

Bài thực hành số 5

QUẢN LÝ VÀO/RA – TẬP TIN – ĐỔI HƯỚNG

1 Nội dung thực hành

- Tập tin: Các khái niệm cơ bản về tập tin của Unix, tổ chức của các tập tin trên đĩa, các thao tác với tập tin.
- Vào ra: Các luồng dữ liệu vào/ra chuẩn và các thao tác đổi hướng chúng. Chuyển dữ liệu giữa các tiến trình.

2 Tổ chức tập tin

2.1 Các kiểu tập tin

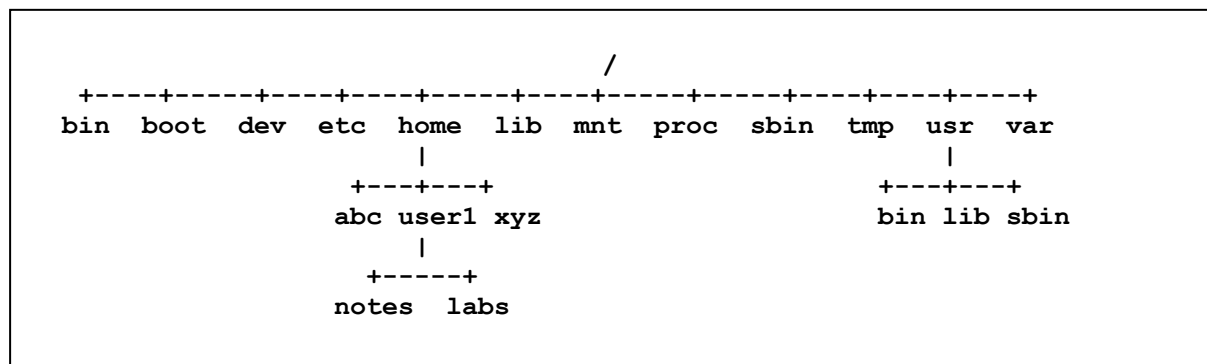
Unix có 3 kiểu tập tin:

- Tập tin bình thường: là một tập hợp thông tin (ASCII, text hoặc binary)
- Tập tin thư mục (tập tin danh mục- directory file): chứa danh sách các tên có thể truy nhập tới. Ví dụ như: các tập tin bình thường, các tập tin đặc biệt hoặc các thư mục con.
- Tập tin đặc biệt: là các tập tin liên quan đến các thiết bị ngoại vi cứng hoặc cơ chế truyền tin. Ví dụ như:
 - Bàn phím là một tập tin đầu vào (input file)
 - Màn hình là một tập tin đầu ra (output file)
 - Máy in là một tập tin đầu ra

2.2 Tổ chức tập tin

Các tập tin của Unix được tổ chức theo dạng cây. Thư mục gốc (root) của cây được biểu diễn bằng ký tự /. Các tập tin và các ổ đĩa khác chỉ là nhánh của Root.

Cấu trúc cây cơ sở của hệ Unix được bố trí như sau:



Trong đó:

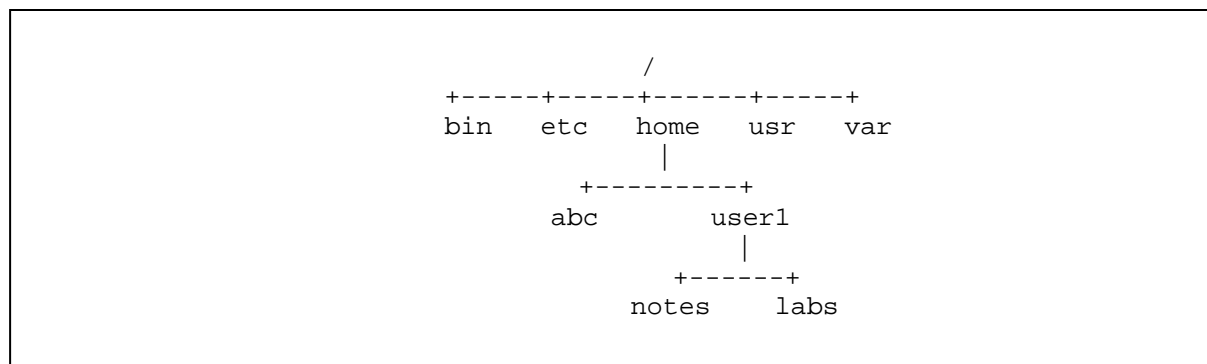
1. **/** – Là thư mục gốc (root) chứa tất cả các thư mục khác
2. **/bin** – chứa các tập tin nhị phân (chương trình) cần cho việc khởi động hệ thống hoặc làm việc trong chế độ người dùng đơn để duy trì hệ thống
3. **/boot** – chứa các kernel image và các tập tin cấu hình khi khởi động (boot configuration files)
4. **/dev** – chứa các tập tin đặc biệt được sử dụng để tương tác với các thiết bị phần cứng
5. **/etc** – chứa các tập tin cấu hình hệ thống
6. **/home** – chứa các thư mục home người dùng
7. **/lib** – chứa các module thư viện được dùng bởi các câu lệnh
8. **/mnt** – chứa các thư mục trỏ đến các thiết bị lưu trữ (mount point for other storage devices)
9. **/proc** – một hệ thống tập tin giả lập cho việc chuyển đổi dữ liệu giữa các tiến trình
10. **/sbin** – chứa các câu lệnh cần thiết cho việc quản trị hệ thống
11. **/tmp** – chứa các tập tin tạm thời (temporary files)
12. **/usr** – chứa các tập tin chương trình, các thư viện, các tài liệu, ... được dùng bởi những người dùng (normal users)
13. **/var** – chứa các tập tin dữ liệu thường xuyên thay đổi như: các tập tin log hệ thống, mail và các tập tin ghi nhận in ấn...

3 Đường dẫn (path)

3.1 Đường dẫn tuyệt đối

Một đường dẫn tuyệt đối sẽ chỉ ra vị trí chính xác của một tập tin hay thư mục được bắt đầu từ thư mục gốc (root).

✎ Ví dụ:



Đường dẫn tuyệt đối của thư mục **notes** là **/home/user1/notes**. Một đường dẫn tuyệt đối luôn bắt đầu với **/**

3.2 Đường dẫn tương đối

Một đường dẫn tương đối sẽ chỉ ra vị trí của một tập tin hay thư mục mà đường dẫn được xác định bắt đầu từ thư mục hiện hành

☞ Ví dụ :

Nếu như thư mục đang làm việc là */home* thì đường dẫn tương đối đến thư mục *notes* sẽ là : *user1/notes*

Một số ký hiệu đặc biệt có thể được sử dụng trong các đường dẫn tương đối:

- . Thư mục hiện hành
- .. Thư mục cha
- ~ thư mục home của người dùng hiện đang login

☞ Ví dụ:

Nếu thư mục đang làm việc là */home/user1*, thì ký hiệu *..* là một đường dẫn tương đối đến */home*

4 Di chuyển giữa các thư mục

Để di chuyển giữa các thư mục trong cây của Unix, ta dùng 2 lệnh sau đây:

- **cd** : chuyển đến thư mục cần đến (việc tương tác đến một vài thư mục có thể bị khóa tùy thuộc vào quyền người dùng)
- **pwd** : hiển thị tên thư mục đang làm việc

Tại thời điểm bắt đầu phiên làm việc, ta ở trong thư mục home. Muốn xem tên thư mục này, ta dùng lệnh *pwd*.

☞ Ví dụ:

user1 có thư mục home là */home/user1*

```
$pwd
/home/user1
```

Để di chuyển giữa các thư mục ta dùng lệnh *cd* với tên thư mục cần chuyển đến

☞ Ví dụ:

```
$cd /usr/bin
$pwd
/usr/bin
$cd ..
$pwd
/usr
```

Để về thư mục home của người dùng khi ta đang ở bất kỳ đâu, dùng câu lệnh:

☞ Ví dụ:

```
$cd
$pwd
/home/user1
```

5 Các thao tác cơ sở với thư mục

5.1 Xem nội dung thư mục

- Xem nội dung thư mục hiện đang làm việc:
\$ls
- Xem nội dung thư mục khác, ví dụ thư mục /bin:
\$ls /bin
- Xem thêm thông tin của các tập tin trong thư mục
\$ls -l
Hoặc:
\$ll

Không có sự hạn chế dùng ký tự nào khi đặt tên tập tin, tuy nhiên ta cần chú ý:

- + Không dùng các ký tự đặc biệt (trừ dấu chấm '.' hoặc dấu gạch dưới '_') vì phần lớn các ký tự đó được dùng trong cú pháp của lệnh shell
- + Tập tin có tên bắt đầu bằng dấu chấm '.' Là tập tin ẩn (hidden)
- + Phân biệt ký tự hoa và thường

Đặt tên tập tin bằng cách dùng các metacharacter (? và “)

- + Ký tự ‘ “ ’ thay thế 1 chuỗi ký tự
- + Ký tự ‘?’ thay thế 1 ký tự

✎ Ví dụ:

```
$ll /bin/c”  
$ll /bin/c?
```

5.2 Các trường cho một mục từ của thư mục:

Khi xem một thư mục trong Unix, chúng ta có thể đọc được những thông tin như:

- Quyền người dùng (Permissions) :
Người dùng hay người quản trị cần phải cấp quyền cho những người dùng khác để cho phép họ có thể tương tác đến các tập tin hay thư mục. Mỗi tập tin hay thư mục có các quyền đọc, viết, hoặc thực thi (read, write and execute permissions) còn tùy thuộc trên từng người dùng hay nhóm người dùng. Mặc định người dùng đã tạo ra tập tin hay thư mục (còn được gọi là owner) sẽ có tất cả các quyền trên nó.
- Liên kết (Link) :
Người dùng có thể tạo ra các liên kết (link) đến các tập tin hiện có. Hệ thống tập tin sẽ theo dõi các liên kết này
- Owner và nhóm (Owner and group) :
Unix là một hệ điều hành đa người dùng. Đối với mỗi tập tin hay thư mục, hệ thống tập tin sẽ theo dõi người dùng đã tạo ra nó (owner) và nhóm người dùng
- Kích thước (Size)
Được tính theo byte
- Thời gian (Date)

☞ Ví dụ: Một mục từ của thư mục:

\$ls -l test

-rwxrw-r-- 1 xyz staff 12 Mar 7 12:35 test

- Đây là một tập tin bình thường
- **rwx** : quyền của owner (xyz): read, write, execute
- **rw** : quyền của nhóm owner: read, write, not execute
- **r--** : quyền cho nhóm người dùng khác: read, not write, not execute
- **1** : Số liên kết đến tập tin
- **12** : Kích thước tập tin (bytes)
- **Mar 7 12:35** : Thời gian tập tin được sửa đổi sau cùng
- **test** : Tên của tập tin

5.3 Các câu lệnh khác có liên quan:

Các câu lệnh tập tin và thư mục, sinh viên cần tham khảo ở phần thực hành số 2

6 Bảo vệ các tập tin người dùng

6.1 Mô tả người dùng

6.1.1 Khái niệm:

Một người sử dụng được mô tả bằng các thông tin sau:

- Tên
- Mật khẩu (nếu có)
- Số nhận dạng (uid: user identify number)
- Số của nhóm (gid: group identify number)
- Chú thích (nếu có)
- Thư mục home của người dùng (home directory)
- Tên chương trình cho chạy lúc bắt đầu phiên làm việc (nếu có): bash shell script

Các thông tin trên được chứa trong tập tin `/etc/passwd`

☞ Ví dụ:

```
$cat /etc/passwd | more
```

6.1.2 Lệnh **useradd**:

Lệnh này ở trong thư mục `/etc`, nó cho phép:

- Hiển thị danh sách người dùng
- Thêm người dùng mới (chỉ người quản trị hệ thống có quyền)

6.2 Mô tả nhóm người dùng

6.2.1 Khái niệm:

Một nhóm người dùng là tập hợp của 1 số người dùng có thể dùng chung các tập tin của nhau

Một nhóm người dùng được mô tả bằng các thông tin sau:

- Tên của nhóm
- Mật khẩu (nếu có)
- Số của nhóm (gid: group identify number)
- Danh sách những người khách (nếu có)

Các thông tin trên được chứa trong tập tin `/etc/group`

✎ Ví dụ: `$cat /etc/group`

6.3 Lệnh `groupadd`

Lệnh này ở trong thư mục `/etc`, nó cho phép:

- Hiện thị danh sách các nhóm người dùng
- Thêm nhóm mới (chỉ người quản trị hệ thống có quyền)
- Xem tên các tập tin trong thư mục theo cột: `$ls`

Khi dùng lệnh `ls -l` ta có thể phân biệt các kiểu tập tin bằng cách xem ký tự đầu của dòng hiển thị, nếu là:

- d : tập tin thư mục
- : tập tin bình thường
- c hoặc b : tập tin đặc biệt

6.4 Các quyền người dùng trên tập tin/thư mục

Khi tập tin được tạo ra, các thông tin sau đồng thời được ghi lại :

- uid của người tạo tập tin
- gid của người tạo tập tin
- các quyền trên tập tin

Tập tin được bảo vệ bởi một tập hợp các bit định nghĩa quyền trên tập tin:

r w x	r w x	r w x
owner	group	other

Trong đó :

r	quyền đọc
w	quyền ghi
x	quyền thực thi

Đối với thư mục:

r	quyền đọc nội dung thư mục
w	quyền tạo và xóa tập tin trong thư mục
x	quyền qua lại (crossing) thư mục

Lưu ý: các quyền với thư mục chỉ có hiệu lực tại một mức nhất định, thư mục con có thể được bảo vệ trong khi thư mục cha thì không.

7 Vào/Ra chuẩn – Đối hướng

7.1 Các tập tin vào/ra chuẩn

Khi cho 1 lệnh chạy, Shell tự động mở 3 tập tin vào/ra chuẩn:

Vào chuẩn (stdin) fd=0

Ra chuẩn (stdout) fd=1

Lỗi chuẩn (stderr) fd=2

Ký hiệu fd là mô tả tập tin (file descriptor)

Thông thường đầu vào chuẩn là bàn phím, đầu ra chuẩn và lỗi chuẩn là màn hình.

7.2 Đối hướng đầu ra chuẩn

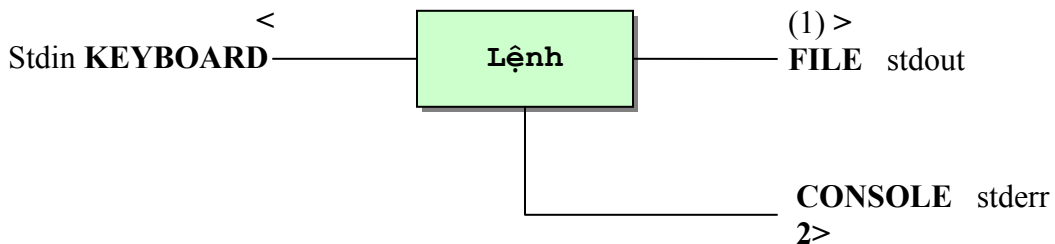
Ta có thể đổi hướng các số liệu vào 1 tập tin (thay vì ra màn hình) theo các cách sau:

\$lệnh > tên_tệp

Nếu tập tin chưa tồn tại, nó sẽ được tự động tạo ra. Nếu đã tồn tại, nội dung cũ sẽ bị xóa.

\$lệnh >> tên_tệp

Với cách này, số liệu sẽ được ghi thêm vào cuối tập tin.



Đối hướng đầu vào đầu ra chuẩn

☞ Ví dụ:

\$ls /bin > file1

\$ls /bin >> file1

Lệnh echo : dùng để xuất dữ liệu ra thiết bị xuất chuẩn (mặc định là màn hình).

Cú pháp: echo "This is a message".

☞ Ví dụ:

\$ echo "Hello World !"

Hello World !

Tuy nhiên ta có thể chuyển hướng dữ liệu xuất ra các thiết bị xuất khác thiết bị xuất chuẩn bằng cách sử dụng cơ chế định hướng lại.

✎ Ví dụ:

```
$ echo "Hello World !" > file.txt
```

Dùng lệnh ls để kiểm tra lại kích thước của tập tin file.txt

```
$ ls -l file.txt
```

```
-rw-r--r-- 1 root root 10 Dec 28 14:13 file.txt
```

7.3 Đổi hướng đầu ra lỗi chuẩn:

- Đổi hướng vào một tập tin:

```
$lệnh      2 >   file1
```

Hoặc:

```
$lệnh      2 >>  file1
```

- Đổi hướng vào tập tin số liệu (vào đầu ra chuẩn):

```
$lệnh      >    file1  2 >  &1
```

- Đổi hướng vào tập tin null:

```
$lệnh      2 >   /dev/null
```

7.4 Đổi hướng đầu vào chuẩn:

- Số liệu vào chuẩn từ một tập tin:

```
$lệnh      <    file1
```

- Đổi hướng lệnh từ đầu vào chuẩn:

```
$lệnh      <<   STRING
```

Số liệu và/hoặc lệnh mà lệnh này cần đọc STRING

Cách đổi hướng này cho phép ta đưa các khai báo cho một lệnh trong shell script

✎ Ví dụ: dùng lệnh cat tạo một tập tin:

```
$ cat > file1 <<OK
```

```
Hello
```

```
World !
```

```
OK
```

```
$ cat file1
```

```
Hello
```

```
World !
```

```
$
```


7.5 Chuyển dữ liệu giữa các tiến trình

Hai dòng lệnh:

```
$lệnh1 > temp
```

```
$lệnh2 < temp
```

Có thể được thay thế bằng một dòng lệnh sau:

```
$lệnh1 | lệnh2
```

Khi đó đầu ra chuẩn của lệnh **lệnh1** sẽ là đầu vào chuẩn của lệnh **lệnh2**, và tập tin trung gian **temp** là không cần thiết.

Ký hiệu | gọi là ống (pipe)

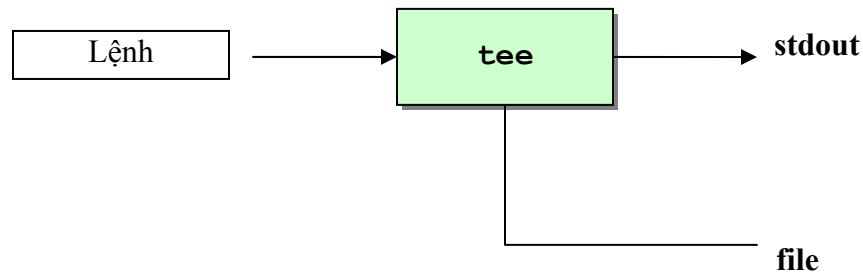
✎ Ví dụ:

```
$ls -l | pg
```

7.6 Đổi hướng kép (double) đầu ra chuẩn:

Lệnh **tee** cho phép đổi hướng kép đầu ra chuẩn: vừa hướng dữ liệu đầu ra về hướng khác, vừa hiển thị ra màn hình

```
$lệnh | tee file
```



Đổi hướng kép đầu ra chuẩn

✎ Ví dụ:

```
$ps -ef | tee file_ps | grep $LOGNAME
```

Có thể dùng **tee** với tùy chọn **-a** (**tee -a**), khi đó dữ liệu đầu ra sẽ được ghi tiếp vào cuối tập tin **file_ps**

8 Bài tập

Sinh viên phải đọc kỹ phần lý thuyết và làm các ví dụ trên trước khi làm phần bài tập

8.1 Bài 1

Cho biết thư mục home của người dùng hiện đang làm việc và mô tả các quyền trên thư mục home của người dùng đó (sinh viên cần chỉ ra chi tiết mục từ của thư mục home).

8.2 Bài 2

Liệt kê tên tất cả các thư mục dưới /. Mô tả nội dung và các quyền trên từng thư mục (sinh viên cần chỉ ra chi tiết mục từ của từng thư mục).

8.3 Bài 3

Hãy giải thích chi tiết mục từ thư mục:

drwxr-xr-- 2 abc users 1024 Mar 7 12:35 ddd

8.4 Bài 4

Trong hệ thống tập tin linux, thư mục nào có chứa các tập tin cấu hình hệ thống ?

8.5 Bài 5

Cho biết tên của hai thư mục trong hệ thống tập tin linux có chứa các tập tin thư viện lệnh ?

8.6 Bài 6

Hãy cho biết sự khác nhau giữa thư mục / (mà nó được gọi là thư mục gốc - root) và thư mục /root ?

8.7 Bài 7

Hiện thị thư mục hiện hành của người dùng, sau đó thực hiện tuần tự các lệnh sau và cho biết tên thư mục hiện hành tương ứng:

1. `cd ~`
2. `cd .`
3. `cd ..`
4. `cd ten_tai_khoan_login`

8.8 Bài 8

Giả sử thư mục hiện hành là /var/spool. Hãy cho biết tên thư mục tương ứng khi thực hiện tuần tự các lệnh sau:

```
cd mail  
cd ../../spool/mail
```

8.9 Bài 9

Chuyển đến thư mục /var và trả lời các câu hỏi sau:

1. Có bao nhiêu thư mục con dưới var ?
2. Thử xem nội dung thư mục /sbin, thông báo gì được hiển thị ? tại sao ?
3. Tạo thư mục mới **userxxx** là thư mục con của var, thông báo gì được hiển thị ? tại sao ?
4. Chuyển đến thư mục home của người dùng và tạo một thư mục con **dddd** (**dddd** là mã số của sinh viên), thông báo gì được hiển thị ? tại sao ?
5. Chuyển đến thư mục **dddd** và tạo một tập tin mới **ffff** với nội dung “This is a message”. Cho biết kết quả
6. Vẫn ở trong thư mục **dddd**. Thực hiện xóa thư mục **dddd** dùng câu lệnh **rmdir**, thông báo gì được hiển thị ? tại sao ?

8.10 Bài 10

Liệt kê tất cả các tập tin trong thư mục home của người dùng hiện đang làm việc và cho biết các tập tin ẩn

8.11 Bài 11

Thực hiện các yêu cầu sau:

1. Hiển thị 4 dòng cuối của tập tin /etc/passwd
2. Hiển thị 6 dòng cuối của tập tin /etc/group
3. Hiển thị 8 dòng cuối của tập tin /etc/login.defs
4. Hiển thị 20 dòng cuối của tập tin /etc/login.defs và cho nhận xét

8.12 Bài 12

Giả sử người dùng hiện ở tại thư mục home của mình, lần lượt thực hiện các yêu cầu sau:

1. Sử dụng câu lệnh touch để tạo 7 tập tin : one, two, three, four, five, r1 và r2. Liệt kê thư mục để đảm bảo rằng việc tạo là thành công. Cho biết quyền trên các tập tin.
2. Cũng với yêu cầu như câu 1 nhưng các tập tin được tạo trong thư mục /var. Cho biết kết quả thực hiện. So sánh với kết quả câu 1 và giải thích.
3. Làm một bản sao của tập tin .bash_profile được gọi là bprofile. Kiểm tra lại nội dung thư mục.
4. Tạo một liên kết có tên lbprofile mà nó trỏ đến tập tin bprofile. Cho biết sự khác biệt giữa lbprofile và bprofile.
5. Chép 7 tập tin trên đến thư mục /tmp. Kiểm tra lại nội dung thư mục /tmp
6. Chép 7 tập tin trên đến thư mục /root. Kết quả thực hiện. So sánh với kết quả câu 5 và giải thích.
7. Chép tập tin /etc/passwd đến thư mục home của mình. So sánh quyền tương tác trên tập tin passwd vừa được sao chép với tập tin /etc/passwd. Giải thích.
8. Tạo 2 thư mục con dir1 và dir2. Sau đó tạo thư mục con dir2a trong thư mục dir2. Cho biết quyền trên các thư mục.

9. Chuyển tập tin one đến thư mục dir1. Kiểm tra kết quả thực hiện
10. Dùng một câu lệnh đơn để chuyển tất cả các tập tin mà có ký tự đầu tiên là “f” đến thư mục dir2. Kiểm tra lại thư mục dir2
11. Dùng một câu lệnh đơn để chuyển hai tập tin r1 và r2 đến thư mục dir2a (không dùng ký tự đại diện *,?). Kiểm tra lại thư mục dir2a
12. Xóa thư mục dir2a

8.13 Bài 13

Chuyển đến thư mục home của người dùng, tuân tự thực hiện các câu lệnh sau:

1. Tạo chương trình (script) hello với nội dung :
 print “Hello”
 print “How are you ?”
2. Tạo chương trình reply với nội dung :
 print “Hello”
 print “Fine. And you ?”
3. Xem các quyền trên các tập tin trên
4. Chạy 2 chương trình trên. Cho biết kết quả và nhận xét
5. Cấp quyền thực thi cho 2 chương trình trên. Chạy lại chúng, cho biết kết quả và nhận xét
6. Bỏ các quyền thâm nhập tới hello và reply của những người cùng nhóm (group) và của những người khác (other)
7. Thay đổi quyền thâm nhập sao cho những người cùng nhóm có khả năng đọc và chạy các tập tin hello và reply.
8. Hãy thử thay đổi người hoặc nhóm sở hữu của tập tin trên (hello hoặc reply). Sau đó xóa tập tin trên. Cho biết kết quả và giải thích.
9. Tạo tập tin mới có tên hello1. Nội dung tập tin : echo Hello1. Sau đó sửa quyền thâm nhập để cho phép người dùng hiện đang làm việc có thể chạy được. Điều gì xảy ra nếu như một người dùng khác chạy tập tin hello1 đó. Cho biết kết quả và giải thích.
10. Để tập tin hello1 có thể chạy được bởi người dùng nào đó trong hệ thống (user1 chẳng hạn) thì cần có điều kiện gì ? Đưa ra các giải pháp có thể.
11. Để người dùng nào đó trong hệ thống (user1 chẳng hạn) cấp quyền thực thi cho người dùng khác (user2 chẳng hạn) thì đòi hỏi người dùng user1 cần có những quyền gì ? Đưa ra giải pháp cụ thể (các câu lệnh để thực hiện).
12. Đổi tên tập tin hello1 thành rm. Thực hiện việc xóa một tập tin nào đó đang tồn tại (hello chẳng hạn). Điều gì sẽ xảy ra ? giải thích.

8.14 Bài 14

Phân biệt các lệnh more, less, head, tail và cat. Cho ví dụ minh họa

8.15 Bài 15

Sử dụng lệnh find để thực hiện các yêu cầu sau:

1. Tìm tất cả các tập tin có phần mở rộng .html
2. Tìm tất cả các tập tin thuộc về người dùng root
3. Tìm thư mục có tên là lib trong thư mục /usr
4. Đưa ra một ví dụ về lệnh find mà khi thực hiện nó sẽ hiển thị thông báo lỗi: "permission denied". Giải thích.
5. Liệt kê tất cả các tập tin mà chúng bị sửa đổi cách đây không quá 1 ngày
6. Liệt kê tất cả các thư mục rỗng

8.16 Bài 16

Sử dụng lệnh grep thực hiện các yêu cầu sau:

1. Hiển thị số dòng và các dòng của tập tin /etc/passwd mà có chứa chuỗi root
2. Hiển thị số dòng và các dòng của tập tin /etc/passwd mà không chứa chuỗi root
3. Liệt kê các tập tin trong thư mục /etc/init.d mà có chứa chuỗi echo
4. Hiển thị các dòng trong tập tin /etc/login.defs mà không là các dòng chú thích
5. Hiển thị danh sách người dùng mà tập tin shell khi đăng nhập của họ là /bin/bash

8.17 Bài 17

Giả sử người dùng hiện đang ở tại thư mục home của mình, sử dụng lệnh linux để thực hiện các yêu cầu sau:

1. Lưu danh sách các thư mục con trong thư mục hiện hành vào tập tin one. Sau đó kiểm tra lại nội dung tập tin one.
2. Hiển thị danh sách người dùng đã đăng nhập ra màn hình đồng thời lưu danh sách đó vào tập tin users. Sau đó kiểm tra lại nội dung tập tin users.
3. Hiển thị tất cả các tiến trình hiện đang chạy đồng thời lưu những thông tin đó vào cuối tập tin users.
4. Hiển thị nội dung thư mục /etc theo từng trang
5. Đếm số byte, số từ (word) và số dòng (line) trong tập tin /etc/passwd
6. Sắp xếp nội dung tập tin /etc/group
7. Hiển thị nội dung thư mục var/spool/mail được sắp xếp theo kích thước
8. Liệt kê các dịch vụ sử dụng giao thức tcp
9. Nhập vào một danh sách họ tên sinh viên từ bàn phím đồng thời lưu danh sách đó đã được sắp xếp vào tập tin students. Kiểm tra lại nội dung tập tin students

8.18 Bài 18

Tạo tập tin “commands” trong thư mục home của người dùng có nội dung như sau:

cal	/usr/bin	9416
date	/bin	2860
who	/usr/bin	8692
cat	/bin	18056
lpd	/usr/sbin	42012

Sau đó thực hiện các yêu cầu:

1. Giải thích output của pipeline khi thực thi lệnh : `sort -r commands | head -1`
2. Giải thích output của pipeline khi thực thi lệnh : `grep ^c commands | wc -l`
3. Giải thích output của pipeline khi thực thi lệnh :
`grep at commands | sort | head -1`