password khi login.
- Trường *lastchg* là ngày mà password thay đổi cuối.
- Trường *min* là ngày nhỏ nhất giữa ngày thay đổi password.
- Trường *max* là số ngày lớn nhất mà password sẽ được chấp nhận.
- Trường *warn* có chứa số ngày thông báo trước khi password bị quá hạn.
- Trường *inactive* là trường số ngày mà username vẫn còn tác dụng trước khi không được cho phép login.
- Trường *expire* chỉ số ngày xác định mà người sử dụng được quyền login vào hệ thống.
- Trường *flag* hiện không sử dụng.

Group File

Đây là file liên quan đến quyền hạn của người sử dụng, của nhóm trên hệ thống file trong hệ điều hành Unix. Cấu trúc file như sau:

group_name:password:group_id:list

- Trường *group_name* chứa tên của group.
- Trường *password* là phần mã hoá của mật khẩu nhóm nếu có.
- Trường *group_id* là giá trị ID của group.
- Trường *list* là danh sách các ID của người sử dụng thực nhóm.

Ví dụ:

root::0:root
other::1:root,hpdb
bin::2:root,bin
sys::3:root,uucp

adm::4:root,adm

daemon::5:root,daemon

mail::6:root

lp::7:root,lp

tty::10:

nuucp::11:nuucp

users::20:root,dave,charlene,john,georgia,operator,steve,judy,wayne,jamie

nogroup:*:-2:

systech::110:dave,disdb,diskf,disjs,dispm,diskj

dba::201:oracle,john,kathy,pete

Hosts.equiv: Chỉ ra danh sách các host và các user khi truy nhập hệ thống với các lệnh rlogin, rcp, rsh khi truy nhập không cần mật khẩu.

Cấu trúc file như sau:

host1

host2 user

+@group1

@group được lưu trong file netgroup

netgroup: Chứa các group

netgroup name name ...

netgroup (hostname, user, domain)

.

.rhosts: Tương tự như trên

2. Group and User administration

a) Group administration

Tạo group

Xoá group

b) User administration

Tạo người sử dụng

Có ba cách để tạo mọt người sử dụng hệ thống là:

- Soạn file passwd, shadow và các file tương ứng bằng tay.
- Sử dùng lệnh useradd.
- Sử dụng các tiện ích Graphical User Interface (GUI) được các hệ điều hành UNIX hỗ trợ.

Trước khi tạo một người sử dụng ta cần phải:

- Chọn tên người sử dụng. Tên này phải duy nhất trong hệ thống.
- Gán user ID (giá trị này chưa được sử dụng trước đây) và Group mà user thuộc vào.
- Chọn group mag user sử dụng nó làm primary group.
- Chọn thư mục home của người sử dụng.

Lệnh Useradd : Lệnh tạo người sử dụng.

useradd [-c comment] [-d dir] [-e expire] [-f inactive]

[-g group] [-G group [, group...]] [-m [-k skel_dir]]

[- u uid [-o]] [-s shell] login

- -c Ghi chú Ví dụ: -c "Temp user"
- -d Thư mục home của user.
- -e Ngày quá hạn trường này không yêu cầu phải có. Nếu có thì việc kiểm tra

ngày quá hạn sẽ có hiệu lực. Ví dụ: -e "January 1, 1995" hoặc -e 1/1/95

- -f Chỉ ra số ngày không hoạt động tối đa trước khi không được chấp nhận. Nếu không có tuỳ chọn này thì việc kiểm tra bị bỏ qua.
- -g group—Chỉ ra primary group ID của user.
- -G group [, group ...]—Chỉ ra group mà user thuộc vào. (secondary groups). Giá trị có thể là tên hoặc group ID
- -m [-k skel_dir]—Nếu không có tham số –k -> tạo thư mục home cho user và copy các file từ /etc/skel. Nếu có tham số -k -> tạo thư mục home và copy các file tự skel_dir thay cho ở /etc/skel.
- -u uid [-o]—Dặt user ID. Nếu có -o thì user ID này là không duy nhất. Tuy nhiên UNIX không khuyến nghị dùng nhiều user chung một ID.
- -s shell—Chỉ login shell.
- login—Chỉ tên người sử dụng dùng để login vào hệ thống.

Xoá người sử dụng

Trước khi xoá một người sử dụng ta cần quan tâm xem thực sự muốn xoá người sử dụng hay chỉ cần tạm thời làm mất hiệu lực của người sử dụng. Xem xét các lý do sau trước khi quyết định.

- Nếu user có khả năng sẽ được quyền quay trở lại hệ thống, việc sử dụng lại dữ liệu sẽ khó và còn tránh việc sử dụng ID đã sử dụng vì khi đó các dữ liệu do người bị loại bỏ sẽ được gán lại cho người mới tạo.
- Cần thiết trong trường hợp hồi phục các file dữ liệu liên quan đến người sử dụng đã bị xoá. Nếu dữ liệu hồi phục mà có của user đã bị xoá thì sau đó ta khó xác định dữ liệu này là của ai -> việc sử dụng lại sẽ rất khó khăn.

Để tạm thời không cho user thâm nhập hệ thống dùng lệnh *passwd -l <user>*

Muốn loại bỏ hoàn toàn những gì liên quan đến người sử dụng thì trước hết cần:

- Tìm các file mà người sử dụng là chủ và in dah sách các file này ra. (Tìm bằng lệnh find).

- Muốn sử dụng lại các file này hãy gán chủ sở hữu mới.
- Xoá toàn bộ các file cần phải xoá.
- Xoá user bằng các xoá các thông tin liên quan đến user trong các security file, hoặc dùng tiện ích mà hệ điều hành UNIX cung cấp hoặc dùng lệnh userdel.

Lệnh userdel: Lệnh xoá người sử dụng.

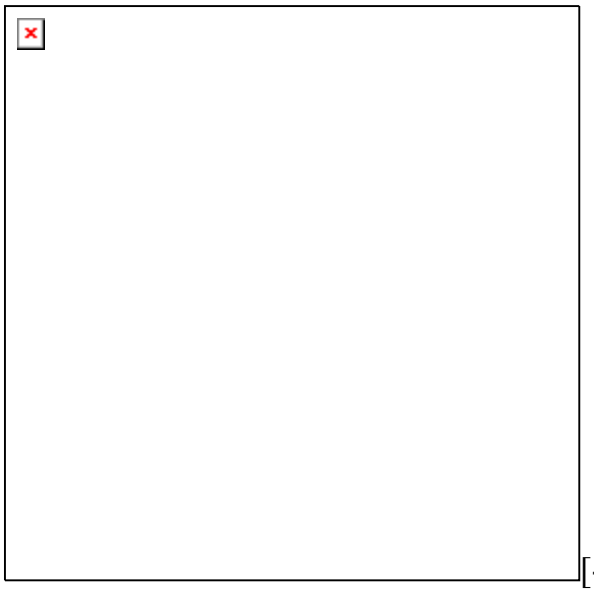
userdel [-r] <username>

- username Tên user cần xoá.
- -r Xoá toàn bộ thư mục home của user.

Thay đổi thông tin người sử dụng.

Để thay đổi thông tin liên quan đến người sử dụng dùng lệnh.

usermod [-c uid comment] [-d dir [-m]] [-e expire] [-f inactive]

 [-g gid] [-G gid[,gid]]

[-l new_username] [-s shell] [-u uid [-o]] username

new_username Là tên user mới.

Psuedo Users

Trong mỗi hệ điều hành UNIX đều có chứa mộy vài psuedo user. Các user này

được hệ thống sử dụng nhằm một số mục đích nào đó. Các thuộc tính của chúng không nên thay đổi. Sau đây là một số các psuedo user :

- daemon Used by system server processes
- bin Owns executable user command files
- sys Owns system files
- adm Owns accounting files
- uucp Used by UUCP
- lp Used by lp or lpd subsystems
- nobody Used by NFS

Thiết lập môi trường người sử dụng

Khi người sử dụng login vào hệ thống một số file được gọi thực hiện để xác lập môi trường làm việc cho người sử dụng.

/etc/profile File thiết lập môi trường hệ thống được gọi thực hiên đối với tất cả người sử dụng login vào hệ thống.

Các file ẩn trong thư mục home của người sử dụng.

- .login Dùng với Csh thực hiện sau quá trình lập môi trường hệ thống.
- .profile Dùng Sh với ksh Thực hiện khi người sử dụng login sau quá trình thiết lập môi trường của hệ thống.
- .rhosts Danh sách Remote host/username mà được gán quyền khi sử dụng Rlogin, rexec, rsh, rcp, truy nhập các file và lệnh không cần mật khẩu.
- .mailrc Khởi động file cho mail mà cho phép đặt các tuỳ chọn hoặc các aliases.
- .xinitrc File khởi động cho X windowing.
- .xsession
- .xdefault

3. System access permissions

Xem tài liệu end user

4. Acounting

UNIX accounting system thu thập thông tin về việc sử dụng tài nguyên máy tính của các nhóm hoặc của người sử dụng đơn lẻ. Ta có thể sử dụng thông tin này giống như các bảng kê với người sử dụng tài nguyên hệ thống. Các Accounting report cung cấp các thông tin giúp cho người quản trị hệ thống xem xét tài nguyên hệ thống đang được sử dụng, quản lý tài nguyên, đặt các giới hạn và hạn mức truy nhập tài nguyên ...

Các tiện ích được cung cấp có thể đặt chạy tự động hoặc thực hiện banừg tay từ người quản trị hệ thống.

Khi hệ điều hành UNIX được khởi động và hệ thống accounting được khởi động chạy thì các thao tác thống kê bắt đầu hoạt động. Dữ liệu được tập hợp theo các loại sau đây:

- Connect session statistics
- Process usage
- Disk space utilization
- Printer usage

Connect Session Statistics

Thống kê thời gian bắt đầu thực hiện kết nối tới hệ thống và thời gian kết thúc kết nối với hệ thống của một user nào đó. Các thông tin này thường được ghi trong file /var/adm/wtmp, gồm các thông tin sau:

- Tên người sử dụng
- Ngày login/logout
- Thời gian login/logout
- Terminal port

Các thông tin này có thể được sử dụng để đưa ra các loại báo cáo dạng sau:

- Ngày và giờ của mỗi lần kết nối.
- Tên và ID của người sử dụng thực hiện kết nối.
- Thời gian kết nối.
- Địa chỉ thiết bị kết nối

Process Usage

System accounting hầu như thống kê các thông tin bởi các tiến trình đơn lẻ ví dụ:

- Sử dụng bộ nhớ.
- Số User và group numbers chạy process
- Tên của lệnh gọi thực hiện.
- Thời gian chạy và thời gian processor sử dụng bởi process
- Tạng thái vào ra.
- Số dữ liệu được truyền.
- Số lương blocks đọc và ghi trên đĩa của mỗi process

Các thông tin được lưu trong accounting file /var/adm/pacct. File này được truy nhập bởi các lệnh accounting. Sau khi tiến trình kết thúc kernel ghi các thông tin trên file /var/adm/pacct file. Thông tin gồm:

- Process của user ID
- Lệnh thực hiện khởi động process
- Thời gian thực hiện.

System accounting cung cấp lệnh hiển thị, báo cáo, tổng kết các thông tin về tiến trình.

Disk Space Utilization

System accounting cung cấp khả năng cho người quản trị hệ thống để giám sát việc sử dụng đĩa của người sử dụng. Để hạn chế việc sử diụng đĩa người sử dụng có thể

thực hiện việc giới hạn việc sử dụng. Các lệnh này thực các chức năng sau:

- Sử dụng địa bởi các filesystem
- Báo cáo về việc sử dụng đĩa của người sử dụng.
- Trạng thái đĩa và sự sử dụng đĩa của các lệnh system accounting.

Printer Usage

Printer usage data được lưu trong file /var/adm/qacct dưới dạng ASCII. Tiến trình qdaemon sẽ ghi dữ liệu dạng ASCII tới file /var/adm/qacct sau khi công việc in ấn hoàn tất. Các bản ghi ứng với mỗi printer queue có chứa các thông tin sau:

- Tên và ID của người sử dụng.
- Số trang đã in.

VIII.File System and Disk Administration

1. Cấu trúc thư mục trên Unix

- / - Thư mục gốc trên UNIX file system.
- /bin - Là symbol link tới /usr/bin chứa các lệnh user trên UNIX.
- /dev - Có chứa các file thiết bị như printer, keyboard, harddisk ...
- /etc - Chứa các file cấu hình hệ thống và các file liên quan đến quả trị hệ thống.
- /lib - Chứa thư viện trên UNIX.
- /sbin - Chứa lệnh liên quan đến khởi tạo hệ thống.
- /tftpboot - Chứa các file phục vụ cho việc khởi động từ các client.
- /usr - Chứa lệnh và các chương trình ứng dụng hỗ trợ bởi hệ điều hành.
- /var - Chưa các thông tin cấu hình các ứng dụng, hàng đợi ...
- /vmunix Kernal của UNIX
- /opt Chứa các chương trình ứng dụng thường từ hãng thứ ba.

- /home Thư mục home của user.
- /lost+found Chưa các file được recover bởi fsck.

2. Creating file systems

Để tạo hệ thống file cần thực hiện các bước như sau:

- Chọn quyền hạn của đĩa.
- Tạo các partition
- Tạo file system

Mỗi loại UNIX sẽ có các công cụ khác nhau để tạo các file system. Thông thường dùng các lệnh fdisk, format, fdformat, các lệnh tạo cấu trúc block dùng mkfs hoặc newsfs. Mỗi loại hệ điều hành UNIX sử dụng ký hiệu địa diện cho hệ thống file trong các mục tuỳ chọn ứng với các lệnh Ví dụ: Solaris sử dụng ufs, Linux sử dụng ext2, IRIX sử dụng efs và xfs.

Tuy nhiên hệ thống file của UNIX cũng chỉ là nơi lưu dữ liệu trên đĩa và nó cũng được lưu dưới dạng cấu trúc phân cấp và đặt trên các partition.

Với hệ điều hành UNIX các thiết bị đều được thể hiện dưới dạng các file. Các file thiết bị này thường được đặt trong thư mục /dev. Với mỗi hệ điều hành UNIX các file thiết bị này có thể được ký hiệu khác nhau hoặc đặt mức thư mục thấp hơn. Thông thường các hệ điều hành UNIX tự động tạo đúng các file thiết bị mà nó hỗ trợ khi khởi động hệ thống. Ngầm định các file này chỉ được truy nhập bởi người quản trị hệ thống (root).

Với các thiết bị lưu trữ như đĩa cứng tuỳ theo từng loại hệ điều hành va loại đĩa cứng mà có các ký hiệu quy định khác nhau về tên file:

Ví dụ với đĩa IDE, EIDE trên **Linux** thể hiện dưới dạng file sau

/dev/hd[drive][partition]

Với đĩa đầu tiên ký hiệu là hd a cho primary disk và b cho slave, c cho primary secondary disk và d cho slave secondary. Tiếp sau đó là giá trị số ứng với các partition trên mỗi đĩa.

Với đĩa SCSI thay ký hiệu tên filư là /dev/sd thay cho /dev/hd còn các thành phần khác tương tự.

Tạo partition: Để tạo partition dùng lệnh fdisk để tạo partion. (ví dụ với Linux)

fdisk /dev/hda

Các lệnh trong fdisk

p Hiện partition hiện thời

n Tạo partition mới gồm extended và primary partition (1-4).

t Tạo swap partition

w Ghi lại các thay đổi vừa tạo.

Tạo File Systems: Sau khi đã tạo partition thì hệ thống file vẫn chưa được sẵn sàng cho việc sử dung. mà cần phải tạo file system. Để tạo hệ thống file trong Linux dùng lệnh mke2fs (trong sun solaris dùng lệnh newfs), mkswap tạo swap file system.

Ví dụ

mke2fs /dev/hda1

mkswap /dev/hda2

3. Mounting and unmounting file systems

Mounting file systems

Như ta đã biết hệ thống file của UNIX được lưu trữ dưới dạng các cây thư mục nhưng muốn thực hiện được điều này thì nó phải được mount.

Trước khi có thể mount file hệ thống ta cần chọn một điểm mount. Điểm mount là một thư mục trong hệ thống file nơi mà thư mục gốc bao lên nó. UNIX giữ điểm mount và cho phép truy nhập đến hệ thống file dựa trên quyền hạn của người sử dụng hiện thời. Điểm mount có thể là một vị trí nào đó trên cây thư mục. Cần chú ý rằng một hệ thống file chỉ được mount trên một một thư mục.

Để mount hệ thống file dùng lệnh mount cú pháp như sau:.

mount <option> </dev/device> </directory to mount>

- /dev/device là tên của device mà ta muốn mount.
- /directory to mount là thư mục mount mà hệ thống file mount tới

Các tham số tuỳ chọn của option –o có thể là:

- rw read/write
- ro read only
- bg background mount (Nếu mount bị lỗi nó chuyển sang background và tiếp tục cố gắng mount cho đến khi hoàn thành).

Ví dụ: mount -o rw /dev/hda4 /usr

Lệnh mount không tham số sẽ hiện lên tất cả các file systems đang được mount

Chi tiết của lệnh xem tài liệu ứng với loại hệ điều hành UNIX.

Unmounting file systems

Để bỏ mount hệ thống file dùng lệnh umount hoặc umountall

Lệnh umount: Bỏ mount một hệ thống file (điểm mount).

umount <mount point>

Ví dụ:

umount /usr

Lệnh umountall: Bỏ tất cả các điểm mount

Automount file systems

Automount là khả năng tự động mount hệ thống file tại thời điểm khởi động hệ điều hành. Với khả năng tự động cho phép hệ thống sẵn sàng khi quá trình khởi động kết thúc. Để thực hiện được việc tự động mount UNIX sử dụng file đặc biệt là /etc/fstab (/etc/vfstab dưới Solaris). File này chứa danh sách tất cả các partition cần mount tại thời điểm khởi động và thư mục cần được mount tới với các tuỳ chọn kèm theo theo dạng sau:

/dev/device /dir/to/mount ftype parameters fs_freq fs_passno

- dev/device Chỉ device sẽ được mount.
- /dir/to/mount Là thư mục được mount tới trên cây thư mục.
- ftype Là kiểu hệ thống file. Ví dụ dưới Solaris là ufs, dưới Linux là ext2, nfs cho NFS , swap cho swap partitions, và proc cho /proc file system.
- parameters Là biến tuỳ chọn khi mount ứng với –o option.
- fs_freq Được sử dụng bởi dump để xác định hệ thống file cần được dump.
- fs_passno Được sử dụng bởi chương trình fsck để xác định trình tự kiểm tra đĩa tại thời điểm khởi động.

Các dòng bắt đầu bằng dấu # là các dòng chú thích.

Ví dụ về file fstab dưới Linux systems:

```
#
# Sample /etc/fstab file for a Linux machine
#
# Local mounts
/dev/sda1    /          ext2       defaults    1    1
/dev/sda2    /usr       ext2       defaults    1    1
/dev/sda3    /usr/data  ext2       defaults    1    1
/dev/cdrom   /cdrom     iso9660    ro          1    1
/dev/sda4    /dos       msdos      defaults    1    1
/dev/sdb1    /data      ext2       defaults    1    1
```

Một số kiểu hay dùng trong Linux

- ext2 Kiểu filesystem được dùng chủ yếu trên Linux partition.

- iso9660 Kiểu ISO 9660 filesystem được dùng với CD-ROM disks.
- sysv Kiểu Nhằm hỗ trợ cho dạng UNIX System V filesystem.
- msdos Kiểu DOS partition mà Linux có thể truy nhập.
- hpfs Kiểu High Performance filesystem bởi Linux.

4. Managing disk use

Để biết các thông tin về việc sử dụng đĩa UNIX dùng các tiện ích sau:

Lệnh df: Cho biết thông tin về việc sử dụng đĩa, dung lượng đã được sử dụng và chưa được sử dụng và theo tỷ lệ phần trăm.

df <option>

Các tham số thường dùng:

- -a Hiển thị tất cả các partition bao gồm cả swap và /proc.
- -i, Hiển thị thông tin inode thay cho block.
- -k Hiển thị dạng KB.
- -t<type>Hiển thị chỉ những filesystems có kiểu phù hợp với type chỉ định.

Ví dụ: df –t

Filesystem	1024-blocks	Used	Available	Capacity	Mounted on
/dev/hda3	247871	212909	22161	91%	/
/dev/hda6	50717	15507	32591	32%	/var
/dev/hda7	481998	15	457087	0%	/local
server1:/var/spool/mail	489702	222422	218310	50%	/var/spool/mail

Các cột thông tin gồm:

- Filesystem Chỉ file system
- 1024-blocks Chỉ số block trong file system. (Tổng dung lượng.)

- Used Số block đã sử dụng.
- Available Số block chưa sử dụng.
- Capacity Phần trăm lượng đĩa đang sử dụng hiện tại.
- Mounted on Chỉ vị trí được mount trên cây thư mục.

Lệnh du: Hiển thị tổng đĩa sử dụng trên từng thư mục hoặc từng file.

du <option> <directory>

Các tham số thường dùng:

- -a Hiển thị số đếm trên tổng các file và các thư mục.
- -b Hiển thị kích thước dạng byte.
- -c Hiển thị tổng cục bộ.
- -k Hiển thị kích thước dạng KB.
- -l Hiển thị kích thước của tất cả các file.
- -s Chỉ hiện số tổng.
- -x Bỏ qua các filesystem khác mà mount vào trong thư mục hiện thời.

Ví dụ: du

409 ./doc

945 ./lib

68 ./man

60 ./m4

391 ./src

141 ./intl

873 ./po

3402 .

Chỉ số lượng block được sử dụng bởi mỗi thư mục.

5. Checking file system integrity

Trong quá trình hoạt động hệ thống không tránh khỏi diễn ra trạng thái một ai đó đột ngột ngắt điện máy tính, hệ thống cấp điện cho máy bị mất, một ai đó nhấn nút reset .... các hiện tượng này sẽ dẫn đến hiện tượng thông tin trên hệ thống vẫn còn tồn tại mà chưa được ghi ra đĩa. Khi xảy ra các lỗi này thì ta cần phải thực hiện việc kiểm tra tính toàn vẹn của hệ thống điều này là cần thiết nếu như cấu trúc hệ thống là không còn đúng. Để thực hiện việc kiểm tra và sửa chữa các lỗi này dùng lệnh fsck.

Lệnh fsck thự hiện theo nhiều giai đoạn. Mỗi giai đoạn thực hiện một nhiệm vụ riêng và các giai đoạn sau đều dựa vào kết quả thực hiện ở giai đoạn trước đó.

fsck thực hiện duyệt bắt đầu từ superblock liên quan đến các vùng disk blocks, pathnames, directory connectivity, link reference counts, và vùng trống của blocks và inodes.

Các giai đoạn kiểm tra của lệnh fsck:

Phase 1: Kiểm tra Block và Size giai đoạn này kiểm tra danh sách inode, tìm các inode bị vô hiệu. Lỗi này được thông báo như sau:

UNKNOWN FILE TYPE I=inode number (CLEAR)

PARTIALLY TRUNCATED INODE I=inode number (SALVAGE)

block BAD I=inode number

block DUP I=inode number

Phase 2: Kiểm tra đường dẫn, giai đoạn này xoá bỏ các thư mục từ các bad inodes được tìm thấy ở giai đoạn 1 và tiến hành kiểm tra thư mục với các con trỏ inode mà bị vượt quá phạm vi hoặc con trỏ trỏ tới bad inode. Lỗi này được thông báo như sau:

ROOT INODE NOT DIRECTORY (FIX?)

I=OUT OF RANGE I=inode number NAME=file name (REMOVE?)

UNALLOCATED I=inode number OWNER=O MODE=M SIZE=S MTIME=T

TYPE=F

(REMOVE?)

BAD/DUP I=inode number OWNER=O MODE=M SIZE=S MTIME=T TYPE=F (REMOVE?)

Phase 3: Kiểm tra kết nối, giai đoạn này phát hiện lỗi ở các thư mục không tham chiếu, bằng cách tạo thư mục lost+found nếu cần và chuyển các phần không đúng vào thư mục lost+found.

Phase 4: Kiểm tra đếm tham chiếu, giai đoạn này sử dụng thông tin trong giai đoạn 2 và 3 để kiểm tra các file không tham chiếu và đếm liên kết không đúng trên các file, directory, hoặc file đặc biệt. Thông báo như sau:

UNREF FILE I=inode number OWNER=O MODE=M SIZE=S MTIME=T (RECONNECT?)

LINK COUNT FILE I=inode number OWNER=O MODE=M SIZE=S MTIME=T COUNT=X

(ADJUST?)

LINK COUNT DIR I=inode number OWNER=O MODE=M SIZE=S MTIME=T COUNT=X

(ADJUST?)

BAD/DUP FILE I=inode number OWNER=O MODE=M SIZE=S MTIME=T (CLEAR)

Phase 5: Kiểm tra Cylinder Groups, giai đoạn này kiểm tra các block tự do block và các inode chưa sử dụng. Nó sẽ tự động sửa đổi lại danh sách các inode tự do cho đúng nếu cần thiết, tuy nhiên có các yêu cầu đòi người quản trị phải trả lời.

What Do I Do After fsck Finishes?

Lệnh fsck:

6. Backup and restore

Xem thêm phần end user

Bản thân trong hệ điều hành UNIX hỗ trợ nhiều công cụ hỗ trợ việc lưu trữ và hồi phục trên các thiết bị vậ lý khác khau. Nó hỗ trợ các mức sao lưu và hồi phục giúp cho việc tối ưu trong công tác sao lưu và đảm bảo việc quản trị dễ dàng.

Ngoài các lệnh tar, cpio, dd, tuỳ theo hệ điều hành hỗ trợ các công cụ khác phục vụ cho công tác sao lưu và hồi phục dữ liệu.

IX.Printer administration

Unix hỗ trợ máy in cắm trực tiếp trên cổng parallel, serial cũng như hỗ trợ cho máy in mạng. Việc cấu hình máy in tương đối đơn giản nhờ các tiện ích hỗ trợ bởi các hệ điều hành. Tuy nhiên trên một số loại version cũ việc cấu hình lại phải tự làm bằng tay qua lệnh (mknod) tương đối phức tạp.

Thông thường các cổng parallel thường tương ứng với các file /dev/lp0, /dev/lp1, hoặc /dev/lp2 phụ thuộc vào số cổng mà máy có.

Các tiện ích liên quan đến printer bao gồm

/etc/printcap Chứa cấu hình máy in .

/usr/lib/lpd Kiểm soát và cung cấp các dịch vụ in

/usr/ucb/lpr Thực hiện việc chuyển các print job vào printer queue.

/usr/ucb/lpq Chương trình kiểm tra hàng đợi in.

/usr/ucb/lprm Thực hiện xoá các print job từ print queue.

/etc/lpc Quản trị print job và printer queue.

Chương trình lpd thực hiện việc kiểm soát và cung cấp các dịch dịch vụ in. Các thông tin xác định cấu hình gồm tên máy in, cổng in, loại máy in ... Các thông tin này có thể được thay đổi bởi người quản trị.

lpd [-l] [port]

Với tham số -l ghi lại các thông tin tới log file mỗi lần kiểm soát các request.

Khi lpd nhận được print request (gọi là print job) các trang in được đưa đến một vùng gọi là spool(thông thường là thư mục /usr/spool/lp). Và màn hình in được giải phóng để người sử dụng có thể thực hiện các công việc khác. Sau đó lpd sẽ thực hiện việc gửi dữ liệu từ spool ra máy in tương ứng.

Việc quản lý printer thông qua tiện ích là lpc. Nó cho phép người quản trị thực hiện một số chức năng như hiển thị thông tin trạng thái máy in, cho phép hoặc không cho phép in, cho phép hoặc không cho phép các hàng chờ in, loại bỏ các print request, thay đổi mức độ ưu tiên trên hàng chờ in. Ngoài ra còn tiến ích lpq và lprm cho phép thực hiện quản lý các hàng đợi in.

X. Network administration

1. UUCP (Unix to Unix copy)

UUCP là một nhóm các lệnh và các dịch vụ cung cấp dịch vụ mạng đơn giản cho phép người sử dụng truyền dữ liệu trên UNIX từ một máy này sang máy khác qua serial port (modem) thông thường dùng để truyền file, e-mail, chạy các lệnh trên máy ở xa. Hiện nay có nhiều version của UUCP trên các hệ điều hành các hệ điều hành UNIX tuy nhiên chúng vẫn tuân thủ theo quy tắc cấu hình chung của UUCP.

Các thành phần của UUCP: Các lệnh trong UUCP chia làm hai lớp: lệnh của người sử dụng và các lệnh quản trị.

Các chương trình người sử dụng là được chứa trong /usr/bin

- uucico - Công cụ nối kết các máy tính xa cho phép truyền tập tin hay thực hiện lệnh.
- uucp - Sao chép một tập tin từ một máy đến máy khác, uucp tạo tập tin dữ liệu và tập tin làm việc, xếp hàng các công việc để truyền và gọi uucico daemon nối kết với máy từ xa.
- uustat - Hiển thị trạng thái của yêu cầu truyền (uucp,uuto,uux) nó cũng kiểm soát hàng chờ truyền.

Các lệnh quản trị

Đa số các lệnh quản trị là ở trong /usr/lib/uucp các tập tin dữ liệu và shell script là ở trong /etc/uucp.

- uucleanup - Xóa thư mục spool nó thường được thực hiên từ shell script được gọi là uudemon.cleanup mà được khởi động bởi cron.
- Uutry -Kiểm tra khả năng gọi và gỡ rối, nó kéo theo uucico dể thiết lập thông tin giữa máy tính cục bộ và máy tính từ xa.
- uucheck - Kiểm tra sự tồn tại của các thư mục chương trình và các tập tin support của uucp nó có thể cũng kiểm tra tập tin Permission để kiểm tra ngữ pháp.

Deamons Có ba deamon trong hệ thống uucp. Các deamon này diều khiển truyền tập tin và thực hiện lệnh. Chúng cũng có thể được chạy từ shell.

- uucico- Chọn thiết bị sử dụng để nối kết, thiết lập nối kết đến máy tính từ xa, thực hiện các yêu cầu login và kiểm tra chủ quyền, truyền các tập tin dữ liệu, thực thi , đưa kết quả vào log và báo cho người sử dụng bởi mail. Khi uucico gọi máy từ xa nó liên lạc với uucico của máy từ xa. uucico được gọi bởi uucp, uuto,uux và uusched, uutry.
- uuxqt - Thực hiện các yêu cầu thực thi từ xa, nó sẽ kiểm tra thư mục spool để tìm tập tin thực thi (X.file) đã được gửi từ máy từ xa. Khi tập tin X.file đã tìm thấy uuxqt mở nó và nhận một danh sách các tập tin dữ liệu được yêu cầu thực thi. Nó sẽ kiểm tra chủ quyền truy xuất của cáac tâp tin này. Kế đó nó đọc tập tin permission để kiểm tra chủ quyền thực hiện lệnh được yêu cầu. uuxqt được thực hiện bởi uudemon.hour shell script.
- uusched -Định thời biểu các hàng chò trong thư mục spool. Trước khi khởi động uucico, uusched sắp thứ tự ngẫu nhiên các máy tính từ xa sẽ được gọi. uusched được khởi động ở thời gian boot bởi /etc/rc nó sẽ được thực hiện bởi uudemon.hour.

Khởi động uucp : uucp sẽ khởi động với các lệnh trong shell script mà quét vòng các máy từ xa, định thời biểu truyền và xoá các log cũ.

$su uucp

$crontab < /usr/lib/uudemon.crontab

Các file cấu hình /usr/lib/uucp

Systems, Devices, Permissions, Dialers, Dialcodes

Devices Chứa các thông tin liên hệ vị trí, tốc độ của bộ gọi tự động(ACU), đường tự động và thiết bị mạng.

Format	Type	Line	Line2	Class	Dialer-Token Pairs
Ví dụ	ACU	cua0	-	1200	Hayes \D

Type ACU Sử dụng với modem

Direct Sử dụng với đường dây trực tiếp

Line Tên thiết bị ví dụ modem /dev/cua0

Class Tốc độ

Dialer-token pairs tương ứng với loại modem trong tập tin Dialers

Dialers Xác định cách trao đổi ban đầu trước khi truyền dữ liệu như.Ví dụ :

Hayes =,-, "" \dA\pTE1V1X1Q0S2=255S12=255S50=2\r\c OK\r \EATDT\T\r\c CONNECT

Ý nghĩa của các kí tự escape

\p ngưng (1/4 giây)

\d trể (2 giây)

\D Chỉ số điện thoại không biến đổi bởi tập tin Dialcodes

\T Chỉ số điện thoại biến đổi bởi tập tin Dialcodes

\K Thêm vào một BREAK

\E Hiện thi kiểm tra

\e Cấm hiển thi kiểm tra

\r Đầu dòng

\c Không xuống dòng

\n Xuống dòng

Systems Chứa các thông tin cần thiết bởi uucico để thiết lập nối kết với máy từ xa. Mỗi phần tử của tập tin thể hiện một máy tính có thể được gọi. Ví dụ :

Format	System-Name	Time	Type	Class	Phone	Login
Ví dụ	eagle	Any	ACU	1200	555678	in:nuucp ord:okasa

System-Name -Tên của máy từ xa.

Time - Thời gian máy từ xa có thể bị gọi.

Ví dụ: Wk1700-0800 mọi ngày trong tuần từ 17h-8h

Login - Chứa các thông tin login.

Dialcodes: Chứa các mã điện thoại viết gọn. Khuôn dạng như sau:

viết tắt	số
HN	014

Permissions Xác định chủ quyền login, truy xuất tập tin và thực hiện lệnh của máy từ xa. Mỗi phần tử của tập tin là một dòng có dạng tên=giá trị. Có thể có các chủ quyền sau :

LOGNAME= Xác định các máy có thể log vào.

MACHINE= Xác định các máy từ xa mà máy cục bộ có thể log.

REQUEST= yes/no cấm hay cho phép máy từ xa nhận các tập tin từ máy cục bộ.

SENDFILES=yes/no cấm hay cho phép máy từ xa gửi các tập tin đến máy cục bộ.

READ/WRITE= Xác định các thư mục được phép đọc hay ghi.

Ví dụ:

READ=/var/spool/uucppublic

WRITE=/var/spool/uucppublic

NOREAD/NOWRITE Xác định các thư mục cấm đọc hay ghi.

COMMANDS Xác định các lệnh mà máy từ xa có thể thực hiện.

2. TCP/IP and Neworks

a) TCP/IP

TCP/IP là giao thức mạng chuẩn được hỗ trợ trong UNIX .

Thông thường khi cài đặt hệ thống bao giờ cũng được yêu cầu xác định các thông tin liên quan đến hệ thống như host name, domain (nếu host nằm trong domain), địa chỉ IP. Tuy nhiên có thể cấu hình mạng sau khi cài đặt bằng các tiện ích.

Đặt tên host: Có thể dùng lệnh hostname để đặt tên của host (phải người có thẩm quyền). Có thể đặt tên dạng tên vùng nếu nó nằm trong domain.

Ví dụ: hostname jac.domain

Cấu hình loopback driver: Để thiết lập cấu hình TCP/IP mạng dùng lệnh ipconfig

ifconfig <interface> <IP_address> netmask <mask> broadcast <address>

- interface - Xác định kiểu network interface card. (Ví dụ 3Com 3C5x9 như elx0).
- IP_address - Địa chỉ IP gán cho network interface. Có thể sử dụng bằng host name cung cấp trong /etc/hosts với địa chỉ IP tương ứng.
- netmask mask - Là subnetwork mask. Có thể bỏ qua nếu dùng giá trị default.
- broadcast address - Là địa chỉ broadcast cho network. Nếu không đặt sẽ lấy giá trị ngầm định.

Ví dụ: Để thiết lập the loopback driver:

ifconfig lo 127.0.0.1

Muốn đưa địa chỉ loopback vào trong kernel routing tables dùng lệnh:

route add 127.0.0.1 hoặc route add localhost

Cấu hình Ethernet Interface:Muốn thực hiện cấu hình với Ethernet driver thì một phải yêu cầu đòi hỏi phải biết chính xác Ethernet driver và cấu hình dùng *ifconfig* để kernel biết về interface và sau đó thêm route tới các máy trên mạng nếu nó gắn với mạng.

Ví dụ: Ethernet device là /dev/elx0

ifconfig elx0 192.168.70.2 netmask 255.255.255.0

Để thêm thành phần trên vào kernel routing table để kernel biết về địa chỉ mạng local machine và gửi chính xác dữ liệu tới nó dùng lệnh.

route add -net 192.168.70.0

TCP/IP Setup Files

Khi cài đặt và cấu hình TCP/IPcó một số file yêu cầu lưu giữ cấu hình liên quan đến dịch vụ của UNIX host.

/etc/hosts - Là file có chứa ánh xạ host names tới IP address tương ứng.

Ví dụ:

```
# Internet host table
# IP address    hostname    aliases
127.0.0.1               localhost
201.190.2.1             jade
```

/etc/networks - Chứa các network name ánh xạ tới địa chỉ mạng IP tương ứng.

Ví dụ:

```
# Name        Network Number
testnetw            201.190.2
loopback-net        127
```

/etc/services - Chứa các dịch vụ ở tầng TCP/IP application như FTP, TELNET, RLOGIN ... ứng với nó là các port number tương ứng.

Ví dụ:

```
#    assigned numbers from rfc1060
```

```
#service        port/transport
tcpmux          1/tcp
echo            7/tcp
echo            7/udp
discard         9/tcp           sink null
discard         9/udp           sink null
systat          11/tcp          users
systat          11/udp          users
.
```

/etc/protocols - Chứa các số xác định IP protocol service user. Các thông tin này giúp IP việc xác định routing data tới user protocols.

/etc/ethers – Tạo bởi hệ thống trong quá trình cài đặt TCP/IP. File này ánh xạ địa chỉ Ethernet tới địa chỉ IP. Các thông tin này phải có khi muốn cung cấp các dịch vụ RARP hoặc BOOTPD

Ví dụ:

```
# ether_mac_addr.    hostname    comments
00:00:c0:9e:41:26    violin       #strings dep't
02:60:8c:15:ad:18    bass
```

/etc/netmasks - Chứa netmask tương ứng với địa chỉ network IP.

```
#Network subnet masks
#Unless your network is subnetted, do not bother to maintain this file
#Network       subnet mask
134.54.0.0       255.255.255.0
167.12.0.0       255.255.192.0
```

/etc/hosts.equiv - Xác lập quyền truy nhập của các máy và các user ở xa khi dùng các tiện ích ví dụ: rlogin, rcp, rsh ...

~/.rhosts Tạo trong thư mục home của user xác định quyền truy nhập của user xác định.

Các tiện ích và các dịch vụ chủ yếu cung cấp trên ICP/IP

Một số các daemon cung cấp các dịch vụ trên Unix được điều khiển bởi inetd.

- ftpd - Cung cấp các dịch vụ truyền file trên máy Unix qua TCP port 20.
- telnetd - Cung cấp các dịch vụ cho phép cung cấp các dịch vụ cho các máy kết nối dạng terminal.
- rshd - Cho phép thực hiện các lệnh shell từ các máy xa.
- logind- Cho phép thực hiện login từ xa.
- execd - Chon phép thực hiện các lệnh từ các máy ở xa.
- comsat- Kiểm soát các mail đến cung cấp các thông tin cho các processes mà yêu cầu nó thực hiện.
- talkd - Cho phép người sử hội thoại với nhau qua dùng keyboard và screen của terminal ở bất kỳ đâu trên mạng.
- uucpd- Cung cấp dịch vụ truyền dữ liệu sử dụng UUCP trên mạng.
- tftpd-Cung cấp dịch vụ tftp trên mạng. Hỗ trợ khả năng remote boot và truyền file.
- fingerd- Kiểm soát user trên mạng.
- rquotad- Kiểm soát việc sử dụng đĩa của người sử dụng.
- walld-Cho phép thực hiện việc gửi các message tới màn hình của các các user trên hệ thống
- rstatd This daemon returns performance statistics about this system
- cmsd-Quản lý calendar trên server.
- routed - Thực hiện việc kiểm soát RIP.
- gated- Cho phép thực hiện nhiều thông tin về route bao gồm RIP, OSPF (Open Shortest Path First), EGP (Exterior Gateway Protocol) ...
- nfsd- Kiểm soát và cung cấp các dịch vụ NFS trên mạng.

- biod-(Block Input/Output Daemon). Chạy trên NFS client thực hiện việc kiểm soát quá trình đọc và ghi dữ liệu trên NFS server.
- mountd - Kiểm soát, cung cấp các yêu cầu mount từ NFS client.
- lockd - Điều khiển việc lock file trên mạng.
- rpcbind- Cung cấp các dịch vụ RPC (Remote Procedure Calls).
- sendmail- Thực hiện kiểm soát, trao đổi mail giữa các host (qua SMTP).
- named daemon và các database file cho phép UNIX cung cấp các dịch vụ DNS server trên UDP port 53.
- Ngoài ra với các thành phần mở rộng sau này các hệ điều hành còn cung các daemon hỗ trợ cung cấp các dịch vụ mở rộng khác như các dịch vụ X-terminal, HTTP ...

Các tiện ích (xem trên End user)

b) PPP

PPP là giao thức được sử dụng tương đối phổ biến trong thực hiện kết nối truyền thông giữa các máy hoặc giữa các mạng qua các serial line. Tuy nhiên cấu hình PPP trên UNIX thì tướng đối phức tạp. UNIX chia các PPP ra làm hai phần, một là mức High-Level Data Link Control (HLDC) protocol, thực hiện việc gửi các PPP datagram giữa hai máy, và PPP daemon gọi là pppd thực hiện kiểm soát các protocol trên hệ thống HLDC và thiết lập các biến truyền thông.

Khi thực hiện kết nối thông qua PPP thì người sử dụng không bị yêu cầu nhập các dấu nhắc kết shell hoặc login vì phần này sẽ do PPP đkiểm soát. Để thiết lập kết nối PPP nhất thiết loopback driver phải được thiết lập.

Trong UNIX để tăng cường tính an toàn cho hệ thống khi thực hiện kết nối PPP giữa các hệ UNIX cần phải có user đặc biệt để thực hiện login vào hệ thống khi thực hiện kết nối. User này không sử dụng các shell bình thường mà sử dụng một chương trình đặc biệt để khởi động và cấu hình PPP. Ngoài ra PPP hỗ trợ PPP Authentication để xác định các kết nối trên hệ thống.

Ví dụ: trong Linux là pppscript (trong hệ Unix system V dùng file aspppls ...)

ppp:*:201:51:PPP account:/tmp:/etc/ppp/pppscript

Nội dung file pppscrip này như sau:

#!/bin/sh

mesg n

stty -echo

exec pppd -detach silent modem crtscts

Trước khi PPP hoạt động nó yêu cầu phải thiết lập cuộc kết nối trước đó tới máy ở xa trước khi nó thực hiện kiểm soát kết nối. Điều này có thể thực hiện qua chương trình thực hiện kết nối (chat program). Thông thường sử dụng qua UUCP trong file này chú ý đến nội dung trong file Systems.

Để thực hiện việc kiểm soát kết nối sử dung PPP thì nhất thiết chương trình pppd daemon phải được gọi chạy.

Ví dụ: Nếu máy sử dụng COM1 để thực hiện cho kết nối PPP và kết nối ở tốc độ 38,400 baud có thể sử dụng lệnh như sau:

pppd /dev/cua1 38400 crtscts defaultroute

Trong hệ điều hành UNIX system V thường sử dụng một script tên là asppp để khởi động pppd dựa trên các file cấu hình đã đặt.

c) DNS

Trước đây trong UNIX để giải quyết vấn đề ánh xạ tương ứng tên tới địa TCP/IP sử dụng /etc/hosts. Khi thực hiện các kết nối TCP/IP sẽ thực hiện kiểm tra trong /etc/hosts kiểm tra tra tên và đọc địa chỉ của máy kết nối. Nếu tên không có trong file này thì máy sẽ không thể thực hiện kết nối theo tên. Với phương pháp này mà khi cần kết nối với nhiều máy trên hệ thống thì việc quản lý nó sẽ trở lên rất phức tạp. DNS ra đời và phát triển để giải quyết các vấn đề trên đồng thời nó còn tạo ra các khả năng quản trị tên phân cấp và cung cấp các dịch vụ tên vùng. Trên DNS bao giờ cùng gồm có hai phía là người cung cấp các dịch vụ (DNS server) và người sử

dụng các dịch vụ (client).

Thông thường mỗi một zone thường có hai master name để duy trì hoạt động của cả zone là primary master server và secondary master server (backup server).

DNS Client

DNS client thực hiện việc kết nối thông qua các DNS query để thực hiện lấy ác ánh xạ tương ứng tới địa chỉ đích cần kết nối. DNS query có thể có nhiều loại. Hầu hết trong số này là các query hostname đến IP address. Các query giải quyết vấn đền ánh xạ IP address tới hostname là PTR hay pointer queries.

Ngoại trừ việc cấu hình để liên kết với name server. Client thông thường kiểm tra trong */etc/hosts* file để lấy địa chỉ tương ứng với name của máy chỉ định.

Để cấu hình DNS client (resolver) tạo file */etc/resolv.conf.* Sử dụng file này để xác định các domain name mà nó thuộc vào bao gồm địa chỉ IP của primary, secondary, hoặc cache name server. Cấu trúc file */etc/resolv.conf* gồm các trường

keyword value

Ví dụ:

```
domain      ham.com          #Domain
nameserver  198.53.18.1      #Primary name server
nameserver  198.53.18.3      #Secondary name server
```

DNS serve*r*

Để cấu hình name server bao gồm tạo các database và startup file. Số lượng các file này phụ thộc vào quy mô của tổ chức, cấu trúc liên mạng, và số domain mà nó được uỷ nhiệm quản trị.

Để khởi động DNS server UNIX gọi chạy một in.named daemon. Chương trình này sẽ đọc các file cấu hình trên database file và thực hiện các dịch vụ theo cấu hình đã định.

Các DNS database và startup file:

named.hosts File này xác định domain mà name server là người cung cấp và duy trì các ánh xạ từ tên tới địa chỉ IP.

named.rev Chứa thông tin về ánh xạ ngược địa chỉ IP tới tên.

named.local Chứa thông tin để giải quyết địa chỉ loopback (127.0.0.1) tới *localhost*.

named.ca File có chứa tên và địa chỉ của các root domain server.

named.boot File đầu tiên mà *named* (the DNS daemon) lúc khởi động và sử dụng nó để xác định database filename, và vị trí của chúng trong file system trên hệ thống cũng như là máy ở xa.

DNS Resource Records (RR)

Chuẩn DNS RFC 1033 định nghĩa nhiều loại resource record (RR). Mỗi loại tương ứng với một khía cạnh của cơ sở dữ liệu chung. Ví dụ kiểu record dùng để ánh xạ tên tới địa chỉ IP tương ứng (A), hoặc ánh xạ địa chỉ tới tên (PTR) hoặc sử dụng để duy trì ánh xạ giữa các domain name server (NS)...

Thông thường cấu trúc một RFC như sau:

[name] [ttl] class type data

- name Tên của đối tượng resource sẽ được mô tả bởi resource record. *name* có thể là tên host hoặc tên của domain.
- Ttl Là giá trị Time-to-live.
- Class Xác định class của DNS record.
- Type Xác định kiểu RR record. Thông thường người ta sử dụng các kiểu SOA, NS, A, MX và PTR.
- Data Dữ liệu thực sự của đối tượng *name* chỉ định.

Cấu hình primary name server

Để cấu hình named chạy và cung cấp dịch vụ như primary name server cần phải tạo và thay đổi các file dữ liệu sau:

- named.hosts
- named.rev
- named.local
- named.ca
- named.boot

named.hosts Là file duy trì ánh xạ hostname tới địa chỉ IP cho tất cả các trong zone. Muốn thực hiện truy nhập theo DNS thì named.hosts phải có chứa tên ánh xạ tới địa chỉ IP của tất cả các host trên mạng. Thêm vào đó nó còn chứa các thông tin khác như mail exchangers (hoặc mail routers), hoặc loại CPU ...

Ví dụ: file named.hosts mà jade là primary server trong domain harmonics.com:

```
;
; Section 1: The SOA record
;
harmonics.com.   IN   SOA   jade.harmonics.com.  root.jade.harmonics.com. (
            2    ; Serial
            14400      ; Refresh (4 hours)
            3600       ; Retry (1hr)
            604800     ; Expire ( 4 weeks )
            86400 )    ; minimum TTL (time-to-live)
;
; Section 2: The following are the name server for the harmonics domain. Notice how the second
; entry does not specify the domain name for which cello is being the name server. This implies that
; domain name is same as one specified in previous record.
;
harmonics.com.      IN   NS   jade.harmonics.com.
                    IN   NS   cello.harmonics.com.
;
```

```
; Section 3: The following are the mail exchangers of the domain harmonics.com
;
harmonics.com.      IN   MX   1  flute.harmonics.com.
                    IN   MX   2  bass.harmonics.com.
;
; Section 4: The following records map hosts' canonical names to their corresponding
; IP addresses
;
localhost.harmonics.com.    IN A    127.0.0.1
;
tenor.harmonics.com.        IN A    100.0.0.3
soprano.harmonics.com.      IN A    100.0.0.4
flute.harmonics.com.        IN A    100.0.0.5
xrouter                     IN A    100.0.0.10
cello.harmonics.com.        IN A    198.53.237.2
violin.harmonics.com.       IN A    198.53.237.3
bass.harmonics.com.         IN A    198.53.237.4
;
; Section 5: Multihomed hosts
;
jade.harmonics.com.         IN A    198.53.237.1
                            IN A    100.0.0.2
```

- Section 1 Có chứa SOA record là jade.harmonics.com là người cung cấp DNS server cho domain harmonics.com. File named.hosts có thể chỉ có chứa SOA record, và nó phải là record đầu tiên trong file.
- Section 2 Gồm các NS record là các host jade và cello như là tên servers của domain harmonics.com. Chú ý rằng các record này không được chỉ định hai server đều là primary mà phải có secondary server.

- Section 3 Gồm hai MX records xác định mail exchangers chính và phụ cho domain harmonics.com.
- Section 4 Gồm tất cả các A records mà ánh xạ host name tới địa chỉ IP addresses. Khi mà client thực hiện query thì name server sẽ trả lại địa chỉ IP cho host, *named* quét các A records trong named.hosts phù hợp với tên mà client yêu cầu và trả lại địa chỉ IP tương ứng với host name.
- Section 5 Tương ứng với jade là multihomed gồm hai A record. Khi mà *named* được query cho địa chỉ IP của jade, hoặc bất kỳ multihomed, *named* đơn giản là trả lại tất cả các địa chỉ mà nó tìm thấy.

named.rev named sử dụng named.rev để giải quyết các kiểu PTR query. Nó trả lại host name tương ứng với query lấy địa chỉ IP tương ứng với host trong named.rev file. Trong named.rev file bao gồm cả SOA record.

Ví dụ: file 100.rev

```
100.in-addr.arpa.    IN SOA    jade.harmonics.com.  root.jade.harmonics.com (
                1             ;serial
                14400         ; Refresh (4 hours)
                3600          ; retry   ( 1 hour )
                604800        ; expire  ( 1 week )
                86400 ) ; TTL = 1 day
;
; name servers
;
100.in-addr.arpa.    IN NS    jade.harmonics.com.
100.in-addr.arpa.    IN NS    cello.harmonics.com.
;
; Reverse address mappings
;
2.0.0.100.in-addr.arpa.        IN    PTR    jade.harmonics.com.
3.0.0.100.in-addr.arpa.        IN    PTR    tenor.harmonics.com.
```

```
4.0.0.100.in-addr.arpa      IN   PTR   soprano.harmonics.com.
5.0.0.100.in-addr.arpa      IN   PTR   flute.harmonics.com.
10.0.0.100.in-addr.arpa     IN   PTR   xrouter.harmonics.com.
```

Trong tất cả các file named.rev phần đầu tiên bao giờ cũng là SOA record tiếp sau đó là NS và cuối cùng là PTR records.

named.local Có chứa cấu hình của localhost ứng với địa chỉ IP 127.0.0.1.

Ví dụ:

```
0.0.127.in-addr.arpa. IN SOA jade.harmonics.com. root.jade.harmonics.com. (
            1    ; serial
            14400      ; refresh ( 4 hours )
            3600       ; retry ( 1 hour )
            604800     ; expire ( 1 week )
            86400 )    ; TTL = 1 day
; name servers
;
0.0.127.in-addr.arpa.   IN   NS    jade.harmonics.com.
0.0.127.in-addr.arpa.   IN   NS    cello.harmonics.com.
;
; reverse address PTR mapping
;
1.0.0.127.in-addr.arpa. IN   PTR    localhost
```

named.ca Để nâng cao hiệu quả của các dịch vụ DNS, để giảm các traffic trên mạng. DNS cho phép server lấy dữ liệu từ cache của nó để trả lời các query từ các client. Nó thực hiện điều này bằng cách lưu các trả lời ứng với các query trong bộ nhớ tương ứng với các query mà client yêu cầu tới servers để sau này có thể lấy nó để dùng lại. DNS thực hiện điều này bằng cấu hình trong file named.ca tương ứng với các domain khác. Tất cả thông tin có chứa trong named.ca file được sử dụng để khởi động các cache buffer trên DNS server mmỗi khi named deamon(in.named) khởi động. Thông thường trong named.ca file bao gồm các thông tin về các root

server vì những thông tin này thường ổn định trong thời gian dài.

Ví dụ:

```
;
; Section 1: NS records for the root domain servers
;
.   99999999          IN    NS    A.ROOT-SERVERS.NET
99999999              IN    NS    B.ROOT-SERVERS.NET
99999999              IN    NS    C.ROOT-SERVERS.NET
99999999              IN    NS    D.ROOT-SERVERS.NET
99999999              IN    NS    E.ROOT-SERVERS.NET
99999999              IN    NS    F.ROOT-SERVERS.NET
99999999              IN    NS    G.ROOT-SERVERS.NET
99999999              IN    NS    H.ROOT-SERVERS.NET
99999999              IN    NS    I.ROOT-SERVERS.NET
;
; Section 2: Root servers A records
;
A.ROOT-SERVERS.NET    99999999    IN    A    198.41.0.4
B.ROOT-SERVERS.NET    99999999    IN    A    128.9.0.107
C.ROOT-SERVERS.NET    99999999    IN    A    192.33.4.12
D.ROOT-SERVERS.NET    99999999    IN    A    128.8.10.90
E.ROOT-SERVERS.NET    99999999    IN    A    192.203.230.10
F.ROOT-SERVERS.NET    99999999    IN    A    192.5.5.241
G.ROOT-SERVERS.NET    99999999    IN    A    192.112.36.4
H.ROOT-SERVERS.NET    99999999    IN    A    128.63.2.53
I.ROOT-SERVERS.NET    99999999    IN    A    192.36.148.17
```

named.boot: Tại thời điểm khởi động DNS server lấy các thông tin tham chiếu trong named.boot file để lấy các thông tin xác định cấu hình dịch vụ DNS là primary, secondary, hoặc cache server. Đồng thời xác địch các DNS database file bao gồm named.hosts, named.rev, named.ca và named.local.

Ví dụ: Về file named.boot nó sử dụng để cấu hình jade như primary name server:

```
directory       /etc/named
primary         harmonics.com               named.hosts
primary         100.in-addr.arpa            100.rev
primary         237.53.198.in-addr.arpa     198.53.237.rev
primary         0.0.127.in-addr.arpa        127.localhost
cache           .       named.ca
```

Mỗi dòng trong file là dòng xác định cấu hình của DNS server mà DNS deamon(in.named) đọc mỗi khi nó khởi động.

- Từ khoá *directory* ở dòng đầu tức DNS deamon sử dụng default location mà tất cả các DNS database file lưu trong nó /etc/named.
- Dòng 2, 3, 4 với từ khoá primary nói rằng đó là các DNS server là cung cấp các dịch vụ DNS ứng với domain xác định.
- Dong cuối cấu hình server thực hiện cache server cho các root domain. Các thông tin chi tiết trong named.ca.

Cấu hình secondary name server

Secondary name server là server mà nhận các dữ liệu tương ứng với zone từ primary server. Secondary server không có các named.hosts và named.rev file riêng của nó. Thay vào đó khi khởi động secondary server yêu cầu primary server truyền các bản copy của các file này. Sau đó secondary server khởi động với nội nội dung cấu hình dựa trên file nó nhận được.

Chỉ có một số thay đổi trong file named.boot của secondary DNS server so với primary server ví dụ:

```
directory       /usr/lib/named
secondary       harmonics.com               100.0.0.2
secondary       100.in-addr.arpa            100.0.0.2
secondary       237.53.198.in-addr.arpa     100.0.0.2
```

```
primary     0.0.127.in-addr.arpa     named.local
cache       .       named.ca
```

Trong named.boot bao gồm hầu hết là các dòng secondary thay cho primary. Dòng thứ hai chỉ ra rằng cấu hình như một secondary server với domain harmonics.com, và thực hiện copy tất cả các dữ liệu liên quan từ server với địa chỉ IP 100.0.0.2. Dòng 3, 4 cấu hình named như là secondary server cho reverse domains 100.in-addr.arpa, và 237.53.198.in-addr.arpa.

Dòng 5 cà 6 chứa các thông tin liên quan đến local filenames. Cả hai file named.local và named.ca rất ít khi thay đổi nội dung.

Secondary server được khởi động gần giống như primary server. Khi host chạy ở mức 2 tại thời điểm boot, các startup scripts kiểm tra sự tồn tại của named.boot file. Nếu nó tồn tại thì tuỳ theo cấu hình mà nó thực hiện các thao tác tiếp theo.

Trong thực tế sẽ xảy ra trường hợp khi secondary server khởi động mà primary server không sẵn sàng. DNS cho phép cấu hình các địa chỉ IP thay thế trong lệnh secondary từ các server có thể có chứa các zonal data, và cấu hình server duy trì các bản copy của file.

Ví dụ:

```
secondary harmonics.com 100.0.0.2 100.0.0.4
```

Nó có thể cải tiến bằng cách cho phép secondary name server duy trì các bản copy của các file nhận được. Bằng cách này secondary sẽ luôn có các bản copy của zonal data. Tuy nhiên dữ liệu này sẽ được kiểm tra quá hạn được ghi trong SOA record. Nếu primary server được tìm thấy trong zone thì secondary server sẽ yêu lấy lại các thông tin và bỏ các dịch vụ trước đó.

Để cho phép named sử sử dụng copy trong cột cuối cùng của từ khoá secondary chỉ tên file mà dữ liệu sẽ copy.

Ví dụ:

```
directory     /usr/lib/named
secondary     harmonics.com          100.0.0.2    named.hosts
```

```
secondary    100.in-addr.arpa            100.0.0.2    100.rev
secondary    237.53.198.in-addr.arpa     100.0.0.2    198.53.237.rev
primary      0.0.127.in-addr.arpa        named.local
```

Dòng directory /usr/lib/named chỉ ra rằng sử dụng datafile trong thư mục /usr/lib/named.

Cấu hình cache only server

Cache-only server không dựa vào các database file của nó. Cache-only server chỉ lưu các thông tin tương ứng với các query mà được sử dụng cho sau này nếu có thể.

Để cấu hình Cache-only server. Thêm vào named.boot dòng lệnh sau:

```
;
; Cache-only server for the harmonics.com domain
;
primary    0.0.127.in-addr.arpa    /usr/lib/named/named.local
cache      .          /usr/lib/named/named.ca
;
```

Lệnh nslookup: Cho phép chạy chế độ interactive để thực hiện kiểm tra hoạt động của dịch vụ DNS.

- Sử dụng nslookup để query local server
- Sử dụng nslookup để query remote server
- Sử dụng nslookup để download DNS database

Sử dụng nslookup để Query Local Server

Ta có thể sử dụng nslookup để kiểm tra các server mới, các thay đổi về cấu hình trên server và để xác định các sự cố đối với dịch vụ DNS.

Để thực hiện được điều này phải login vào người quản trị mạng (root) và sử dụng lệnh nslookup. Ngầm định là nslookup trả các tướng ứng với các name query (ánh xạ name-to-address).

Ví dụ:

```
# nslookup
Default Server:  jade.harmonics.com
Address:  100.0.0.2
> cello
Server:  jade.harmonics.com
Address:  100.0.0.2
Name:    cello.harmonics.com
Address: 198.53.237.2
```

Sử dụng nslookup để Query Remote Server

nslookup có thể sử dụng để thực hiện các query tới các remote server trên mạng. Remote servers có thể trên cùng mạng hoặc có thể ở đâu đó trên Internet. Điều này cho phép kiểm tra các sự cố về dịch vụ DNS trên bất kỳ các DNS server nào. Ví dụ: dùng nslookup qua host jade (primary server) để query cello (secondary server) cho địa chỉ của soprano

```
# nslookup
Default Server:  jade.harmonics.com
Address: 198.53.8.1
> soprano cello.harmonics.com
Server:  cello.harmonics.com
Address:  198.53.237.2
Name:    soprano.harmonics.com
Address:  100.0.0.4
```

Sử dụng nslookup để Download DNS Database

nslookup có thể thực hiện để truyền dữ liệu DNS trực tiếp ra các đầu ra chuẩn như màn hình hoặc như file.

Ví dụ:

```
# nslookup
Default Server:  jade
Address:  0.0.0.0
```

```
> ls harmonics.com
[jade]
harmonics.com.        server = jade.harmonics.com
jade                  198.53.8.1
harmonics.com.        server = cello.harmonics.com
cello                 198.53.237.2
tenor                 100.0.0.3
soprano               100.0.0.4
localhost             127.0.0.1
harmonics             server = jade.harmonics.com
jade                  198.53.8.1
soprano                 100.0.0.4
xrouter               10.0.0.10
cello                 198.53.237.2
> exit
#
```

d) NIS (Network Information Service)

Với các mạng lớn và kết nối trên nhiều máy thì mỗi máy phải tạo các account riêng nếu như muốn truy nhập. Với việc tạo nhiều các account trên một hệ thống như vậy sẽ dẫn đến việc gặp rất nhiều khó khăn trong công tác quản trị và sử dụng.

NIS là dịch vụ mạng (phát triển Yellow Pages protocol) cung cấp directory service, đáp ứng việc quản trị tập trung trên các mạng lớn. Nó là cho phép thực hiện các truy nhập phân tán trên hệ thống mạng sử dụng quyền truy nhập thông qua một trung tâm là NIS master (hay ypmaster) để xác nhận quyền truy nhập. Trên một mạng lớn thông thường người ta có thể thiết lập thêm các slaves (hay ypslaves) để thực hiện tạo thành các máy dự phòng (lưu toàn bộ thông tin về quyền truy nhập trên toàn hệ thống từ master). Trong trường hợp master server gặp sự cố slave có thể thực hiện chức năng thay thế.

NIS lưu giữ tất cả các thông tin trong “maps" mỗi map ứng với một vùng trên

network điều này cho phép một vài groups sử dụng chung một NIS master nhưng lại có quyền truy nhập khác nhau. NIS map không tương ứng với DNS domain mà nó cho phép nhiều mềm dẻo hơn trong việc cấu hình. Maps bao gồm các record ghi dưới dạng ASCII.

Thành phần của NIS bao gồm:

- domain
- maps
- Daemon

  ypser Server process

  ypbind Liên kết các process

  ypxfrd Daemon truyền dữ liệu liên kết tốc độ cao.

  roc.ypupdated Thực hiện cập nhập dữ liệu khi dữ liệu map thay đổi.

  in.named Thực hiện dịch vụ DNS (optional)

- Utilities

  ypcat Liệt kê dữ liệu trong map

  ypwich Hiện tên NIS server và map server

  ypwatch Tạo key ứng với giá trị trong map

  ypinit Cài đặt và xây dựng NIS database

  yppoll Lấy các số thứ tự từ server

- Các tiện ích khác

  yppush Chuyển dữ liệu từ master tới slave.

  ypset Đặt liên kết thực sự tới server.

  ypxfr Truyền dữ liệu từ master tới slave với tốc độ lớn.

  makedbm Tạo dbm file cho một NIS map

Thông thường NIS server có thể hỗ trợ một vài map files thông thường là các file

chuẩn trên UNIX như group, hosts, networks, passwd, protocols, rpc, services sau đó nó được chuyển thành các file tương ứng ví dụ:

/etc/group group.byname, group.bygid

/etc/hosts hosts.byname, hosts.byaddr

/etc/passwd passwd.byname, passwd.byuid

Tất cả các file này được lưu dưới khuôn DBM gọi làn gdbm. Có thể sử dụng tiện ích ypmakedbm để tạo database file cho NIS.

3. NFS (Network File System)

Một hệ thống sử dụng trên cơ sở mạng UNIX với khả năng xử lý phân tán và client/server. Trên mạng gồm có nhiều workstation kết nối với hệ thống máy chủ và kết nối giữa các workstation với nhau. Nhiều ứng dụng lớn có thể được lưu trên các server. Có một số tiện ích cho phép truy nhập tới các file ở xa nhưng chỉ cho phép thực hiện các truyền file hoặc dạng terminal. Để tích hợp hệ thống trên mạng cho phép truy nhập tài nguyên ở xa UNIX đưa ra dịch vụ NFS.

NFS cho phép thực hiện việc đọc và ghi các file trên NFS servers. Việc truy nhập từ các client tới NFS servers có thể thực hiện thông qua ánh xạ (mount) nó đến một thư mục trên máy cục bộ (gọi là điểm "mount"). Sau đó việc truy xuất đến thư mục của máy từ xa được thực hiện bằng cách truy xuất trên máy cục bộ qua điểm "mount".

Ví dụ: mount -F nfs –o ro Remote:/export/app /temp/data

Khi sử dụng lệnh này client kiểm tra remote machine xem có quyền truy nhập thư mục hay không. Nếu có quyền thì nó gửi một thông tin điều khiển sử dụng để định hướng tất cả các yêu cầu truy nhập từ client. Trương trình thực hiện việc kiểm soát và cung cấp các dịch vụ là nfsd. Trong hệ điều hành UNIX một máy có thể vừa là NFS server vừa là client.

Việc sử dụng các dịch vụ NFS giúp cho việc quản trị tập trung trên hệ thống lớn được thuận lợi hơn (nhất là trong các công tác sao lưu và hồi phục, quản trị các ứng

dụng ...)

NFS server

Hoạt động của NFS trên server được điều khiển bởi các deamon là rpc.mountd, nfsd và file cấu hình thiết lập danh sách các thư mục mà client có thể thực hiện mount trong file /etc/exports. File /etc/exports được đọc mỗi lần mountd daemon nhận được yêu cầu mount thư mục.

Ví dụ:

/etc/exports for merlin

/usr/database/data chatton(rw) big_roy (rw) wizard (rw)

/usr/book chatton(rw) wizard (ro)

/usr/bin/bigapp big_roy(rw) wizard (ro)

/usr/ftp (ro)

Muốn thay đổi tập tin /etc/exports phải có quyền superuser trên máy server. Sau khi thay đổi để cập nhật lại thông tin cho server sử dụng *lệnh share*:

NFS client

Client truy xuất các tập tin trên server bằng cách mount các thư mục mà server xuất. Khi client mount một thư mục trên server đó là một quá trìng sử dụng chuỗi lệnh gọi thủ tục từ xa (Remote Procedure Call) cho phép client truy xuất đến thư mục trên server. Thông tin này được deamon rpc.mountd xử lý và xác định client được phép hay không được phép truy xuất đến các thư mục của server. Tiến trình client tìm server xuất ra các thông tin mà client cần và sau đó thiết lập đường truyền giữa client và server gọi là binding. Sự ràng buộc NFS xảy ra trong suốt quá trình client mount một thư mục từ xa. Việc mount thư mục từ xa có thể được thực hiện khi khởi động hoặng thông qua lệnh mount hay cơ chế automounter. Tập tin /etc/fstab liệt kê các thư mục mà client mount trong khi khởi động. Với automounter client có thể tự động mount các thư mục trong quá trình làm việc mà không cần phải gọi lệnh mount.

4. Mail

Trong hệ điều hành Unix server thường hỗ trợ hai thành phần gồm Mail User Agent (MUA-mail, mailx elm) và Mail Transport Agent (MTA -sendmail). Với các thành phần này người gửi các message có thể thực hiện việc soạn, gửi các mesage tới các user trên các máy và đọc các message.

Với MTA cho phép thực hiện việc xác định đường sẽ gửi message dựa theo địa chỉ của người nhận và chuyển nó vào mailbox của người nhận. UNIX hỗ trợ việc truyền mail qua mạng TCP/IP hoặc qua dial-up sử dụng UUCP và sử dụng SMTP trong định dạng mail (Simple Mail Transfer Protocol).

Chương trình sendmail có thể được cấu hình làm mail router, final delivery agents, SMTP client, SMTP-server tuỳ theo cấu hình của người quản trị.

sendmail như Mail Router

sendmail có thể thực hiện chức năng như là mail router tức là nó có thể lấy letter, xem xét địa chỉ người nhận và quyết định cách tốt nhất để gửi nó.

Đầu tiên sendmail xác định một số thông tin nó cần như thời gian và tên của máy chủ mà nó đang chạy. Nhưng việc cấu hình nó như thế nào lại phụ thộc vào cấu hình của người quản trị trong file sendmail.cf. Các thông tin trong sendmail.cf sẽ xác định phương pháp điều khiển mail và đường đi của mail sẽ được gửi. .

sendmail như MTA-Client (Sender) và Server (Receiver) SMTP

sendmail có thể thực hiện chức năng MTA với SMTP protocol. Bởi vì SMTP là giao thức kết nối định hướng luôn có các client và server. SMTP client gửi letter tới SMTP server qua SMTP port.

sendmail có thể là một SMTP client hoặc một SMTP server. Khi chạy nó với chức năng MUA nó sẽ trở thành SMTP client. Khi hệ thống khởi động và chạy nó ở dạng daemon mode nó có thể thể hiện như là một SMTP server nhận các mail đến.

Cấu hình sendmail

5. UNIX client

Trong hệ thống việc chọn các client phù hợp cũng là một biện pháp là tăng hiệu quả sử dụng hệ thống. Tuy nhiên người ta có thể chộn lẫn các loại client trên một mạng.

Gồm một số các loại client sau:

Diskless clients—Là client không có đĩa riêng của nó. Thực hiện việc khởi động thông qua mạng nên đòi hỏi trên mạng phải có bootp server và NFS server cung cấp các dịch vụ về khởi động mạng và đĩa cho client. Vì diskless client không có đĩa riêng và khởi động từ mạng cho nên nó đòi hỏi việc sử dụng một lượng tài nguyên lớn trên các server

Dataless clients—Là các client có phần đĩa riêng của nó nhưng vẫn sử dụng các dịch vụ, các tiện ích cung cấp trên mạng. Nó không thể tự boot được mà phải boot từ mạng. Tuy nhiên do có phần đĩa riêng cho nên việc đòi hỏi sử dụng đĩa trên mạng không lớn như diskless client.

Standalone system - Là hệ thống có phần đĩa riêng của nó, có thể tự khởi động không đòi hỏi các dịch vụ khởi động trên mạng. Nếu kết nối vào mạng nó có thể sử dụng các dịch vụ của mạng mà được cung cấp bởi các server.

Time sharing system. – Là hệ thống với standalone system và các terminal kết nối với nó thông qua trực tiếp các serial line hoặc qua modem. Các terminal không có disk, printer, và file và CPU riêng của nó. Mà nó sử dụng chia sẻ các tài nguyên này với server.

Ngoài các hệ thống chạy trên cơ sở hệ điều hành Unix. Các Unix server còn hỗ trợ các kết nối từ các client sử dụng các hệ điều hành khác kết nối với nó thông qua các dịch vụ TCP/IP, X-terminal, SMTP ...

Quản trị các client: Việc quản trị quyền truy nhập trên hệ thống (xem phần trên) bao gồm phần quản trị quyền người sử dụng truy nhập tên hệ thống, quản trị việc sử dụng tài nguyên như đĩa, máy in, DNS, ...

Tuy nhiên với các client yêu cầu các dịch vụ khởi động từ server thì đòi hỏi phải có một vùng dữ liệu được tổ chức theo cấu trúc cần thiết phục vụ cho hoạt động của các client này. Trên đó chứa các thông tin về khởi động hệ thống, về cấu hình mạng, về định ngiã môi trường ...Để thực hiện việc tổ chức này, một số hệ điều hành UNIX cung cấp các tiện ích cho phép tạo và xoá các phần thông tin này.