

BÀI 6

SHELL VÀ CÁC LỆNH SHELL CƠ BẢN

6.1 Shell – Bộ thông dịch lệnh

6.1.1 Shell là gì?

Shell môi trường hoặc giao diện dòng lệnh (command-line interface) được sử dụng để tương tác với hệ thống máy tính. Nó cung cấp một cách để nhập lệnh và thực thi chúng trên hệ thống. Hệ điều hành MS-DOS và Windows thì có môi trường đánh lệnh command-line, hệ điều hành Unix và Linux có môi trường Shell. Chú ý rằng Shell không là một thành của hệ điều hành mà nó sử dụng hệ điều hành để thực thi lệnh, thao tác file...

Hệ điều hành Linux có thể có nhiều loại Shell khác nhau, dưới đây là một vài Shell thông dụng:

Tên Shell	Được phát triển bởi	Ghi chú
BASH (Bourne-Again SHell)	Brian Fox and Chet Ramey Free Software Foundation	Là Shell thông dụng nhất trong HĐH Linux
CSH (C SHell)	Bill Joy University of California (For BSD)	Là Shell có cú pháp rất giống ngôn ngữ C
KSH (Korn SHell)	David Korn AT & T Bell Labs	

Một vài lệnh liên quan đến shell

Liệt kê tất cả các loại shell có thể có trong hệ thống hiện tại dùng lệnh:

```
$cat /etc/shells
```

Kiểm tra loại Shell đang được sử dụng:

```
$echo $SHELL
```

6.1.2 Khởi động Shell

Nếu chúng ta đăng nhập vào hệ điều hành Linux dưới dạng Text thì mặc nhiên chúng ta sẽ phải trực tiếp làm việc với Shell. Trường hợp dùng Linux dưới dạng đồ họa (graphic) thì Shell sẽ được khởi động khi ta khởi động cửa sổ terminal (trong Fedora vào Applications / Accessories / Terminal). Khi Shell được khởi động người dùng sẽ được đón chào và bắt đầu nhập lệnh từ dấu nhắc (prompt) như sau:

```
$
```

Tuy nhiên dạng thức của dấu nhắc có thể thay đổi tùy hệ thống. Nó có thể bao gồm cả tên của máy tính (hostname), thư mục hiện hành đang làm việc của bạn (current working directory).

6.1.3 Shell script là gì?

Như đã nói ở trên shell là nơi cho phép người dùng nhập lệnh (thông thường từ bàn phím) và thực thi lệnh. Nhưng thay vì người dùng nhập tuần tự các câu lệnh và thực thi chúng một cách tuần tự thì người dùng có thể lưu các lệnh này vào một file text và yêu cầu shell thực hiện file này. Điều này được gọi là shell script.

6.2 Các lệnh cơ bản của Linux

6.2.1 Lệnh chuyển thư mục cd và đường dẫn trong linux

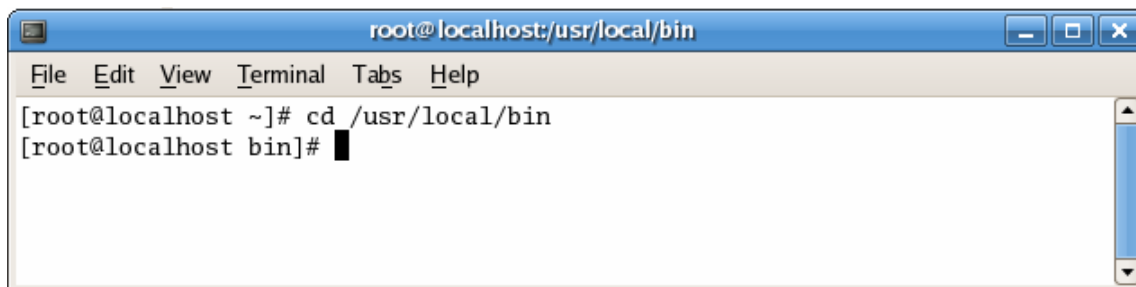
Mục đích: Lệnh cd cho phép bạn di chuyển đến những điểm khác nhau trên cây thư mục.

Cú pháp tổng quát:

\$cd path

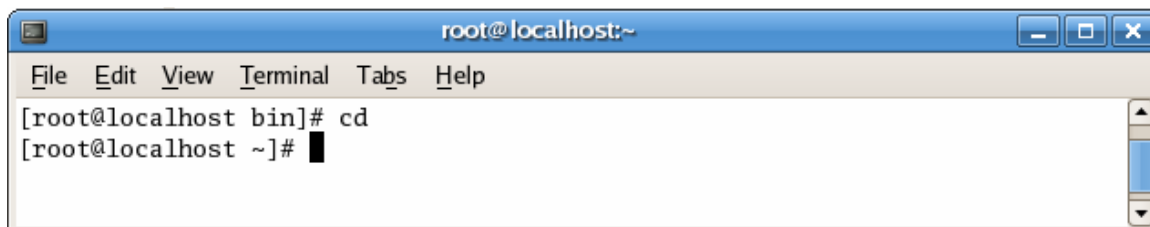
Trong đó path là đường dẫn chỉ đến thư mục mà ta muốn chuyển đến. Ví dụ:

Chuyển đến thư mục /usr/local/bin thực hiện như sau:



```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# cd /usr/local/bin  
[root@localhost bin]#
```

Chuyển về thư mục của người dùng hiện tại ta đánh lệnh: cd

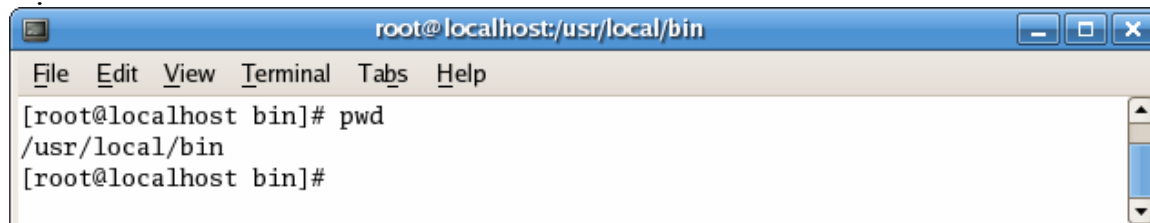


```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost bin]# cd  
[root@localhost ~]#
```

6.2.2 Đường dẫn (Path) trong hệ điều hành Linux

Dùng để định vị một tài nguyên nào đó (thông thường là file) trong hệ thống cây thư mục. Để xác định xem bạn đang ở vị trí nào (hay còn gọi là thư mục hiện hành) trong cây thư mục sử dụng lệnh: \$pwd.

Ví dụ:



```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost bin]# pwd  
/usr/local/bin  
[root@localhost bin]#
```

Có 2 loại đường dẫn thường được sử dụng là: Đường dẫn tuyệt đối và đường dẫn tương đối.

Đường dẫn tuyệt đối

Xác định đầy đủ toàn bộ đường dẫn từ thư mục gốc (ký hiệu: /) đến nơi cần định vị.

Ví dụ:

/dev

/usr/bin

/usr/local/bin

Để chuyển vào thư mục /usr/local/bin dùng lệnh như sau:

```
$cd /usr/local/bin
```

Đường dẫn tương đối

Đường dẫn tương đối là đường dẫn được tính từ vị trí hiện hành đến vị trí cần định vị. Một đường dẫn tương đối không bao giờ bắt đầu với ký tự '/

Ví dụ: giả sử chúng ta đang ở thư mục hiện hành là /usr. Khi đó để chuyển đến thư mục /usr/local/bin chúng ta sử dụng lệnh:

```
$cd local/bin
```

(Trong khi nếu sử dụng đường dẫn tuyệt đối thì phải đánh lệnh: \$cd /usr/local/bin)

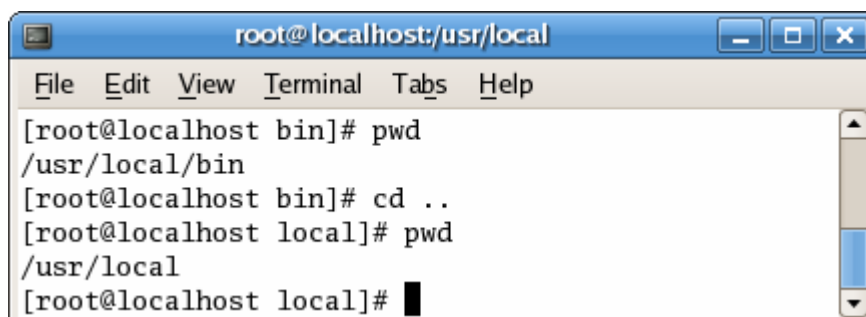
6.2.3 Thư mục cha “..”

Thư mục cha của thư mục hiện hành được ký hiệu bằng chuỗi hai chấm: “..” Ký hiệu này thường được sử dụng với lệnh cd để chuyển về thư mục cha của thư mục hiện hành:

Ví dụ: Giả sử chúng ta đang ở thư mục hiện hành là /usr/local/bin và thực hiện lệnh

```
$cd ..
```

Khi đó chúng ta sẽ được di chuyển về thư mục /usr/local (thử kiểm tra lại bằng lệnh \$pwd).

A screenshot of a terminal window titled 'root@localhost:/usr/local'. The window has a menu bar with 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help'. The terminal shows the following commands and output:

```
[root@localhost bin]# pwd
/usr/local/bin
[root@localhost bin]# cd ..
[root@localhost local]# pwd
/usr/local
[root@localhost local]#
```

Chúng ta hoàn toàn có thể sử dụng ký hiệu “..” trong đường dẫn tương đối. Giả sử có 2 đường dẫn /usr/local/bin và /usr/local/share và chúng ta đang có thư mục hiện hành là /usr/local/bin. Để di chuyển qua thư mục /usr/local/share ta dùng lệnh:

```
$cd ../share
```

Lệnh này có nghĩa là yêu cầu hệ điều hành di chuyển về thư mục cha của bin (là /usr/local) và chuyển tiếp đến thư mục share.

6.2.4 Thư mục hiện hành “.”

Ký hiệu dấu chấm “.” được dùng để thể hiện thư mục hiện hành và được sử dụng để yêu cầu shell thực hiện một chương trình hay shell script nào đó trong thư mục hiện hành.

Ví dụ: \$./myprog

myprog là một shell script trong thư mục hiện hành.

6.2.5 Thư mục cá nhân “~”

Trong hệ điều hành Linux mỗi người dùng khi đăng nhập vào đều có một thư mục cá nhân để lưu trữ dữ liệu riêng. Khi muốn trở về thư mục cá nhân người dùng có thể dùng lệnh **cd** không tham số. Cho dù đang ở bất kỳ thư mục nào trong cây thư mục.

Ngoài ra ký hiệu “~” cũng được dùng để xác định là thư mục cá nhân. Ví dụ ta thực hiện chương trình myprog trong thư mục hiện hành, với tham số là myfile.txt ở thư mục cá nhân

nhập bằng lệnh sau:

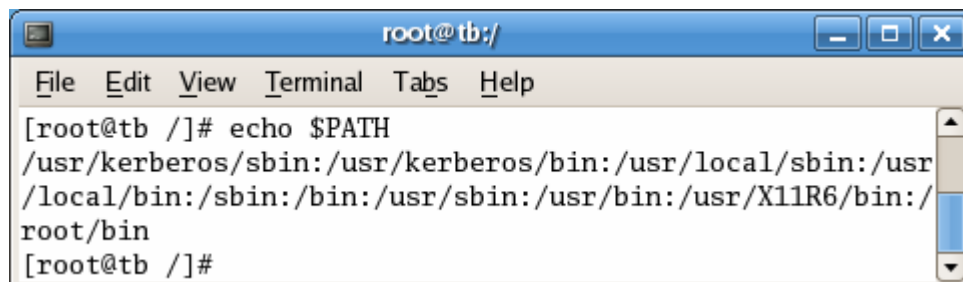
```
$ ./myprog ~/myfile.txt
```

6.2.6 Biến môi trường PATH

Biến PATH là một biến trong môi trường shell. PATH lưu danh sách các đường dẫn mà shell dựa vào đó để tìm kiếm một chương trình hay file nào đó khi cần thiết sau khi tìm kiếm ở thư mục hiện hành không có. Để kiểm tra biến PATH đang lưu những đường dẫn nào ta dùng lệnh:

```
$echo $PATH
```

Chú ý: PATH phải là ký tự hoa.

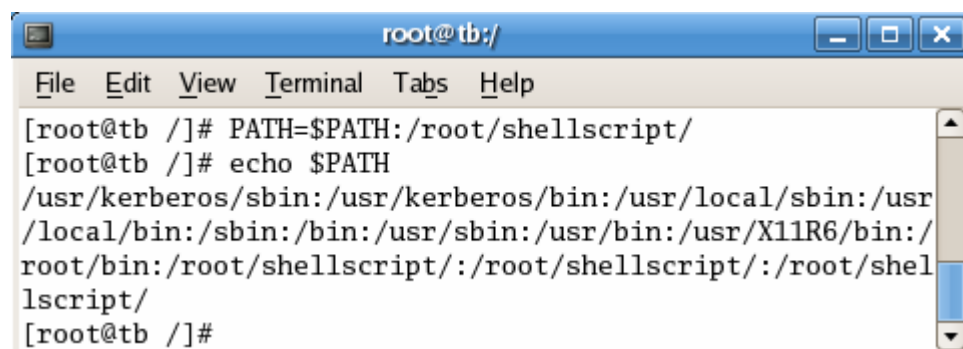


```
root@tb:/  
File Edit View Terminal Tabs Help  
[root@tb /]# echo $PATH  
/usr/kerberos/sbin:/usr/kerberos/bin:/usr/local/sbin:/usr  
/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/X11R6/bin:/  
root/bin  
[root@tb /]#
```

Để thêm đường dẫn mới vào danh sách của biến PATH ta có thể gắn đường dẫn cần thêm vào sau biến PATH như sau:

```
PATH=$PATH:<path>
```

Ví dụ: (chú ý PATH=\$PATH phải viết liền nhau)



```
root@tb:/  
File Edit View Terminal Tabs Help  
[root@tb /]# PATH=$PATH:/root/shellscript/  
[root@tb /]# echo $PATH  
/usr/kerberos/sbin:/usr/kerberos/bin:/usr/local/sbin:/usr  
/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/X11R6/bin:/  
root/bin:/root/shellscript:/root/shellscript:/root/shel  
lscript/  
[root@tb /]#
```

Đường dẫn /root/shellscript/ đã được đưa vào biến PATH.

6.3 Các lệnh liên quan đến việc khảo sát thư mục và file: ls, less và file

6.3.1 Lệnh ls (list files và thư mục)

Lệnh ls được dùng để liệt kê nội dung của thư mục. Đây là lệnh được dùng thường nhất trong Linux. Lệnh ls được sử dụng rất nhiều cách với các tham số tương ứng. Dưới đây là một vài trường hợp sử dụng của lệnh ls.

\$ls: liệt kê file trong thư mục hiện hành. Ví dụ:

```

root@localhost:~
File Edit View Terminal Tabs Help
[root@localhost ~]# ls
anaconda-ks.cfg Desktop install.log install.log.syslog
[root@localhost ~]#

```

\$ls -l: liệt kê files trong thư mục hiện hành với thông tin đầy đủ (long format). Ví dụ:

```

root@localhost:~
File Edit View Terminal Tabs Help
[root@localhost ~]# ls -l
total 60
-rw----- 1 root root 1187 Oct 10 22:25 anaconda-ks.cfg
drwxr-xr-x 2 root root 4096 Oct 10 22:29 Desktop
-rw-r--r-- 1 root root 28320 Oct 10 22:24 install.log
-rw-r--r-- 1 root root 4152 Oct 10 22:21 install.log.syslog
[root@localhost ~]#

```

Kết quả của lệnh ls với tham số -l được chia thành 7 cột:

-rw-----	1	root	root	1187	Oct 10 22:25	anaconda-ks.cfg
drwxr-xr-x	2	root	root	4096	Oct 10 22:29	Desktop
-rw-r--r--	1	root	root	28320	Oct 10 22:24	install.log
-rw-r--r--	1	root	root	4152	Oct 10 22:21	install.log.syslog
(1)	(2)	(3)	(4)	(5)	(6)	(7)

Cột (1): xác định quyết truy xuất của file hoặc thư mục. Ký hiệu “-” thể hiện đó là một file bình thường, ký tự “d” xác định đó là thư mục. Tập hợp các ký tự theo sau được chia làm 3 nhóm, mỗi nhóm 3 ký tự. Các ký tự này có thể là r (read), w (write), x (execute). Nhóm 3 ký tự đầu thể hiện quyền của người sở hữu (owner), nhóm 3 ký tự kế thể hiện quyền cho nhóm (group) và nhóm 3 ký tự cuối thể hiện quyền cho tất cả những người dùng khác.

Cột (2): thể hiện số lượng nối kết kay liên kết (link) đến tập tin hoặc thư mục (kháiniệm nối kết sẽ được đề cập sau)

Cột (3): xác định chủ sở hữu của tập tin hoặc thư mục

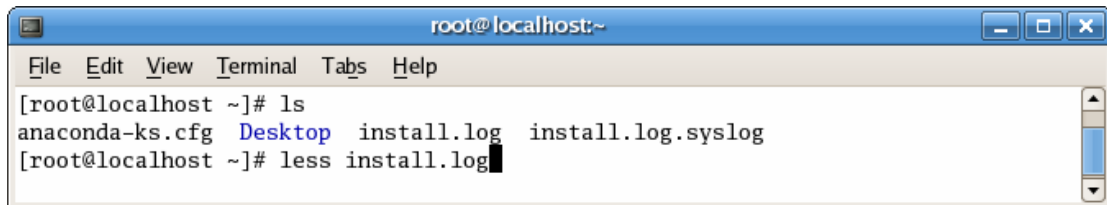
Cột (4): tên của nhóm có quyền tương ứng trên tập tin hoặc thư mụcCột (5): kích thước tập tin hoặc thư mục tính theo bytes

Cột (6): thời gian mà file hay thư mục được cập nhật.Cột (7): tên file hoặc thư mục.

6.3.2 Lệnh less

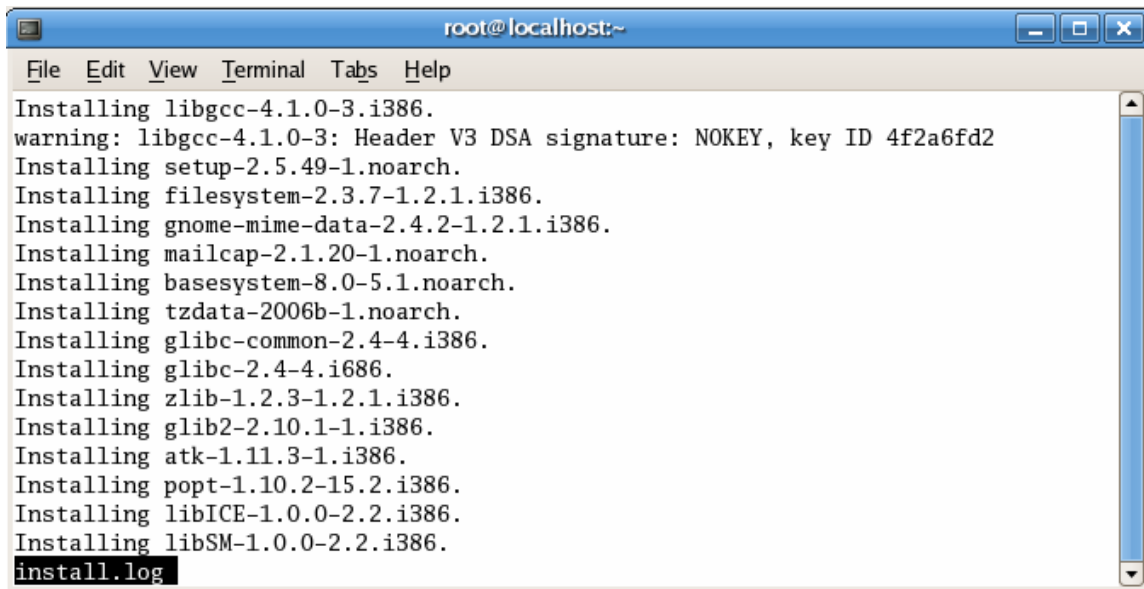
Lệnh less cho phép xem nội dung của file text. Cú pháp: \$ less tên_file

Ví dụ:



```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# ls  
anaconda-ks.cfg Desktop install.log install.log.syslog  
[root@localhost ~]# less install.log
```

Kết quả thực hiện lệnh trên như sau:



```
root@localhost:~  
File Edit View Terminal Tabs Help  
Installing libgcc-4.1.0-3.i386.  
warning: libgcc-4.1.0-3: Header V3 DSA signature: NOKEY, key ID 4f2a6fd2  
Installing setup-2.5.49-1.noarch.  
Installing filesystem-2.3.7-1.2.1.i386.  
Installing gnome-mime-data-2.4.2-1.2.1.i386.  
Installing mailcap-2.1.20-1.noarch.  
Installing basesystem-8.0-5.1.noarch.  
Installing tzdata-2006b-1.noarch.  
Installing glibc-common-2.4-4.i386.  
Installing glibc-2.4-4.i686.  
Installing zlib-1.2.3-1.2.1.i386.  
Installing glib2-2.10.1-1.i386.  
Installing atk-1.11.3-1.i386.  
Installing popt-1.10.2-15.2.i386.  
Installing libICE-1.0.0-2.2.i386.  
Installing libSM-1.0.0-2.2.i386.  
install.log
```

Nội dung của file được liệt kê theo từng trang màn hình. Để duyệt qua nội dung của file ta dùng các phím chức năng như sau:

Page Up hoặc b: cuộn lên 1 trang

Page Down hoặc space: cuộn xuống 1 trang

G: di chuyển đến dòng cuối của file

1G: di chuyển đến dòng đầu của file

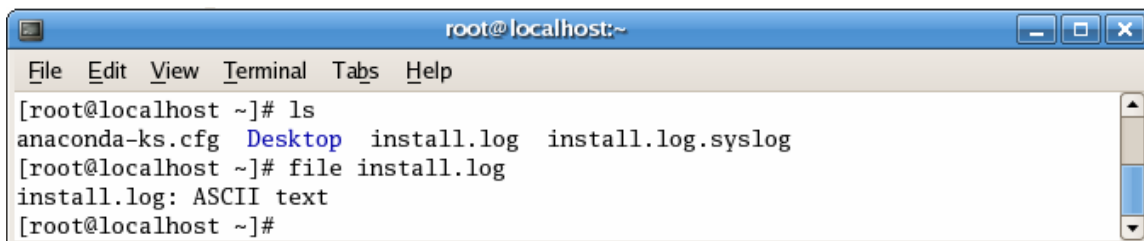
/chuỗi: nhảy đến dòng chứa chuỗi cần tìm: tiếp tục tìm chuỗi đã tìm trước đó

q: thoát

6.3.3 Lệnh file

Lệnh file được dùng để xác định file thuộc loại file gì, chẳng hạn như file: ASCII text, Bourne-Again shell script text, JPEG image data, RPM...

Cú pháp: \$file tên_file



```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# ls  
anaconda-ks.cfg Desktop install.log install.log.syslog  
[root@localhost ~]# file install.log  
install.log: ASCII text  
[root@localhost ~]#
```

6.4 Các lệnh thao tác file

6.4.1 Ký tự đại diện (wildcard)

Ký tự đại diện được dùng để xác định tên file hay tập hợp các tên phải theo một khuôn mẫu (pattern) nào đó. Dưới đây liệt kê các ký tự đại diện và ý nghĩa của nó:

Ký tự đại diện	Ý nghĩa
*	Trùng khớp với bất kỳ một hay nhiều ký tự nào
?	Trùng khớp với bất kỳ ký tự đơn nào
[tập_hợp_các_ký_tự]	Trùng khớp với bất kỳ ký tự nào nằm trong tập_hợp_các_ký_tự. Tập_hợp_các_ký_tự có thể được thể hiện dưới dạng 1 khoản. Ví dụ [A-Z] thể hiện tập hợp các ký tự hoa.
[! tập_hợp_các_ký_tự]	Trùng khớp với bất kỳ ký tự nào KHÔNG nằm trong tập_hợp_các_ký_tự

Dưới đây là một vài ví dụ minh họa việc sử dụng ký tự đại diện.

Ví dụ	Giải thích
*	Tất cả tên file
g*	Tất cả các file có tên bắt đầu bằng chữ g
b*.txt	Tất cả các file có tên bắt đầu bằng chữ b và kết thúc với .txt
Data???	Tất cả các file có tên bắt đầu với Data và tiếp theo là chính xác 3 ký tự.
[abc]*	Tất cả các file có tên bắt đầu bằng a hoặc b hoặc c theo sau là bất kỳ ký tự gì
[A-Z]*	Tất cả các file có tên bắt đầu bằng ký tự hoa theo sau là bất kỳ ký tự gì
BACKUP.[0-9][0-9][0-9]	Tất cả các file có tên bắt đầu với BACKUP kế đến là dấu chấm và tiếp theo là 3 ký số
[!a-z]*	Tất cả các file có tên bắt đầu KHÔNG phải là ký tự thường theo sau là bất kỳ ký tự gì

6.4.2 Lệnh cp (copy)

Lệnh cp thực hiện copy file hoặc thư mục. Dưới đây là các dạng cơ bản của lệnh cp.

Cú pháp	Giải thích
cp file1 file2	Copy nội dung của file1 vào file2. Nếu file2 chưa tồn tại thì sẽ được tạo ra, ngược lại nội dung của file2 sẽ bị chép chồng.
cp -i file1 file2	Giống như trên, nhưng trong trường hợp file2 đã tồn tại thì người dùng sẽ được nhắc nhở trước khi file2 bị chép chồng.
cp file1 dir1	Copy file1 vào trong thư mục dir1
cp -R dir1 dir2	Nếu dir2 chưa tồn tại thì dir2 được tạo ra và nội dung thư mục 1 được chép vào. Trường hợp dir2 đã tồn tại thì thư mục dir1 được tạo ra trong dir2 và nội dung của thư mục dir1 được chép vào dir1 trong dir2.

6.4.3 Lệnh mv (move)

Lệnh mv dùng để di chuyển một hoặc nhiều files vào một thư mục khác, hoặc lệnhmv cũng có thể được dùng để đổi tên file. Dưới đây là một số cách dùng của lệnh mv.

Cú pháp	Giải thích
mv file1 file2	Nếu file2 không tồn tại thì file1 đổi tên thành file2. Trường hợp file2 đã tồn tại thì nội dung của file2 bị chép chồng và file1 sẽ bị xóa.
mv file1 file2 file3 dir	Di chuyển file1, file2, file3 vào thư mục dir nếu thư mục dir tồn tại, nếu không lệnh mv sẽ báo lỗi.
mv dir1 dir2	Nếu dir2 không tồn tại thì thư mục dir1 đổi tên thành dir2. Trường hợp dir2 đã tồn tại thì dir1 sẽ được di chuyển vào dir2.

6.4.4 Lệnh rm (remove)

Được dùng để xóa file hoặc thư mục. Dưới đây là vài cách dùng của lệnh rm.

Cú pháp	Giải thích
Rm file1 file2	Xóa file1 và file2
Rv -r dir1 dir2	Xóa thư mục dir1 và dir2 cùng với toàn bộ nội dung của nó

6.4.5 Lệnh mkdir (make directory):

Được dùng để tạo thư mục. Cú pháp: \$mkdir tên_thư_mục

6.4.6 Lệnh "which"

Bạn có thể kiểm tra một chương trình nào đó có nằm trong các thư mục được chỉ ra bởi PATH hay không bằng cách dùng lệnh which. Ví dụ, ở đây ta sẽ thấy rằngtrong hệ thống Linux hiện tại không có chương trình tên sense:

```
$ which sense
```

```
which: no sense in (/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/usr/X11R6/bin) Trường
```


hợp khác ta lại thành công khi tìm lệnh ls:

```
$ which ls
```

```
/usr/bin/ls
```

6.4.7 Sử dụng "which -a"

Dùng cờ -a để yêu cầu which hiển thị tất cả các chương trình được yêu cầu có trong các thư mục được mô tả bởi PATH:

```
$ which -a ls
```

```
/usr/bin/ls
```

```
/bin/ls
```

6.4.8 Lệnh whereis

Nếu bạn muốn tìm nhiều thông tin hơn ngoài vị trí của một chương trình bạn có thể dùng chương trình whereis :

```
$ whereis ls
```

```
ls: /bin/ls /usr/bin/ls /usr/share/man/man1/ls.1.gz
```

Ta thấy rằng that ls có mặt trong hai nơi là /bin và /usr/bin. Hơn thế chúng ta còn được thông báo rằng có tài liệu hướng dẫn sử dụng nằm trong thư mục

```
/usr/share/man.
```

6.4.9 Lệnh find

Lệnh find là một tiện ích khác cho phép bạn tìm kiếm các tập tin. Với find bạn không bị giới hạn trong phạm vi tìm kiếm chương trình, bạn có thể tìm kiếm một tập tin bất kỳ mà bạn muốn bằng cách sử dụng một tiêu chuẩn tìm kiếm nào đó. Ví dụ, để tìm một tập tin có tên README, bắt đầu từ thư mục /usr/share/doc ta thực hiện lệnh sau:

```
$ find /usr/share/doc -name README
```

```
/usr/share/doc/ion-20010523/README
```

```
/usr/share/doc/bind-9.1.3-r6/dhcp-dynamic-dns-examples/README
```

```
/usr/share/doc/sane-1.0.5/README
```

6.4.9.1 Sử dụng các ký tự đại diện với lệnh find

Bạn có thể sử dụng các ký tự đại diện trong tham số -name, mà bạn bao bọc nó bằng cặp nháy đơn hay đặt ký tự \ phía trước ký tự đại diện đó. Ví dụ , chúng ta muốn tìm tập tin README với các phần mở rộng khác nhau như sau:

```
$ find /usr/share/doc -name README\*
```

```
/usr/share/doc/iproute2-2.4.7/README.gz
```

```
/usr/share/doc/iproute2-2.4.7/README.iproute2+tc.gz
```

```
/usr/share/doc/iproute2-2.4.7/README.decnet.gz
```

```
/usr/share/doc/iproute2-2.4.7/examples/diffserv/README.gz
```

```
/usr/share/doc/pilot-link-0.9.6-r2/README.gz
```

```
/usr/share/doc/gnome-pilot-conduits-0.8/README.gz
```

```
/usr/share/doc/gimp-1.2.2/README.i18n.gz
```

```
/usr/share/doc/gimp-1.2.2/README.win32.gz
```

```
/usr/share/doc/gimp-1.2.2/README.gz
```

```
/usr/share/doc/gimp-1.2.2/README.perl.gz[578 additional lines snipped]
```

6.4.9.2 Tìm không phân biệt chữ hoa và chữ thường với find

Ta có thể dùng lệnh sau:

```
$ find /usr/share/doc -name '[Rr][Ee][Aa][Dd][Mm][Ee]*' Hay dùng tham số -iname:
```

```
$ find /usr/share/doc -iname readme\*
```

6.4.9.3 Tùy chọn type của find

Tùy chọn -type cho phép bạn tìm kiếm các đối tượng trong hệ thống tập tin theo những kiểu khác nhau. Các tham số có thể của tùy chọn -type là b (cho thiết bị dạng khối), c (thiết bị dạng ký tự), d (thư mục), p (ống dẫn có tên), f (tập tin thường), l (liên kết mềm), và s (socket). Ví dụ, để tìm kiếm các liên kết mềm trong thư mục /usr/bin mà nó có chứa chuỗi vim ta thực hiện lệnh sau:

```
$ find /usr/bin -name '*vim*' -type l
```

```
/usr/bin/rvim
```

```
/usr/bin/vimdiff
```

```
/usr/bin/gvimdiff
```

6.9.4 find và tùy chọn mtime

Tùy chọn -mtime cho phép bạn chọn các tập tin dựa trên thời gian cập nhật saucùng của nó. Tham số của mtime là những khoảng 24 giờ, và có thể thêm dấu cộng (có nghĩa là sau) hoặc dấu trừ (có nghĩa là trước), thời điểm mốc được tính tại thời điểm cách thời điểm hiện tại 24 giờ. Ví dụ, Xem xét trường hợp sau:

```
$ ls -l ?
```

```
-rw - - - - - 1 root root 0 Jan 7 18:00 a
```

```
-rw - - - - - 1 root root 0 Jan 6 18:00 b
```

```
-rw - - - - - 1 root root 0 Jan 5 18:00 c
```

```
-rw - - - - - 1 root root 0 Jan 4 18:00 d
```

```
$ date
```

```
Mon Jan 7 18:14:52 EST 2002
```

Bạn có thể tìm các tập tin mà nó được tạo ra trong vòng 24 giờ vừa qua bằng lệnh sau:

```
$ find . -name \? -mtime -1
```

```
./a
```

Hoặc bạn có thể tìm các tập tin mà chúng được tạo ra trước cách đây đã hơn 24 giờ:

```
$ find . -name \? -mtime +0
```

```
./b
```

```
./c
```

```
./d
```

6.9.5 Tùy chọn -size

Tùy chọn -size cho phép bạn tìm các tập tin dựa trên kích thước của chúng. Mặc định, đối số của -size là các khối 512-byte, tuy nhiên việc thêm vào các hậu tố làm mọi việc dễ dàng hơn. Các hậu tố được chấp nhận là b (khối 512-byte), c (bytes), k (kilobytes), and w (2-byte). Bên cạnh đó, bạn có thể thêm vào dấu cộng để nói rằng lớn hơn hoặc dấu trừ để nói là nhỏ hơn.

Ví dụ, để tìm các tập tin bình thường mà nó nhỏ hơn 50 bytes ta dùng lệnh sau:

```
$ find /usr/bin -type f -size -50c
```

```
/usr/bin/krdb
```

```
/usr/bin/run-nautilus
```

```
/usr/bin/sgmlwhich
```

```
/usr/bin/muttbug
```

6.5 Quyền truy xuất file (file permission)

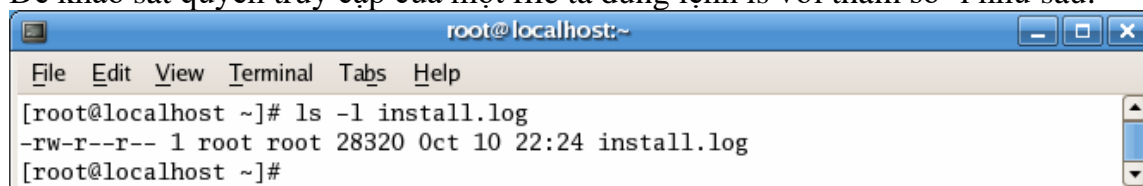
Với mục đích là xây dựng một hệ điều hành đa người dùng, ngay từ lúc đầu HĐH Unix (và Linux sau này) đã được thiết kế để đáp ứng yêu cầu này. Trong HĐH đa người dùng phải đảm bảo sao cho hoạt động của một người dùng không thể làm hỏng hệ thống hay phá hoại dữ liệu của người dùng khác. Để tìm hiểu rõ hơn HĐH Linux đã thực hiện việc này thế nào, trong phần này chúng ta tìm hiểu các lệnh sau:

chmodsu chownchgrp

Đây là các lệnh cơ bản nhất liên quan đến việc thiết lập các quyền truy xuất đến file. Tuy nhiên trước khi đề cập kỹ đến các lệnh trên. Chúng ta sẽ tìm hiểu một chút về các quyền truy xuất trên file.

Trên hệ thống Linux (hay Unix) file và thư mục được gán quyền truy cập cho 3 đối tượng: **người sở hữu file** (owner), **nhóm người dùng** (group) và **những người dùng khác** (everybody else). Các quyền có thể được gán cho 3 đối tượng này là: quyền **đọc** (read), quyền **viết** (write) và quyền **thực thi** (execute).

Để khảo sát quyền truy cập của một file ta dùng lệnh ls với tham số -l như sau:

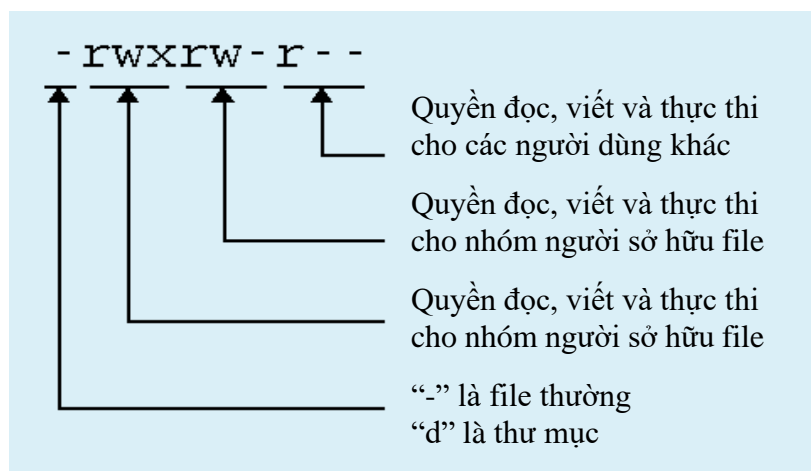


```
root@localhost:~  
File Edit View Terminal Tabs Help  
[root@localhost ~]# ls -l install.log  
-rw-r--r-- 1 root root 28320 Oct 10 22:24 install.log  
[root@localhost ~]#
```

Với lệnh ls -l install.log liệt kê các thông tin của file install.log:

- Chủ sở hữu (owner) của file là root.
- Người dùng root này có quyền read(r) và write(w).
- Những người dùng thuộc nhóm root chỉ có quyền read(r)
- Những người dùng khác cũng chỉ có quyền (r).

Quyền truy xuất của file được qui định bởi cột đầu tiên (vd: -rw-r--r--). Dưới đây là sơ đồ giải thích thông tin của cột đầu tiên này.



6.5.1 Lệnh chmod

Lệnh chmod được sử dụng để thay đổi quyền truy xuất file và thư mục. Khi sử dụng lệnh chmod phải cung cấp quyền cần thiết lập cho file hay thư mục. Quyền cần thiết lập được biểu diễn bởi một dãy số nhị phân như ví dụ sau:

rwX rwX rwX = 111 111 111

rw- rw- rw- = 110 110 110

rwX ----- = 111 000 000

...

Dãy các bit nhị phân này có thể được biểu diễn thành các số thập phân bởi từng nhóm 3bit. Dưới đây là dãy 3bit có thể có dùng để thiết lập quyền truy xuất cho file hoặc thư mục.

rwX = 111 = 7 (read, write và execute) rw- = 110 = 6 (read, write)

r-x = 101 = 5 (read, execute) r-- = 100 = 4 (read)

Để hiểu rõ hơn về việc thiết lập quyền truy xuất chúng ta hãy xem ví dụ sau. Giả sử người dùng user muốn thiết lập file test do mình làm chủ sở hữu chỉ một mình người dùng user này có quyền đọc và viết, tất cả người dùng còn lại không thể truy xuất đến file này.

Như vậy chủ sở hữu có quyền đọc viết thì nhóm 3bit đầu phải là rw- (110 hay 6), 2 nhóm 3bit còn lại là ----- (000 000 hay 00). Khi đó lệnh chmod được sử dụng như sau:



```
user@localhost:~  
File Edit View Terminal Tabs Help  
[user@localhost ~]$ chmod 600 test  
[user@localhost ~]$ ls -l test  
-rw----- 1 user user 8 Oct 12 00:50 test  
[user@localhost ~]$
```

Sau đây là bảng mô tả các thiết lập quyền truy xuất file hoặc thư mục thường sử dụng

Giá trị thiết lập	Mô tả
777 (rwX rwX rwX)	Mọi người đều có toàn quyền, thiết lập này thường rất ít được sử dụng
755 (rwX r-x r-x)	Chủ sở hữu có toàn quyền, những người còn lại chỉ có quyền đọc và thực thi. Thiết lập này thường được dùng cho chương trình mà mọi người có thể dùng.
700 (rwX -----)	Chủ sở hữu có thể đọc, viết và thực thi. Những người khác không có quyền gì cả. Hữu dụng cho những chương trình chỉ được sử dụng và giữ riêng cho chủ sở hữu.
666 (rw- rw- rw-)	Tất cả người dùng có quyền đọc và viết
644 (rw- r-- r--)	Chủ sở hữu có quyền đọc viết, những người khác chỉ có quyền đọc. Hữu dụng cho file dữ liệu, chỉ có chủ sở hữu có thể thay đổi, những người khác chỉ đọc.
600 (rw-----)	Chủ sở hữu có quyền đọc viết, những người khác không có quyền gì cả. Hữu dụng cho file dữ liệu quan trọng mà chủ sở hữu muốn giữ riêng cho mình.

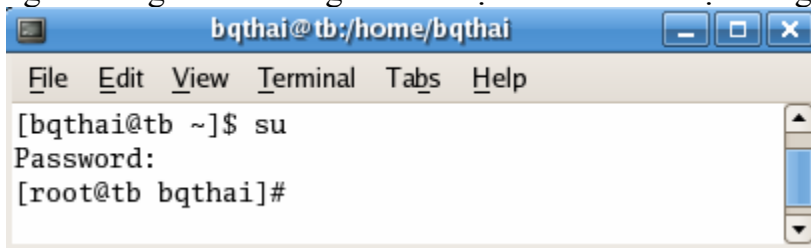
Quyền trên thư mục

Lệnh chmod cũng được dùng để gán quyền truy cập cho thư mục. Việc gán quyền trên thư mục cũng tương tự như trên file, tuy nhiên quyền execute (x) trên thư mục có nghĩa là cho phép liệt kê nội dung trong thư mục đó. Dưới đây là vài ví dụ về gán quyền cho thư mục:

Giá trị thiết lập	Mô tả
777 (rwx rwx rwx)	Mọi người đều có toàn quyền (liệt kê, tạo và xóa).
755 (rwx r-x r-x)	Chủ sở hữu có toàn quyền, những người còn lại chỉ có thể liệt kê nội dung thư mục, không có quyền tạo và xóa file trong thư mục.
700 (rwx -----)	Tất cả mọi người đều không thể truy xuất thư mục này, ngoại trừ người sở hữu có toàn quyền.

6.5.2 Lệnh su (superuser)

Trong một số trường hợp người dùng cần phải có những thao tác của người quản trị (hiển nhiên phải có quyền quản trị) để thao tác một số lệnh vượt quá quyền hạn của một người dùng bình thường. Khi đó lệnh su có thể được dùng trong trường hợp này.



Ví dụ trên cho thấy việc chuyển từ người dùng bqthai sang người dùng root (có quyền quản trị). Khi dùng lệnh su người dùng sẽ được nhắc nhở cung cấp password của người dùng quản trị.

Để trở lại người dùng cũ ta dùng lệnh **exit**.

6.5.3 Lệnh chown

Lệnh chown được dùng để thay đổi quyền sở hữu của một file hay thư mục. Cú pháp như sau:

chown người_sở_hữu_mới tên_file

Để có thể thực hiện lệnh chown phải có quyền root (dùng lệnh su). Ví dụ:



Các lệnh trên đã thay đổi quyền sở hữu của file test từ người dùng bqthai sangroot.

6.5.4 Lệnh chgrp

Lệnh chgrp được sử dụng để thay đổi quyền truy cập của nhóm đối với file và thư mục. Cú pháp và cơ chế hoạt động cũng giống như lệnh chown.

6.6 Bài Tập các mục 6.1, 6.2, 6.3, 6.4 và 6.5

Bài 1: Thực hiện và khảo sát lại tất cả các ví dụ trên

Bài 2: Thực hiện lệnh chuyển thư mục theo thứ tự sau:

- Chuyển về thư mục gốc /
- Chuyển đến thư mục /bin
- Chuyển đến thư mục /usr/local/bin
- Chuyển ra thư mục trên một mức (local)
- Chuyển về thư mục người dùng

Bài 3: Thực hiện tạo một script có tên **hello** đơn giản trong thư mục /usr như sau:

```
$cat >hello
```

```
$echo "Xin chào!!!"
```

```
$^D (bấm ctrl+D)
```

```
$chmod 755 hello
```

- Đứng tại thư mục thực hiện shell này dùng cú pháp: **sh hello**
- Thực hiện chuyển thư mục về thư mục cá nhân và thực hiện shell trên với cú pháp: **sh hello**

Shell hello có thực hiện được hay không? Vì sao?

Bài 4: Sử dụng đường dẫn tuyệt đối để thực hiện shell hello

Bài 5: Xem nội dung của biến PATH hiện tại của hệ thống. Thêm đường dẫn vào biến PATH để shell hello (câu 3) có thể được thực hiện theo cú pháp: **sh hello** cho dù đang đứng ở bất kỳ vị trí nào?

Bài 6: Thực hiện lệnh `ls -l` trong thư mục /bin. Xác định các quyền của người sở hữu, nhóm và người dùng khác trên file **ls** và **sh**?

Thực hiện lệnh `ls -l` trên thư mục gốc. Xác định các quyền và những người dùng hay nhóm người dùng tương ứng của thư mục root?

Bài 7: Thực hành lệnh file bằng cách tìm các file dạng ASCII và dùng lệnh less xem nội dung của các file đó.

Bài 8: Tạo cây thư mục như sau trong thư mục cá nhân của mình.

Myweb

|-images

| |-- icon

| |-- background

| |-- animation

|-databases

|-scripts

|-java

Dùng lệnh tree để xem lại cây thư mục vừa tạo.

Bài 9: Tìm một số file có phần mở rộng .html và .class trong hệ thống file và thực hiện copy vài file .html vào thư mục Myweb và vài file .class vào thư mục Myweb/scripts.

Bài 10: Ý nghĩa của lệnh chmod trong bài 3 là gì?

6.6 Command Prompt trên Windows 10

Command Prompt hay còn gọi là CMD, là một công cụ giúp người dùng có thể nhập các lệnh vào để mở ra một số tác vụ trên hệ điều hành Windows.

Thông thường, người dùng có thể mở các tác vụ bằng cách nhấn vào biểu tượng của tác vụ hoặc qua nhiều bước khác nhau. Nhưng với CMD, bạn chỉ cần nhập lệnh tương ứng với tác vụ thì việc mở tác vụ sẽ cực kỳ nhanh, giúp bạn tiết kiệm nhiều thời gian.

Ví dụ: Thay vì mở Control Panel như cách thông thường thì bạn có thể nhập lệnh "control" vào cửa sổ CMD > Nhấn Enter để mở Control Panel.

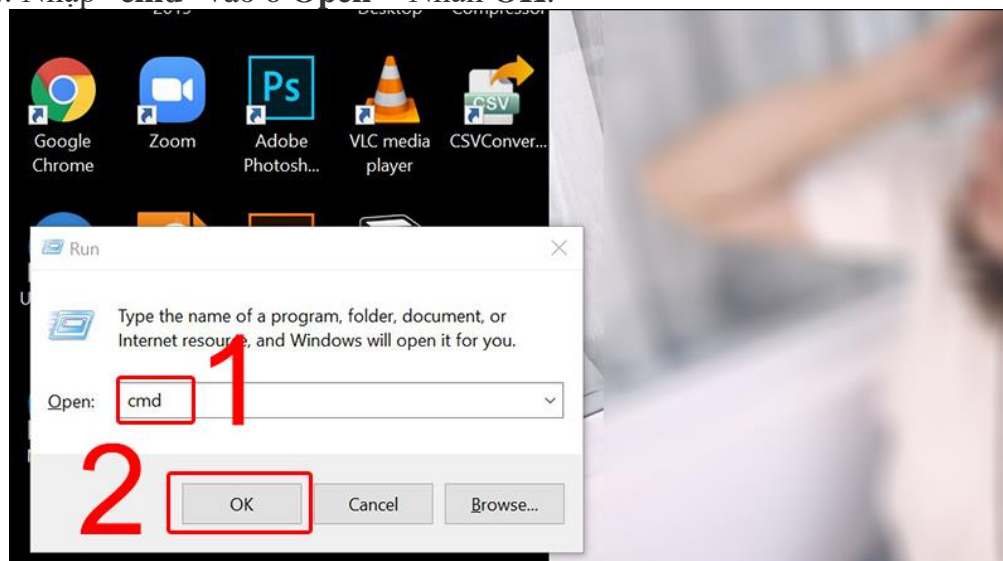
6.6.1 Cách mở Command Prompt trên Windows 10

Trên Windows 10 sẽ có 2 cách mở CMD, bạn có thể lựa chọn 1 cách thích hợp để mở.

1. Sử dụng hộp thoại Run

Bước 1: Nhấn tổ hợp phím **Windows + R** để mở hộp thoại Run.

Bước 2: Nhập "cmd" vào ô **Open** > Nhấn **OK**.

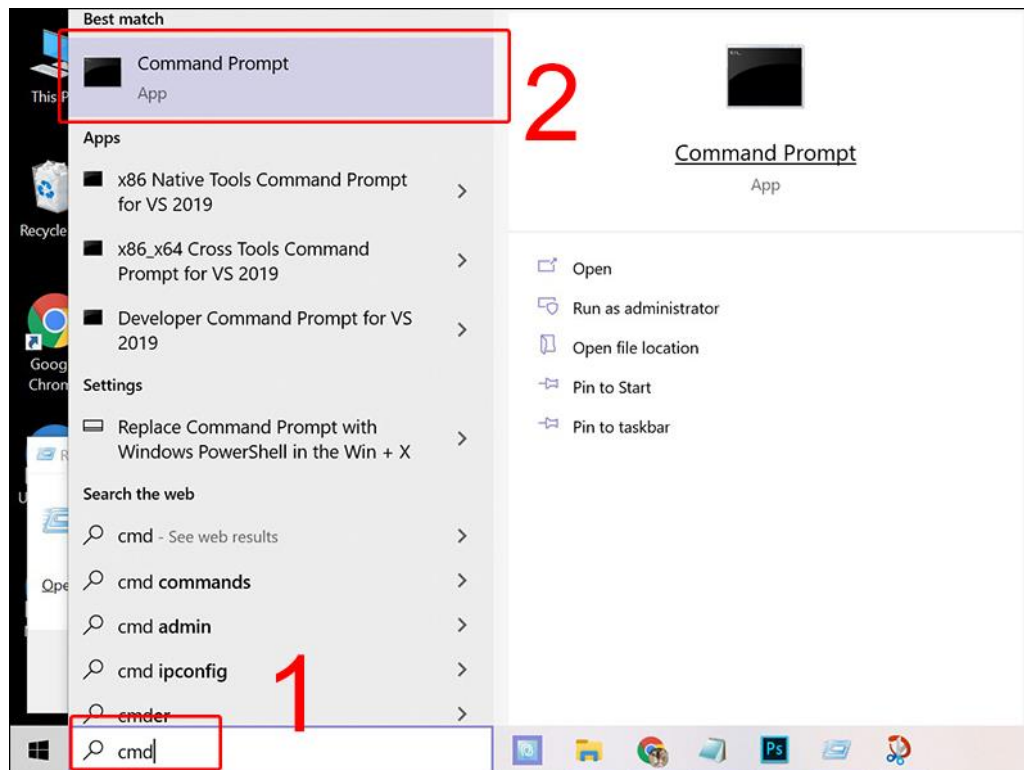


Mở Command Prompt bằng hộp thoại Run

2. Sử dụng nút Start

Bước 1: Nhấn **biểu tượng Windows** ở góc trái màn hình hoặc **nút Windows** trên bàn phím.

Bước 2: Nhập "cmd" trên thanh tìm kiếm > Chọn phần mềm **Command Prompt**.



6.6.2 Các lệnh CMD thông dụng

1. Lệnh PING CMD

Công dụng: Sử dụng lệnh Ping để kiểm tra xem một [máy tính](#) có kết nối mạng không. Lệnh PING gửi các gói tin từ máy tính bạn tới máy tính đích, bạn có thể xác định được tình trạng đường truyền hoặc xác định máy tính đó có kết nối hay không.

Cú pháp: ping ip/host/[t]/[a]/[l]/[n].

Trong đó:

- ip: Địa chỉ IP của máy cần kiểm tra.
- /host: Tên của máy tính cần kiểm tra kết nối mạng (có thể sử dụng địa chỉ IP hoặc tên của máy tính).
- /t: Sử dụng để máy tính liên tục “ping” đến máy tính đích, bấm Ctrl +C để dừng.
- /a: Nhận địa chỉ IP từ tên máy tính (host).
- /l: Xác định độ rộng của gói tin gửi đi kiểm tra.
- /n: Xác định số gói tin gửi đi.

2. Lệnh Tracert

Công dụng: Lệnh giúp bạn thấy đường đi của các gói tin từ máy tính các bạn đến máy tính đích, xem các gói tin đi qua những server hay [router](#) nào...

Cú pháp: tracert ip/host.

Trong đó:

- ip/host: Địa chỉ ip/ tên máy tính.

Lệnh Netstat

Công dụng: Liệt kê các kết nối ra vào máy tính của các bạn.

Cú pháp: Netstat [/a]/[e]/[n].

Trong đó:

- /a: Hiển thị tất cả kết nối và các cổng đang lắng nghe.
- /e: Thông tin thông kê Ethernet.
- /n: Hiển thị các địa chỉ và các số cổng kết nối.

3. Lệnh Ipconfig

Công dụng: Hiển thị cấu hình IP của máy tính các bạn đang sử dụng (tên host, địa chỉ IP, DNS...).

Cú pháp: ipconfig /all.

Lệnh Shutdown

Công dụng: Tắt và khởi động lại máy tính.

Cú pháp (Windows 7):

- Tắt máy: Shutdown -s -t [a].
- Restart máy tính: Shutdown -r -t [a].

Trong đó:

- a: thời gian tắt máy (đơn vị giây).

4. Lệnh Dir

Công dụng: Xem file, folder.

Cú pháp: DIR [drive:] [path][filename].

Trong đó:

- Path: Đường dẫn tới file, folder.
- Filename: Tên file.

Lệnh Del

Công dụng: Xóa file.

Cú pháp: DEL [/p][/f][/s][/q][/a[:attributes]] “tên file cần xóa” CMD.

Trong đó:

- /p: Hiển thị thông tin file trước khi xóa.
- /f: Xóa các file có thuộc tính chỉ đọc (read-only).
- /s: Xóa file đó trong tất cả các thư mục có chứa.
- /q: Xóa không cần hỏi.
- /a[:attributes]: Xóa theo thuộc tính của file (R: Read-only files, S: System files, H: Hidden files).

5. Lệnh Copy

Công dụng: Copy file từ thư mục này sang thư mục khác trong máy tính.

Cú pháp: COPY “địa chỉ cần copy” “địa chỉ lưu file copy” /y.

Trong đó:

- /y: Copy không cần hỏi.

Lệnh RD

Công dụng: Lệnh RD giúp các bạn xóa thư mục.

Cú pháp: RD /s /q “thư mục cần xóa”.

Trong đó:

- /s: Xóa toàn bộ thư mục.
- /q: Xóa không cần hỏi.

7. Lệnh MD

Công dụng: Tạo thư mục mới.

Cú pháp: MD “đường dẫn lưu file cần tạo”\”tên thư mục cần tạo”.

8. Lệnh Taskkill

Công dụng: Tắt một ứng dụng đang chạy.

Cú pháp: taskkill /f /im “tên ứng dụng”.exe.

Lệnh Reg

Công dụng: Tạo, chỉnh sửa Registry.

Cú pháp: REG ADD KeyName [/v ValueName] [/t Type] [/s Separator] [/d Data] [/f].

Trong đó:

- KeyName: Đường dẫn tới Key.
- /v ValueName: Tên value cần tạo.
- /t Type: Kiểu dữ liệu.
- /d Data: Giá trị value.

9. Lệnh Reg delete

Công dụng: Xóa value trong Registry.

Cú pháp: REG DELETE KeyName [/v ValueName] [/f].

Trong đó:

- [/v ValueName]: Tên value cần xóa.

10. Lệnh Regedit.exe

Công dụng: Chạy file .reg.

Cú pháp: Regedit.exe /s “nơi chứa file .reg”.

Trong đó:

- /s: Không cần hỏi.

Lệnh Attrib

Công dụng: Đặt thuộc tính cho file, folder.

Cú pháp: ATTRIB -a -s -h -r “file, thư mục” /s /d hoặc ATTRIB +a +s +h +r “file, thư mục” /s /d.

Trong đó:

- Dấu +: Thêm vào thuộc tính.
- Dấu -: Loại bỏ thuộc tính.
- a: Archive (thuộc tính lưu trữ).
- s: System (thuộc tính hệ thống).
- h: Hidden (thuộc tính ẩn).
- r: Read - only (thuộc tính chỉ đọc).
- /s: Thực hiện với tất cả các file nằm trong thư mục và các thư mục con.
- /d: Đặt thuộc tính cho thư mục và thư mục con.

6.6.3 Các dòng mã CMD trên Windows 10

1. Tasklist

Tasklist cung cấp danh sách toàn bộ các tác vụ đang chạy trong hệ thống. Tất nhiên có một công cụ hiện đại hơn là Task Manager, nhưng tasklist có thể hiển thị những tác vụ bị ẩn khỏi Task Manager.

Một số lệnh mở rộng từ tasklist mà bạn nên biết như **tasklist -svc** (hiển thị dịch vụ liên quan đến từng tác vụ), **tasklist -v** (thông tin chi tiết của mỗi tác vụ) và **tasklist -m** (hiển

thì đường dẫn đến các file .dll liên kết đến tác vụ đang hoạt động.

2. System File Checker

System File Checker là công cụ quét và sửa chữa tự động cho các file hệ thống của Windows. Để sử dụng, bạn cần chạy CMD với quyền **admin**, sau đó nhập **sfc /scannow**. Nếu tìm thấy bất kỳ file bị hỏng hoặc thiếu, công cụ sẽ tự thay thế chúng bằng bản sao được Windows lưu trữ riêng cho mục đích này (thời gian chạy SFC có thể mất nửa giờ).

3. Systeminfo

Lệnh Systeminfo cho bạn biết thông tin cơ bản về hệ điều hành và phần cứng máy tính. Một số thông tin bạn có thể biết với systeminfo như ngày cài đặt Windows, thời gian khởi động gần nhất, phiên bản BIOS, hãng sản xuất máy tính, cấu hình card mạng, bộ xử lý,... Nếu đang kết nối trong hệ thống mạng nội bộ, sử dụng lệnh **systeminfo /s [host_name] /u [domain][user_name] /p [user_password]** để nhận thông tin từ máy tính đó.

4. Powercfg

Powercfg là lệnh rất hữu ích bởi nó giúp bạn quản lý, theo dõi cách mà máy tính tiêu thụ năng lượng. Bạn có thể sử dụng **powercfg hibernate on** và **powercfg hibernate off** để bật tắt tùy chọn ngủ đông, hoặc **powercfg /a** để xem các trạng thái tiết kiệm năng lượng của máy tính.

Một lệnh hữu ích khác là **powercfg /devicequery s1_supported** giúp hiển thị danh sách các thiết bị trên máy tính hỗ trợ Connected Standby (khi được kích hoạt, chúng có thể “đánh thức” máy tính của bạn từ chế độ chờ).

5. PathPing

PathPing là lệnh nâng cao hơn từ ping sẽ hữu ích nếu bạn có nhiều thiết bị mạng cần thử nghiệm. Tương tự ping, sử dụng lệnh này bằng cách gõ **pathping [IP]**. Pathping cung cấp thêm các thông tin về dữ liệu đường dẫn đến địa chỉ, độ trễ, bước nhảy (hop),...

6. File Compare

Lệnh File Compare được sử dụng để tìm ra sự khác nhau trong văn bản giữa hai file, đặc biệt dành cho các lập trình viên muốn tìm ra điểm khác nhau giữa 2 phiên bản của một file, chỉ cần gõ **fc “[đường dẫn đến file 1]” “[đường dẫn đến file 2]”**.

Một số lệnh mở rộng từ fc bao gồm: **fc /b** (chỉ so sánh đầu ra nhị phân), **fc /c** (bỏ qua phân biệt chữ hoa, chữ thường) hay **fc /l** (chỉ so sánh văn bản ASCII).

7. Driverquery

Driver là một phần không thể thiếu giúp kết nối phần cứng với phần mềm, tuy nhiên nếu không được cấu hình đúng cách, nó có thể gây rắc rối cho máy tính của bạn. Mục đích của driverquery là liệt kê chi tiết các driver đang được cài trên máy. Ngoài ra, sử dụng **driverquery -v** nếu muốn có thêm các thông tin (như đường dẫn đến nơi cài đặt driver, ...).

8. Cipher

Nếu bạn thực sự muốn xóa hoàn toàn những dữ liệu đề, bạn có thể dùng lệnh **cipher /w** trong CMD. Lệnh này sẽ ghi đè một dữ liệu ngẫu nhiên nào đó lên ô nhớ để xóa hẳn dữ liệu nằm trên nó trước đây. Ví dụ, **cipher /w c** để xóa dữ liệu đã xóa khỏi thùng rác trong ổ C: (dữ liệu chưa xóa vẫn còn nguyên).

6.7 Cài đặt và sử dụng Linux Bash Shell trên Windows 10

Microsoft đã hỗ trợ cửa sổ dòng lệnh Bash shell ngay trong Windows 10. Windows 10 có một tính năng mới, rất hấp dẫn đối với các nhà phát triển: cửa sổ dòng lệnh Bash trên Ubuntu có thể chạy phần mềm Linux trực tiếp trên Windows. Tính năng này nằm trong "Linux Subsystem for Windows" mà Microsoft vừa mới thêm vào Windows 10. Hiện tại, tính năng này đã được thêm vào bản Insider Preview build 14316 và chỉ hỗ trợ với các phiên bản Windows 10 64-bit mà thôi.

6.7.1 Bash Shell của Windows 10 là gì

Đây không phải là máy ảo, không phải bộ chứa (container) hay phần mềm Linux được biên dịch cho Windows (như Cygwin), mà Windows 10 bổ sung thành phần Linux Subsystem for Windows, dựa trên dự án Project Astoria để chạy ứng dụng Android trên Windows của Microsoft, nhưng hãng đã ngưng dự án này.

Bạn hãy nghĩ tính năng này ngược lại với Wine. Trong khi Wine cho bạn chạy ứng dụng Windows trực tiếp trong Linux thì Linux Subsystem for Windows cho bạn chạy ứng dụng Linux trực tiếp trong Windows. Về mặt kỹ thuật, đây không phải là Linux. Linux là nhân hệ thống (kernel). Thay vào đó, nó cho phép chạy Bash shell và các mã nhị phân hệ như chạy trên Ubuntu Linux.

Hạn chế: Nó không hoạt động với các phần mềm máy chủ và không chạy được với phần mềm có giao diện đồ họa. Mục đích chính của nó là dành cho các nhà phát triển muốn chạy các tiện ích dòng lệnh Linux trên Windows. Những ứng dụng này có thể truy cập trực tiếp đến hệ thống file Windows nhưng ta không thể dùng các lệnh Bash để tự động hóa các chương trình Windows thông thường, hoặc khởi chạy lệnh Bash từ cửa sổ dòng lệnh Windows chuẩn.

6.7.2 Cài Bash lên Windows 10

Để bắt đầu, ta cần đảm bảo hệ điều hành đang chạy sử dụng bản Windows 10 build 14316 trở lên và là bản 64-bit.

Để kiểm tra, ta vào **Settings** và đến mục **Update & Security** → **For Developers**. Kích hoạt "Developer Mode" để bật chế độ dành cho nhà phát triển. → Mở Control Panel → Nhấn "**Programs**" → Nhấn "**Turn Windows Features On or Off**" dưới mục Programs and Features → Kích hoạt "**Windows Subsystem for Linux (Beta)**" trong danh sách và nhấn "OK". → Sau khi thực hiện xong, *hệ thống sẽ nhắc bạn khởi động lại máy tính*.

Nhấn "**Restart Now**" để khởi động lại và Windows 10 sẽ cài thêm tính năng mới này.

Sau khi khởi động xong, nhấn nút Start (hoặc nút Windows), gõ "**bash**" và nhấn "**Enter**".

Lần đầu tiên hệ thống sẽ chạy file bash.exe, và nhắc ta đồng ý các điều khoản sử dụng. Sau đó, lệnh này sẽ tải về ứng dụng "Bash on Ubuntu on Windows" từ Windows Store. Chúng ta sẽ được yêu cầu tạo một tài khoản người dùng, có mật khẩu để sử dụng môi trường Bash.

6.7.3 Cách dùng Bash Shell Ubuntu và cài đặt phần mềm Linux

Đến đây, chúng ta đã có được toàn bộ tính năng cửa sổ dòng lệnh trên Ubuntu. Bởi vì chúng dùng cùng mã nhị phân nên ta có thể sử dụng lệnh apt-get của Ubuntu để cài đặt phần mềm từ các repo Ubuntu như thông thường. Chúng ta có thể truy cập mọi phần mềm chạy trên dòng lệnh của Linux tại đây, mặc dù không phải phần mềm nào cũng chạy mượt mà, nhất là khi đang dùng bản beta.

1. **Mở cửa sổ Bash shell:** Vào menu Start và tìm từ "bash" hoặc "Ubuntu". Chúng ta

sẽ thấy một ứng dụng tên là "Bash on Ubuntu on Windows". Ta có thể pin ứng dụng này để tạo lối tắt cho nó trên menu Start, trên taskbar hay trên màn hình desktop.

Hệ thống file Windows nằm trong đường dẫn `/mnt/c` trong môi trường Bash. Tìm hiểu các lệnh phổ biến của Linux ở đây.

2. Dùng cửa sổ dòng lệnh Windows Command Prompt qua các lệnh DOS để dùng chung một số lệnh ở Bash lẫn Windows.

- Thay đổi thư mục: **cd** trong Bash, **cd** hay **chdir** trong DOS
- Liệt kê nội dung thư mục: **ls** trong Bash, **dir** trong DOS
- Di chuyển hoặc đổi tên file: **mv** trong Bash, **move** và **rename** trong DOS
- Copy một file: **cp** trong Bash, **copy** trong DOS
- Xóa một file: **rm** trong Bash, **del** hay **erase** trong DOS
- Tạo thư mục: **mkdir** trong Bash, **mkdir** trong DOS
- Mở ứng dụng soạn thảo văn bản: **vi** hay **nano** trong Bash, **edit** trong DOS

Điều quan trọng cần nhớ là, không như Windows, Bash và môi trường Linux phân biệt chữ viết hoa và viết thường. Do vậy, "File.txt" sẽ khác với "file.txt".

3. Sử dụng lệnh apt-get để cài đặt và cập nhật phần mềm trong môi trường Ubuntu.

Lưu ý: Cần nhớ bổ sung từ `sudo` vào đầu mỗi dòng lệnh nếu đang dùng với quyền user thông thường, để chạy được với quyền "root", tương đương với quyền Administrator trong Windows.

Dưới đây là vài lệnh liên quan đến **apt-get** mà bạn cần biết:

- Cập nhật các thông tin về các gói phần mềm: **sudo apt-get update**
- Cài một gói ứng dụng: **sudo apt-get install [tên ứng dụng]**
- Gỡ cài đặt một gói ứng dụng: **sudo apt-get remove [tên ứng dụng]**
- Tìm ứng dụng: **sudo apt-cache search [tên ứng dụng]**
- Tải và cập nhật phiên bản mới nhất của các ứng dụng đang có: **sudo apt-get upgrade**

Khi đã tải và cài đặt một ứng dụng nào đó, ta có thể gõ tên nó ở dấu nhắc dòng lệnh và nhấn Enter để chạy ứng dụng.

Lưu ý: Phần mềm bạn cài trong Bash bị hạn chế trong Bash mà thôi. Ta không thể truy cập nó từ Command Prompt, PowerShell hay bất kỳ đâu khác trong Windows. Phần mềm trong Bash cũng không thể tương tác trực tiếp hay khởi chạy các ứng dụng Windows, mặc dù môi trường Bash và Windows đều truy cập được cùng hệ thống file. Tuy vậy, ta vẫn có thể tạo file script Bash (đuôi là `.sh`) để chạy tự động chúng trong môi trường này.