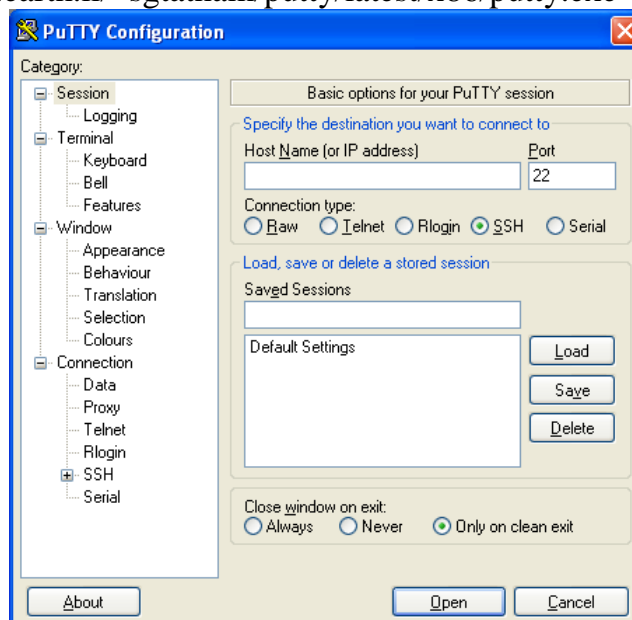


LAB: LINUX COMMANDS

1.1 Chuẩn bị cho môn học

Tại mỗi máy tính, sinh viên sử dụng phần mềm PuTTY để truy cập vào Linux server, phần mềm này có thể lấy từ địa chỉ:

<http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe>



Hình 1 – Giao diện phần mềm PuTTY

Sau khi mở phần mềm PuTTY, tại ô “**Host Name (or IP address)**”, nhập địa chỉ của máy chủ Linux server:

10.22.14.15

Sau đó ấn phím Enter hoặc click vào nút Open, sẽ có một cửa sổ cảnh báo như sau:



Hình 2 – Thông báo xuất hiện khi đăng nhập

Đây chỉ là thông báo nhằm đảm bảo tính xác thực của việc đăng nhập vào hệ thống. Bạn có thể lựa chọn “Yes” hoặc “No”, nếu chọn “Yes” thì hệ thống sẽ ghi nhớ và lần đăng nhập sau của bạn sẽ không xuất hiện cửa sổ cảnh báo như vậy nữa.

Tiếp theo, phần mềm PuTTY sẽ chuyển sang màn hình màu đen với yêu cầu khai báo tên đăng nhập và mật khẩu. Bạn làm theo các bước sau:

- **Login as:** nhập tên truy nhập của từng sinh viên, theo quy tắc:
fun.edu.vn\username_của_bạn
Ví dụ: **fun.edu.vn\anhnk12345**
- **Password:** nhập mật khẩu hiện tại của sinh viên đó, chú ý rằng mật khẩu sẽ không hiện ra khi bạn gõ để đảm bảo bí mật cho mật khẩu.

Một số lưu ý khi nhập thông tin của mỗi sinh viên:

- Nhớ gõ đầy đủ giá trị **fun.edu.vn** trước username của bạn.
- Có thể vì một số lý do nào đó (quá nhiều người cùng truy cập một lúc, mạng có sự cố tạm thời, v.v.) thì quá trình đăng nhập có thể không được, khi đó sinh viên nên đóng phần mềm PuTTY rồi mở lại, sau đó thực hiện việc đăng nhập như trên một lần nữa.

Sau khi đăng nhập thành công, màn hình phần mềm PuTTY sẽ chuyển vào chế độ nhập lệnh. Toàn bộ các thao tác thực hành môn học OS&N sẽ được thực hiện trên màn hình phần mềm PuTTY này.

Sinh viên có thể cài thêm hệ điều hành Linux trên máy tính cá nhân để phục vụ tốt hơn cho việc học tập môn học. Hiện tại có rất nhiều phiên bản (distro) Linux, có thể sử dụng các đĩa Live CD để chạy thử trước để chọn ra bản Linux phù hợp với từng người rồi mới cài đặt vào máy. Hiện tại có 2 bản Linux được nhiều người sử dụng là Fedora và Ubuntu, có thể download bản Live CD tại địa chỉ sau:

<http://mirror-fpt-telecom.fpt.net/fedora/linux/releases/11/Live/i686/Fedora-11-i686-Live.iso> (688MB)

<http://mirror-fpt-telecom.fpt.net/ubuntu-releases/9.04/ubuntu-9.04-desktop-i386.iso> (699MB)

Để tham khảo các đĩa cài đầy đủ, truy cập vào trang chủ của Fedora và Ubuntu:

<http://fedoraproject.org>

<http://www.ubuntu.com>

Ngoài ra, trên Internet còn có nhiều sự lựa chọn khác, có thể đáp ứng nhu cầu tìm hiểu Linux của sinh viên. Có thể tham khảo tại: <http://distrowatch.com>

1.2 Cấu trúc tệp, thư mục trong Linux

Cấu trúc file và thư mục trong hệ điều hành (HĐH) Linux có sự khác biệt so với HĐH Windows. Cần hiểu rõ cấu trúc của file, thư mục trong Linux để tránh nhầm lẫn trong quá trình thực hành môn học.

Trong HĐH Windows, có thể có nhiều ổ đĩa mang tên “C, D, E,...” khác nhau. Tuy nhiên, trong Linux, chỉ có một “ổ đĩa gốc” duy nhất mang tên “/”, dưới đó là các thư mục và các file cấu thành lên hệ điều hành, và kể cả các file của người dùng.

Ký tự “/” vừa có ý nghĩa là thư mục gốc trên cùng, vừa có ý nghĩa phân cách giữa các thư mục. Ví dụ đường dẫn [/home/sv10/monhoc/osn/baitap](#) bao gồm 5 ký tự “/”, trong đó ký tự “/” đầu tiên mang ý nghĩa đó là thư mục gốc, 4 ký tự “/” còn lại nhằm phân cách các thư mục con của thư mục gốc, cụ thể: trong thư mục gốc có thư mục [home](#), trong home có thư mục [sv10](#),..., trong thư mục [osn](#) có thư mục [baitap](#).

Như đã nói ở trên, trong Linux chỉ có một “ổ đĩa gốc” duy nhất. Với Windows, hoàn toàn có thể tồn tại 2 hoặc nhiều hơn các thư mục cùng tên trong điều kiện chúng nằm ở các ổ đĩa khác nhau, như sau:

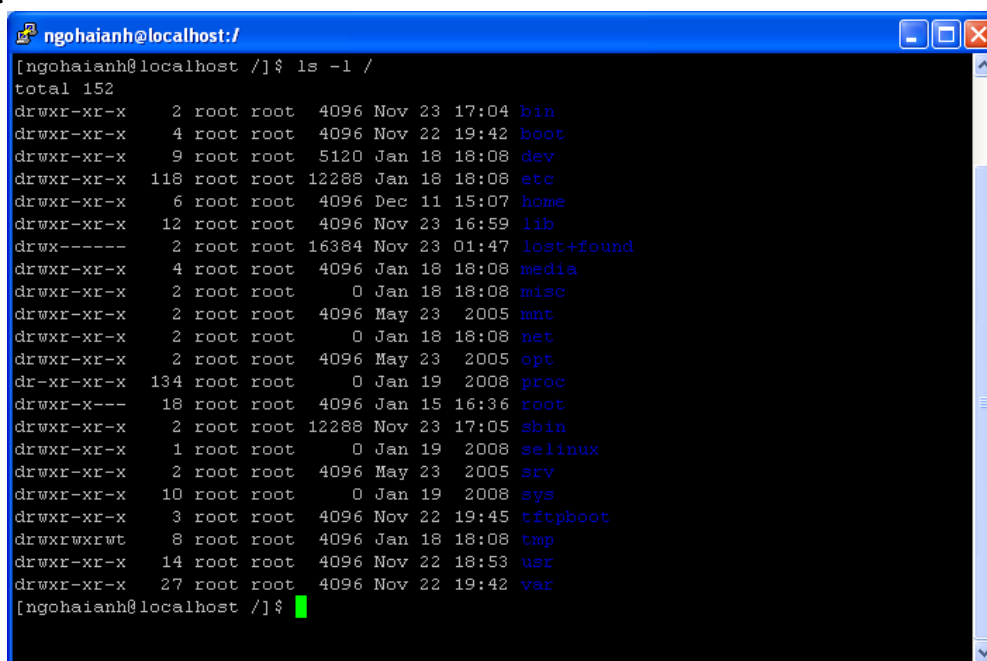
[C:/home/sv10/monhoc/osn/baitap](#)

[D:/home/sv10/monhoc/osn/baitap](#)

[E:/home/sv10/monhoc/osn/baitap](#)

Nhưng trong Linux không thể tồn tại 3 ổ đĩa như vậy.

Sau khi cài đặt Linux, sẽ có những thư mục mặc định được cài đặt. Tùy từng phiên bản Linux khác nhau mà các thư mục mặc định này sẽ khác nhau. Ví dụ với bản Fedora Core thì các thư mục mặc định sau khi cài đặt xong như sau:



```
[ngohaianh@localhost ~]$ ls -l /
total 152
drwxr-xr-x  2 root root  4096 Nov 23 17:04 bin
drwxr-xr-x  4 root root  4096 Nov 22 19:42 boot
drwxr-xr-x  9 root root 5120 Jan 18 18:08 dev
drwxr-xr-x 118 root root 12288 Jan 18 18:08 etc
drwxr-xr-x  6 root root  4096 Dec 11 15:07 home
drwxr-xr-x 12 root root  4096 Nov 23 16:59 lib
drwx----- 2 root root 16384 Nov 23 01:47 lost+found
drwxr-xr-x  4 root root  4096 Jan 18 18:08 media
drwxr-xr-x  2 root root    0 Jan 18 18:08 misc
drwxr-xr-x  2 root root  4096 May 23 2005 mnt
drwxr-xr-x  2 root root    0 Jan 18 18:08 net
drwxr-xr-x  2 root root  4096 May 23 2005 opt
dr-xr-xr-x 134 root root    0 Jan 19 2008 proc
drwxr-xr-x 18 root root  4096 Jan 15 16:36 root
drwxr-xr-x  2 root root 12288 Nov 23 17:05 sbin
drwxr-xr-x  1 root root    0 Jan 19 2008 selinux
drwxr-xr-x  2 root root  4096 May 23 2005 srv
drwxr-xr-x 10 root root    0 Jan 19 2008 sys
drwxr-xr-x  3 root root  4096 Nov 22 19:45 tmp
drwxrwxrwt  8 root root  4096 Jan 18 18:08 tmp
drwxr-xr-x 14 root root  4096 Nov 22 18:53 usr
drwxr-xr-x 27 root root  4096 Nov 22 19:42 var
```

Hình 3 – Các thư mục mặc định sau khi cài Fedora Linux

Tiếp theo, chúng ta cần chú ý đến 2 thư mục đặc biệt trong Linux, đó là thư mục gốc và thư mục người dùng.

Thư mục gốc: ký hiệu bằng ký tự “/”, đây là “cha” của mọi thư mục khác. Như hình 3, các thư mục mặc định [/home](#), [/var](#), [/temp](#),... đều thuộc thư mục gốc này. Để trở về thư mục gốc, sử dụng lệnh **cd** như sau:

\$cd /

Thư mục người dùng: trong HĐH Linux: với mỗi người dùng khác nhau thì có một thư mục tương ứng với người dùng đó. Và các thư mục này đều được đặt trong thư mục [/home](#). Giả sử trên máy tính cài Linux có 3 user là *sv01*, *sv02*, *sv03* thì cấu trúc thư mục [/home](#) sẽ có 3 thư mục con là:

[/home/sv01](#)

[/home/sv02](#)

[/home/sv03](#)

Các thao tác của các user *sv01*, *sv02*, *sv03* sẽ chỉ thực hiện được trong thư mục tương ứng của mình.

Để trở về thư mục người dùng, sử dụng lệnh *cd* như sau:

\$cd ~

Các thư mục đặc biệt: Thông thường, trong hệ điều hành Linux luôn tồn tại một số thư mục đặc biệt như sau:

/bin	: Thư mục này chứa các file
/boot	: Các image file của kernel dùng cho quá trình khởi động thường đặt trong thư mục này
/dev	: Các thư mục này chứa các file thiết bị.
/etc	: Thư mục này chứa các file cấu hình hệ thống
/home	: Thư mục này chứa các thư mục con đại diện cho mỗi user khi đăng nhập. Khi người quản trị tạo tài khoản cho người dùng, là họ cấp cho người dùng một thư mục con trong /home.
/lib	: Thư mục này chứa các file thư viện <i>.so</i> hoặc <i>.a</i> . Các thư viện C và các thư viện liên kết động cần cho chương trình khi chạy và cho toàn bộ hệ thống.
/lost+found	: Khi hệ thống khởi động hoặc khi chạy chương trình <i>fsck</i> nếu tìm thấy chuỗi dữ liệu nào bị thất lạc trên đĩa cứng không liên quan đến các tập tin Linux sẽ gộp chúng lại trong thư mục này
/mnt	: Thư mục này chứa các thư mục kết gán tạm thời đến các ổ đĩa hay thiết bị khác.
/sbin	: Thư mục này chứa các file hay chương trình thực thi của hệ thống thường chỉ cho phép sử dụng bởi người quản trị
/tmp	: Thư mục lưu các file tạm thời trong quá trình chạy
/usr	: Chứa nhiều chương trình con, đặc biệt là <i>/usr/local</i> . Nếu nâng cấp hệ thống thì các chương trình được cài đặt trong <i>/usr/local</i> vẫn giữ nguyên.
/var	: Chứa các file biến thiên bất thường, điển hình là các file dùng làm hàng đợi chứa các dữ liệu cần đưa ra máy in hoặc các hàng đợi chứa mail.

Ngoài ra có một số thư mục quan trọng khác nên quan tâm:

/usr/include và **/usr/local/include**: Chứa các file khai báo hàm (header) cần dùng khi biên dịch chương trình nguồn sử dụng các thư viện của hệ thống.

/usr/src: Chứa mã nguồn (gồm cả mã nguồn của Linux).

/usr/man: Chứa tài liệu hướng dẫn.

1.3 Các lệnh cơ bản trong Linux

1.3.1 Khái niệm về dòng lệnh

Giao diện dòng lệnh được cung cấp khi đăng nhập vào Linux, bao gồm dòng đơn có thể gõ các lệnh kèm theo *tùy chọn (option)* và *đối số (argument)*. Dấu nhắc shell đánh dấu vị trí bắt đầu của một dòng lệnh. Với mỗi một loại

shell khác nhau thì có dấu nhắc khác nhau. Với BASH shell thì dấu nhắc hệ thống là \$. Sau khi đăng nhập vào hệ thống nếu xuất hiện dấu nhắc thì bạn có thể gõ câu lệnh tại dấu nhắc này.

1.3.2 Cú pháp dòng lệnh

\$command option argument

Từ đầu tiên nhập vào trên dòng lệnh phải là tên của câu lệnh. Tiếp theo là các lựa chọn và cuối cùng là các đối số. Các phần trên dòng lệnh phải được cách nhau ít nhất một dấu cách (space). Trước các option thường là dấu trừ "-". Sau khi gõ lệnh, shell sẽ kiểm tra xem lệnh đó có tồn tại thực sự hay không, nếu không shell sẽ thông báo lỗi. Ta có thể sử dụng tại dấu nhắc các phím mũi tên lên, xuống để lấy các lệnh đã thực hiện trước đó.

1.3.3 Một số lệnh cơ bản

Bảng dưới đây liệt kê một số lệnh cơ bản của Linux. Các sinh viên sẽ tìm hiểu hoạt động của các lệnh này bằng cách gõ trên màn hình và quan sát kết quả.

man command	view help
ls	list files and directories
ls -a	list all files and directories, hidden files included
ls -l	list all files and directories, more detail, longer
ls *	list all files and directories with all files inside the directories
mkdir directoryname	make a directory
cd directoryname	change to named directory
cd ..	change to parent directory
cd	change to home-directory
cd ~	change to home-directory
cd /	change to root directory
pwd	display the path of the current directory
touch filename	make an empty file
cat >> filename	make a file with contents
^D	finish a job (Ctrl+D)
clear	clean screen
history	list used commands before
date	get current date
cal	show calendar
cp file1 file2	copy file1 and call it file2
cp -r dir1 dir2	copy dir1 (directory) and call it dir2
mv file1 file2	move or rename file1 to file2
rm filename	remove a file
rmdir directoryname	remove an empty directory
rm -r directoryname	remove a directory
cat filename	Display a file (q, b, space bar)
more filename	display a file a page at a time
head -n filename	display the first n lines of a file, default n=10
tail -n filename	display the last n lines of a file, default n=10
grep 'keyword' file	search a file for keywords
wc filename	count number of lines/words/characters in file
command > file	redirect standard output to a file
command >> file	append standard output to a file

<code>command < file</code>	redirect standard input from a file
<code>command1 command2</code>	pipe the output of <i>command1</i> to the input of <i>command2</i>
<code>cat file1 file2 > file0</code>	concatenate <i>file1</i> and <i>file2</i> to <i>file0</i>
<code>sort</code>	sort data
<code>who</code>	list users currently logged in
<code>whoami</code>	“Who Am I”, username of current user
<code>*</code>	match any number of characters
<code>?</code>	match one character
<code>[]</code>	match one character, use a set
<code>man command</code>	read the online manual page for a command
<code>whatis command</code>	brief description of a command
<code>apropos keyword</code>	match commands with keyword in their man pages
<code>ls -lag</code>	list access rights for all files
<code>chmod [options] file</code>	change access rights for named file
<code>command &</code>	run command in background
<code>^C</code>	kill the job running in the foreground (Ctrl+C)
<code>^Z</code>	suspend the job running in the foreground (Ctrl+Z)
<code>bg</code>	background the suspended job
<code>jobs</code>	list current jobs
<code>fg %1</code>	foreground job number 1
<code>kill %1</code>	kill job number 1
<code>ps</code>	list current processes
<code>kill 26152</code>	kill process number 26152

1.3.4 Đường dẫn tới thư mục

Với cơ chế dòng lệnh (command line), nắm rõ được vấn đề đường dẫn sẽ giúp người dùng rút ngắn thời gian “đi lại”.

- Khi muốn thao tác trên một file hoặc thư mục nào đó, bạn cần phải biết đầy đủ đường dẫn tới nó. Nếu không chỉ rõ đường dẫn, hệ thống sẽ hiểu đường dẫn chính là đến thư mục làm việc hiện thời.
- Dù bạn đứng ở bất cứ đâu cũng có thể thực hiện các lệnh khi chỉ đúng đường dẫn từ nơi bạn đứng tới nơi mà câu lệnh tác động đến.
- Với đường dẫn tới một thư mục, thì ký tự “/” ở thư mục cuối là không bắt buộc. Tức là:

`/home/sv10/osn/` tương đương với `/home/sv10/osn`

- Các ký hiệu thư mục đặc biệt của Linux, hay được sử dụng trong đường dẫn:
 - Thư mục gốc : `/` (dấu gạch chéo)
 - Thư mục người dùng : `~` (dấu ngã)
 - Thư mục hiện thời : `.` (dấu chấm)
 - Thư mục cha của thư mục hiện thời : `..` (dấu chấm chấm)

Nhờ đó, ta có thể sử dụng các ký hiệu đặc biệt trên để dẫn đến thư mục mong muốn làm việc một cách nhanh gọn hơn.

Vì những lý do trên, người dùng cần nắm rõ được đường dẫn từ một *điểm đứng* (thư mục) bất kỳ tới một thư mục bất kỳ.

1.4 Điều hướng dòng ra chuẩn STDOUT với > và >>

Giả sử thay vì hiển thị danh sách các file ra màn hình, bạn muốn đưa danh sách đó vào một file. Nói cách khác, bạn cần điều hướng dòng ra chuẩn vào một file thay vì ra màn hình. Để làm điều này, ta dùng *toán tử điều hướng* > ngay sau câu lệnh và tiếp theo là tên file muốn chuyển dữ liệu vào.

Ví dụ 1:

```
$ls > newdata
```

Kết quả của câu lệnh *ls* sẽ được ghi vào file “newdata” thay vì hiển thị ra màn hình. Cần lưu ý rằng khi điều hướng, một file đích mới sẽ được tạo ra, nếu file này đã tồn tại thì nó sẽ bị ghi đè.

Sử dụng toán tử >>, dữ liệu từ STDOUT sẽ được ghi thêm vào cuối file điều hướng thay vì ghi đè.

Ví dụ 2:

```
$ls > newdata
```

```
$date >> newdata
```

Lúc này file “newdata” có nội dung là kết quả của câu lệnh *ls* và câu lệnh *date*.

1.5 Trình soạn thảo Vim

Chương trình xử lý văn bản luôn luôn có ở mọi phiên bản Linux là *vi* (*visual interpreter*). Mặc dù có những bất tiện nhất định trong việc sử dụng, nhưng vì lại rất gọn nhẹ và được coi như một trình xử lý văn bản sẵn có với Linux. Hiện tại, các hệ điều hành thường dùng một phần mềm soạn thảo text được phát triển từ *vi* là *vim* (*Vi IMproved*), *vim* có một số ưu điểm hơn so với *vi*. Cách tạo mới một file bằng *vim* rất đơn giản:

```
$vim tên_file
```

Ví dụ:

```
$vim mydoc
```

Trình soạn thảo *vim* chạy ở hai chế độ khác nhau:

- Ở chế độ câu lệnh, những gì bạn gõ vào sẽ được hiểu như là ra lệnh cho *vim*. Lệnh sẽ bảo *vim* lưu file, thoát khỏi *vim*, xóa bỏ... Mặc định sau khi dùng lệnh *\$vim mydoc* ở trên thì sẽ chuyển vào chế độ câu lệnh.
- Ở chế độ nhập liệu (chế độ INSERT: bạn sẽ nhìn thấy ở góc dưới bên trái màn hình có chữ INSERT), những gì bạn gõ vào được máy hiểu là nội dung của tệp mà bạn đang chỉnh sửa. Theo chế độ này, *vim* hoạt động như một máy đánh chữ đơn thuần.

Khi chỉnh sửa văn bản, bạn tự do chuyển đổi giữa hai chế độ trên

- Chuyển từ chế độ câu lệnh sang chế độ nhập liệu: dùng phím *i*
- Chuyển từ chế độ nhập liệu sang chế độ câu lệnh: dùng phím *Esc*

Lựa chọn	Ý nghĩa
:w	Lưu những thay đổi của file
:wq	Lưu những thay đổi và thoát khỏi vim
:q	Thoát khỏi vim
:q!	Thoát khỏi vim mà không lưu những thay đổi
:w tên_file	Lưu nội dung đang soạn thảo thành file mới

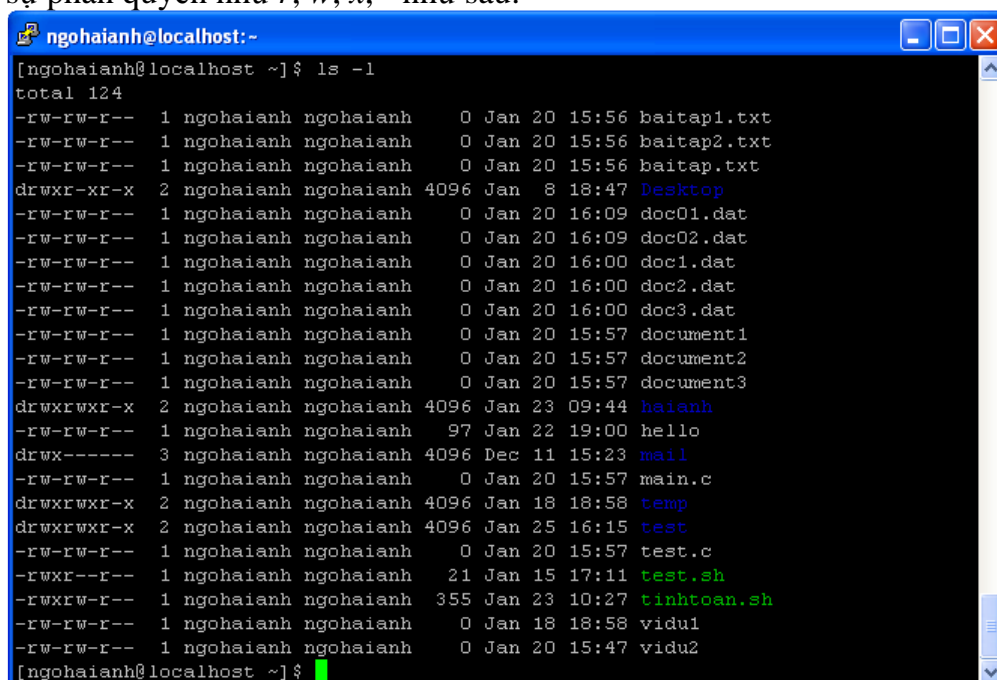
- Sắp xếp số user đang đăng nhập : `$who | sort`
- Đếm số user đang đăng nhập : `$who | wc -l`

1.7 Phân quyền trên file và thư mục

Mỗi file và thư mục trong Linux đều chứa tập các quyền xác định tài khoản nào có thể truy cập vào file hay thư mục đó.

- Có ba kiểu người dùng trên file: *người sở hữu file (owner, user -u)* là người tạo ra file; *nhóm người dùng (group -g)* (thường là những người cùng nhóm với người sở hữu) và *những người dùng khác (other -o)*.
- Có ba kiểu truy cập trên file: *đọc (read -r)*, *ghi (write -w)* và *thực thi (executable -x)*. Khi một file được tạo ra thì mặc định người tạo có quyền *đọc, ghi* file đó¹.

Liệt kê chi tiết file bằng lệnh `ls -l`, ta thấy ví dụ file “test.sh” có những ký tự nói lên sự phân quyền như *r, w, x*, - như sau:

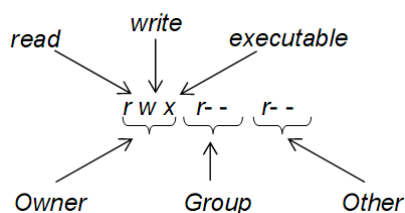


```
ngohaianh@localhost:~$ ls -l
total 124
-rw-rw-r-- 1 ngohaianh ngohaianh 0 Jan 20 15:56 baitap1.txt
-rw-rw-r-- 1 ngohaianh ngohaianh 0 Jan 20 15:56 baitap2.txt
-rw-rw-r-- 1 ngohaianh ngohaianh 0 Jan 20 15:56 baitap.txt
drwxr-xr-x 2 ngohaianh ngohaianh 4096 Jan 8 18:47 Desktop
-rw-rw-r-- 1 ngohaianh ngohaianh 0 Jan 20 16:09 doc01.dat
-rw-rw-r-- 1 ngohaianh ngohaianh 0 Jan 20 16:09 doc02.dat
-rw-rw-r-- 1 ngohaianh ngohaianh 0 Jan 20 16:00 doc1.dat
-rw-rw-r-- 1 ngohaianh ngohaianh 0 Jan 20 16:00 doc2.dat
-rw-rw-r-- 1 ngohaianh ngohaianh 0 Jan 20 16:00 doc3.dat
-rw-rw-r-- 1 ngohaianh ngohaianh 0 Jan 20 15:57 document1
-rw-rw-r-- 1 ngohaianh ngohaianh 0 Jan 20 15:57 document2
-rw-rw-r-- 1 ngohaianh ngohaianh 0 Jan 20 15:57 document3
drwxrwxr-x 2 ngohaianh ngohaianh 4096 Jan 23 09:44 haianh
-rw-rw-r-- 1 ngohaianh ngohaianh 97 Jan 22 19:00 hello
drwx----- 3 ngohaianh ngohaianh 4096 Dec 11 15:23 mail
-rw-rw-r-- 1 ngohaianh ngohaianh 0 Jan 20 15:57 main.c
drwxrwxr-x 2 ngohaianh ngohaianh 4096 Jan 18 18:58 temp
drwxrwxr-x 2 ngohaianh ngohaianh 4096 Jan 25 16:15 test
-rw-rw-r-- 1 ngohaianh ngohaianh 0 Jan 20 15:57 test.c
-rwxr--r-- 1 ngohaianh ngohaianh 21 Jan 15 17:11 test.sh
-rwxrw-r-- 1 ngohaianh ngohaianh 355 Jan 23 10:27 tinhtoan.sh
-rw-rw-r-- 1 ngohaianh ngohaianh 0 Jan 18 18:58 vidu1
-rw-rw-r-- 1 ngohaianh ngohaianh 0 Jan 20 15:47 vidu2
ngohaianh@localhost:~$
```

Hình 1 – Xem quyền trên các tệp, thư mục bằng lệnh `ls` với tham số `-l`

Tương ứng với file “test.sh” có 9 ký tự nói lên quyền đã được thiết lập trên file này:

¹ Có hai điểm cần lưu ý ở đây: **thứ nhất**: quyền *write* trên một file có ý nghĩa không chỉ là “*ghi*” mà ý nói đến quyền được *sửa, xóa, ghi đè*... trên file đó; **thứ hai**: theo suy nghĩ logic thông thường của bạn thì với người sở hữu (*owner*) một file thì người đó phải mặc định được cả ba quyền *read, write, executable* trên file đó, tuy nhiên vì lý do an toàn mà quyền thực thi không bao giờ tự động được cấp, ngay cả với đối tượng *owner*, câu lệnh cấp quyền thực thi phổ biến là `chmod u+x filename` với ý nghĩa là “*cấp quyền thực thi filename cho người sở hữu nó*”).



Ba ký tự đầu tiên biểu diễn *quyền cho người sở hữu* (được phép: đọc, ghi, thực thi), ba ký tự tiếp theo biểu diễn *quyền cho nhóm người dùng* (được phép: đọc) và ba ký tự cuối cùng biểu diễn *quyền cho những người khác* (được phép: đọc). Ký tự “-” thể hiện không có quyền.

Để thiết lập quyền cho file hay thư mục, ta dùng lệnh **chmod** theo cách sau:

\$chmod kiểu_người_dùng+quyền_thêm_vào_tên_file
\$chmod kiểu_người_dùng-quyền_bớt_đi_tên_file

Ví dụ 1: thêm quyền thực thi cho người sở hữu trên file “tesh.sh”:

\$chmod u+x test.sh

Ví dụ 2: Thêm quyền đọc, bỏ quyền ghi và thực thi cho những người dùng khác trên file “test.sh”:

\$chmod o+r-wx test.sh

Nếu muốn gán quyền cho tất cả các người dùng, ta có thể dùng tùy chọn *a* hoặc để trống tại *kiểu_người_dùng*.

Đặt quyền trên thư mục giống như đặt quyền trên file. Quyền *read* sẽ cho phép hiển thị nội dung thư mục, quyền *executable* cho phép di chuyển vào thư mục và quyền *write* cho phép tạo hay xóa các file trong thư mục. Khi bạn tạo một thư mục thì người sở hữu có tất cả các quyền trên thư mục đó.

Để thiết lập quyền truy cập, ngoài cách dùng ký hiệu như trên, ta có thể dùng các số [0-7] [0-7] [0-7] như sau:

Octal	Binary	Permissions
0	000	---
1	001	--x
2	010	-w-
3	011	-wx
4	100	r--
5	101	r-x
6	110	rw-
7	111	rxw

Ví dụ 3: lệnh sau sẽ cho phép cả 3 loại user được làm tất cả 3 thao tác *read*, *write*, *executable* trên một tệp!

\$chmod 777 vidu.dat

BÀI TẬP LAB 1

Bài 1: Thực hiện các thao tác sau:

0. Trở về thư mục người dùng bằng lệnh: `cd ~`
1. Tạo ra một thư mục mang tên của bạn, ví dụ: `minhlong`, `kykhoi`,...
2. Trong thư mục đó tạo ra một thư mục mang tên môn học: `OS_N`, trong thư mục này lại tạo tiếp 2 thư mục con có tên: `OS` và `N`
3. Trong thư mục `OS` tạo:
 - Một thư mục có tên `HW`
 - Một tệp có tên `test1` với nội dung như sau: “`He dieu hanh`”
4. Trong thư mục `N` tạo:
 - Một thư mục có tên `HW`
 - Một thư mục có tên `Assgn`
 - Một tệp rỗng có tên `test5`
5. Copy tệp `test1` sang thư mục `N` với tên là `test4`
6. Xóa tệp `test1` khỏi thư mục `OS`
7. Chèn thêm nội dung vào `test5`
8. Trở về thư mục gốc.

Bài 2: Thực hiện các thao tác sau:

- Trở về thư mục người dùng tương ứng với username bạn dùng để login.
Gợi ý: sử dụng lệnh `cd ~`, sau đó kiểm tra lại vị trí hiện hành bằng lệnh `pwd`
- Xóa toàn bộ tệp hiện có trong thư mục người dùng.
Gợi ý: sử dụng lệnh `rm *`
- Xóa toàn bộ thư mục hiện có trong thư mục người dùng.
Gợi ý: sử dụng lệnh `rm -r *`
(Nhắc lại: tham số `*` mang ý nghĩa là “tất cả”, còn tùy chọn `-r` dùng khi xóa thư mục bởi nó đảm bảo nếu thư mục khác rỗng thì xóa toàn bộ những gì trong đó)
- Tạo mới một tệp rỗng có tên `hoa.txt`
Gợi ý: dùng lệnh `touch hoa.txt`
- Nhập nội dung sau vào tệp `hoa.txt`:
`Hoa hong`
`Hoa lan`
`Hoa hue`
`Hoa nhai`
Gợi ý: dùng lệnh `cat > hoa.txt`, sau đó nhập nội dung, lưu ý phải có 1 dòng trống trước khi kết thúc nhập nội dung tệp, thoát bằng tổ hợp phím Ctrl+D)
- Tạo một thư mục có tên `lab2`, sau đó trong `lab2` lại tạo một thư mục có tên `hoa`
- Đứng ở vị trí thư mục `hoa`, sao chép tệp `hoa.txt` vào thư mục này với tên `vidu1.txt`
Gợi ý: `cp ../../hoa.txt vidu1.txt`
- Tạo một tệp mới có tên `vidu2.txt`, nội dung giống hết tệp `vidu1.txt`
Gợi ý: `mv vidu1.txt vidu2.txt`
- Trở về thư mục người dùng, xóa cả 2 thư mục `lab2` và `hoa`

- Gợi ý: dùng 2 lệnh `cd ~` và `rm -r lab2`
- Lúc này thư mục người dùng của bạn chỉ còn duy nhất tệp *hoa.txt*
 - Tạo một tệp *qua.txt* có nội dung:
`Cam`
`tao`
`Nho`
 - Đếm xem có bao nhiêu loại hoa và bao nhiêu loại quả.
Gợi ý: dùng lệnh `wc` với tùy chọn `-l` trên 2 tệp *hoa.txt* và *qua.txt* như sau:
`wc -l hoa.txt` và `wc -l qua.txt`
 - Tạo một tệp *hoaqu.txt* có nội dung:
`Duoi đây là các loại hoa quả:`
 - Gộp nội dung 2 tệp *hoa.txt* và *qua.txt* vào tệp *hoaqu.txt*
Gợi ý: dùng chuỗi lệnh: `cat hoa.txt qua.txt >> hoaqu.txt`
 - Hiển thị dòng đầu tiên của tệp *hoaqu.txt*
Gợi ý: dùng lệnh `head` kèm tùy chọn `-n`, cụ thể: `head -1 hoaqu.txt`
 - Hiển thị 3 dòng cuối của tệp *hoaqu.txt*
Gợi ý: dùng lệnh `tail` kèm tùy chọn `-n`, cụ thể: `tail -3 hoaqu.txt`

Bài 3: Với tệp *hoaqu.txt* (ở bài 2), thực hiện lệnh đáp ứng yêu cầu sau:

- o Đếm xem có bao nhiêu loài hoa?
- o Đếm xem có bao nhiêu loại hoa quả bắt đầu bằng chữ hoa?
- o Sắp xếp nội dung tệp *hoaqu.txt* theo thứ tự chữ cái rồi lưu nội dung đã sắp xếp vào một tệp có tên *hoaqusx.txt*

Gợi ý: sử dụng lệnh *grep*.

Bài 4: Thực hiện các thao tác sau:

- Kiểm tra lại vị trí có đúng ở thư mục người dùng không? Nếu không trở về thư mục người dùng (lệnh: `cd ~`)
- Tạo một thư mục có tên *lab3* (lệnh: `mkdir lab3`)
- Đếm xem có bao nhiêu người dùng đang đăng nhập vào hệ thống, sau đó sắp xếp kết quả theo thứ tự giảm dần, lưu kết quả đó vào một tệp có tên *user.dat*
Gợi ý: sử dụng kỹ thuật đường ống để kết hợp 3 lệnh sau:
`$who | sort -r | cat > user.dat`
`// tham số -r nghĩa là sắp xếp ngược (reverse)`
- Quan sát hiện tại quyền của từng đối tượng người dùng (*owner*, *group*, *other*) trên tệp *user.dat* hiện nay như thế nào? (lệnh: `ls -l`)
- Bỏ đi quyền đọc (*read*) tệp *user.dat* đối với 2 đối tượng *owner*, *group*. Sau đó kiểm tra lại kết quả bằng cách thử xem nội dung tệp *user.dat* (`cat user.dat`)
Gợi ý: `chmod ug-r user.dat`
- Sử dụng phần mềm *vim* để tạo một tệp mới có tên *vanban*, soạn thảo nội dung như sau:
`Dai hoc FPT`
`Mon hoc OS&N`
`Lab3`
- Xem lại nội dung tệp *vanban* vừa soạn bằng 2 cách: dùng lệnh *cat* và dùng chính phần mềm *vim*.

Bài 5: Tạo ra các tệp rỗng: *test.c*, *test.cpp*, *test.java* rồi thực hiện yêu cầu sau:

Liệt kê các tệp ngôn ngữ C bằng cách dùng ký hiệu đại diện “?”

Liệt kê các tệp ngôn ngữ C và C++ bằng cách dùng ký hiệu đại diện “?”

Liệt kê các tệp ngôn ngữ Java bằng cách dùng ký hiệu đại diện “.”

Bài 6 [tùy chọn]: Cài đặt các phiên bản Linux thông dụng như Fedora hoặc Ubuntu lên máy tính của bạn, vẫn giữ nguyên hệ điều hành (Windows) đã cài trước đó.

@#\$ **THE END** \$#@