Common Linux commands

1. pwd command

Command **pwd** được dùng để tìm đường dẫn của thư mục hiện tại (folder) mà bạn đang ở trong đó. Command này sẽ trả về đường dẫn hoàn chỉnh (đầy đủ), bắt đầu bằng dấu gạch chéo (/). Ví dụ một đường dẫn hoàn chỉnh là **/home/username**.

2. Command cd

Để chuyển hướng trong hệ thống tập tin Linux, bạn có thể sử dụng command **cd**. Nó sẽ cần nhập đường dẫn đầy đủ hoặc tên thư mục bạn muốn chuyển tới.
Nếu bạn đang ở trong **/home/username/Documents** và muốn đến **Photos**, thư mục con của **Documents**, chỉ cần gõ **cd Photos**.
Trường hợp khác là nếu bạn muốn chuyển sang danh mục hoàn toàn mới, như **/home/username/Movies**. Lúc này, bạn phải gõ **cd** theo danh mục đường dẫn hoàn chỉnh như sau:

```
cd /home/username/Movies
```

Có nhiều cách di chuyển nhanh bằng cd như sau:

- **cd ..** (với 2 chấm) để chuyển lên 1 cấp thư mục trên
- **cd** để tới thẳng thư mục home
- **cd-** (với dấu gạch ngang) để chuyển tới thư mục bạn đã ở trước đó

Mặt khác, shell của Linux phân biệt chữ hoa chữ thường nên bạn phải gõ tên danh mục thật chính xác.

3. Command ls

Command ls được dùng để xem nội dung thư mục. Mặc định là command này sẽ hiển thị danh sách file trong thư mục hiện tại.
Nếu bạn muốn xem nội dung thư mục khác, hãy nhập **ls** và sau đó là đường dẫn thư mục. Ví dụ: nhập **ls /home/username/Document** để xem nội dung của **Documents**.
Có nhiều phiên bản để dùng với lệnh **ls** như sau:

- **ls -R** liệt kê các file bao gồm cả các thư mục phụ bên trong
- **ls -a** liệt kê những file ẩn
- **ls -al** liệt kê tất cả file và thư mục với thông tin chi tiết như phân quyền, kích thước, chủ sở hữu, vân vân.

4. Command cat

cat là một trong các lệnh cơ bản trong Linux được sử dụng thường xuyên nhất trong Linux. Nó được dùng để xem nội dung file trên output tiêu chuẩn (sdout). Để chạy lệnh này, gõ **cat** theo sau là tên file và phần mở rộng**.** Ví dụ: **cat file.txt.**
Có nhiều cách để sử dụng **cat** command linux:

- **cat > filename** tạo ra file mới
- **cat filename1 filename2>filename3** nhập 2 files (1 và 2) để lưu kết quả vào file (3)

- để chuyển một file từ in thường tới in hoa hoặc ngược lại, **cat filename | tr a-z A-Z >output.txt**

5. Command cp

Sử dụng command **cp** để sao chép files từ thư mục hiện tại. Chẳng hạn, command **cp scenery.jpg /home/username/Pictures** sẽ tạo bản copy của **scenery.jpg** vào danh mục **Pictures**.

6. Command mv

Công dụng chính của command **mv** là di chuyển files, dù nó cũng có thể được dùng để đổi tên files.
Arguments trong command này tương tự như command cp. Bạn cần nhập **mv**, tên file và điểm đến của thư mục. Ví dụ: **mv file.txt /home/username/Documents**.
Để đổi tên files, cú pháp là **mv oldname.ext newname.ext**

7. Command mkdir

Command **mkdir** được dùng để tạo thư mục mới – giống như **mkdir Music** sẽ tạo thư mục mới gọi là **Music**.
Một số cách dùng cộng thêm của lệnh **mkdir**:

- Để tạo một thư mục mới bên trong thư mục khác, sử dụng lệnh Linux cơ bảnsau: **mkdir Music/Newfile**
- sử dụng **p** (parents) option để tạo thư mục giữa 2 thư mục đã tồn tại. Ví dụ, **mkdir -p Music/2020/Newfile** sẽ tạo thư mục “2020”

8. Command rmdir

Nếu bạn cần xóa thư mục, sử dụng command **rmdir**. Tuy nhiên, rmdir chỉ cho phép bạn xóa các thư mục trống.

9. Command rm

Command **rm** được sử dụng để xóa thư mục cùng và nội dung bên trong. Nếu bạn chỉ muốn xóa thư mục – tương tự như lệnh rmdir – sử dụng **rm -r**. **Lưu ý**: Khi dùng các lệnh cơ bản trong Linux bạn cần cẩn thận, đặc biệt là lệnh này. Bạn cần kiểm tra kỹ bạn đang ở thư mục nào. Nó sẽ xóa mọi thứ và không khôi phục được.

10. Command touch

Command **touch** cho phép bạn tạo files mới trống thông qua dòng lệnh. Ví dụ: nhập touch **/home/username/Documents/Web.html** để tạo file HTML tiêu đề **Web** trong thư mục **Documents**.

11. Command locate

Bạn có thể sử dụng lệnh này để **locate** (định vị) file, giống như lệnh tìm kiếm trong Windows. Hơn nữa, việc sử dụng argument **-i** với lệnh này làm cho nó không còn phân biệt chữ hoa chữ thường, nên bạn có thể tìm file ngay cả khi không nhớ tên chính xác.

Để tìm file chứa hai hoặc nhiều từ, hãy sử dụng dấu hoa thị **(*)**. Ví dụ: command **locate -i school*note** sẽ tìm tất cả file nào chứa từ “school” và “note”, không phân biệt chữ hoa hay chữ thường.

12. Command find

Tương tự như command locate, command **find** cũng tìm files. Sự khác biệt là bạn sử dụng command find để xác định vị trí files trong thư mục nhất định.
Ví dụ, command find **/home/ -name notes.txt** sẽ tìm file tên **notes.txt** trong thư mục chính và thư mục con của nó.
Một vài biết thể để dùng lệnh **find** là:

- Để tìm file trong thư mục hiện tại, dùng lệnh **find . -name notes.txt**
- Để tìm thư mục được dùng, **/ -type d -name notes. txt**

13. Command grep

Đây là một trong số các lệnh cơ bản trong Linux hữu ích được dùng hằng ngày. Command **grep** cho phép bạn tìm kiếm tất cả text thông qua tập tin nhất định.
Để minh họa, **grep blue notepad.txt** sẽ tìm từ **blue** trong file notepad. Các dòng có chứa từ được tìm sẽ hiển thị đầy đủ.

14. sudo command

Command sudo là viết tắt của **“SuperUser Do”**, cho phép bạn thực hiện các tác vụ yêu cầu quyền quản trị hoặc quyền root. Tuy nhiên, không nên sử dụng lệnh này hàng ngày vì dễ xảy ra lỗi nếu làm sai.

15. Command df

Command **df** dùng để nhận báo cáo về dung lượng lưu trữ được sử dụng trên hệ thống, hiển thị theo tỷ lệ phần trăm và KBs. Nếu bạn muốn xem báo cáo tính bằng megabyte, hãy nhập **df -m**.

16. Command du

Nếu bạn muốn kiểm tra dung lượng của file hoặc của thư mục, command **du** (Disk Usage – Dung lượng lưu trữ) sẽ làm chuyện này. Tuy nhiên, bản tóm tắt về dung lượng lưu trữ sẽ hiển thị block numbers của ổ đĩa thay vì định dạng kích thước thông thường. Nếu bạn muốn xem theo byte, kilobyte và megabyte, hãy thêm argument **-h** vào dòng lệnh.

17. Command head

Command **head** được sử dụng để xem dòng đầu tiên của bất kỳ file văn bản nào. Theo mặc định, nó sẽ hiển thị 10 dòng đầu tiên, nhưng bạn có thể thay đổi số này theo ý mình. Ví dụ: nếu bạn chỉ muốn hiển thị 5 dòng đầu tiên, hãy nhập **head -n 5 filename.ext**.

18. Command tail

Command này có chức năng tương tự như command head, nhưng thay vì hiển thị dòng đầu tiên, command **tail** sẽ hiển thị 10 dòng cuối cùng của file văn bản. Ví dụ, **tail -n filename.ext.**

19. diff command

Viết tắt của difference, command **diff** sẽ so sánh nội dung của 2 files từng dòng một. Sau khi phân tích files này, nó sẽ xuất ra các dòng không khớp nhau. Lập trình viên thường dùng lệnh này khi cần thực hiện một số thay đổi chương trình thay vì viết lại toàn bộ mã nguồn.
Hình thức đơn giản nhất của lệnh này là **diff file1.ext file2.ext**

20. Command tar

Command **tar** là command được sử dụng rộng rãi nhất để lưu trữ nhiều file vào **tarball** – một định dạng file Linux phổ biến tương tự định dạng zip, nhưng nén file thì tùy.
Lệnh này khá phức tạp với danh sách chức năng dài như thêm files mới vào một archive hiện có, liệt kê nội dung của archive, giải nén nội dung từ archive và nhiều chức năng khác. Xem lại **một số ví dụ thực tế** để biết thêm về các chức năng khác.

21. Command chmod

Command **chmod** là một command thiết yếu khác, dùng để thay đổi quyền đọc, ghi và quyền thực thi files và thư mục. Vì lệnh này khá phức tạp nên bạn có thể đọc **hướng dẫn đầy đủ** để thực hiện đúng.

22. Command chown

Trong Linux, tất cả files được sở hữu bởi một người dùng cụ thể. Command **chown** cho phép bạn thay đổi hoặc chuyển quyền sở hữu file sang tên người dùng được chỉ định. Chẳng hạn, **chown linuxuser2 file.ext** sẽ biến **linuxuser2** thành chủ sở hữu **file.ext**.

23. Command jobs

Command **jobs** sẽ hiển thị tất cả jobs hiện tại và trạng thái jobs. Job cơ bản là một tiến trình được tạo bởi shell.

24. Command kill

Nếu có chương trình nào đó không phản hồi, bạn có thể chấm dứt chương trình thủ công bằng cách sử dụng command **kill.** Nó sẽ gửi tín hiệu nhất định đến những ứng dụng đang hoạt động sai và hướng ứng dụng tự chấm dứt.
Có tổng cộng **64 tín hiệu** có thể sử dụng, nhưng mọi người thường chỉ dùng 2 tín hiệu:

- **SIGTERM (15)** – yêu cầu chương trình ngừng chạy và chờ một chút để lưu tất cả tiến trình. Nếu bạn không chỉ định tín hiệu khi nhập command kill, tín hiệu này sẽ được sử dụng.
- **SIGKILL (9)** – buộc các chương trình phải dừng ngay lập tức. Tiến trình nào chưa lưu sẽ bị mất.

Ngoài việc biết về tín hiệu, bạn cũng cần biết Số nhận diện quá trình (PID) của chương trình muốn **kill**. Nếu bạn không biết PID, chỉ cần chạy lệnh **ps ux**.
Sau khi biết tín hiệu muốn dùng và PID của chương trình, hãy nhập cú pháp sau:
kill [signal option] PID.

25. Command ping

Sử dụng command **ping** để kiểm tra trạng thái kết nối của bạn với server. Ví dụ: bằng cách nhập **ping google.com**, lệnh sẽ kiểm tra xem bạn có thể kết nối với Google hay không và đo thời gian phản hồi.

26. Command wget

Một dòng lệnh cực kỳ hữu ích của Linux – bạn có thể tải file từ internet xuống với sự trợ giúp của command **wget**. Để làm được, chỉ cần gõ **wget,** đằng sau là link tải xuống.

27. Command uname

Command **uname**, viết tắt của Unix Name, sẽ in thông tin chi tiết về hệ thống Linux của bạn như tên máy, hệ điều hành, kernel, v.v.

28. Command top

Là terminal tương đương với Task Manager trong Windows, command **top** sẽ hiển thị danh sách tiến trình đang chạy và lượng CPU mà tiến trình đó sử dụng. Rất có ích khi bạn giám sát dung lượng lưu trữ tài nguyên trên hệ thống, đặc biệt là biết được quá trình nào cần chấm dứt vì tiêu thụ quá nhiều tài nguyên.

29. Command history

Khi bạn sử dụng Linux một thời gian nhất định, bạn sẽ nhanh chóng nhận thấy bạn đang chạy hàng trăm lệnh mỗi ngày. Vì vậy việc chạy command **history** đặc biệt hữu ích nếu bạn muốn xem lại những command bạn nhập trước đó.

30. Command man

Bạn nhầm lẫn về chức năng của các commands? Đừng lo, bạn có thể học cách sử dụng chúng dễ dàng từ shell của Linux bằng cách dùng command **man**. Ví dụ, nhập **man tail** sẽ hiển thị hướng dẫn thủ công của command **tail**.

31. Command echo

Lệnh này được dùng để chuyển dữ liệu vào một file. Ví dụ, nếu bạn muốn thêm text “Hello, my name is John” vào trong file name.txt, bạn nhập ệnh **echo Hello, my name is John >> name.txt**

32. Command zip, unzip

Sử dụng lệnh **zip** để nén file thành zip archive và lệnh **unzip** để giải nén file zipped trong zip archive.

33. Command hostname

Nếu bạn muốn biết tên của một máy / mạng máy tính bạn chỉ cần gõ **hostname**. Thêm option **-I** vào cuối file để hiển thị địa chỉ IP address của network bạn.

34. Command useradd, userdel

Vì Linux là một hệ máy đa người dùng, có nghĩa là nhiều hơn 1 người tương tác với máy cùng lúc. **useradd** được dùng để tạo user mới, còn **passwd** đặt mật khẩu cho một tài khoản user. Để thêm user có tên John, **useradd John** rồi thiết lập password bằng lệnh **passwd 123456789.**
Để xóa một user cũng tương tự như thêm user. Để xóa user account, nhập lệnh **userdel UserName**

Thủ thuật và một số mẹo

Sử dụng lệnh **clear** để dọn dẹp terminal cho gọn nếu có quá nhiều command đã được thực thi trước.
Thử dụng nút **TAB** để tự điền vào thông tin đang gõ. Ví dụ, nếu bạn gõ Documents, bạn chỉ cần gõ một số ký tự đầu (như nếu dùng cd thì gõ **cd Docu,** rồi nhấn TAB). Termianl sẽ tự động hoàn tất cho bạn, kết quả nó sẽ hiện **cd Documents**.

Ctrl+C và **Ctrl+Z** được dùng để dừng bất kỳ command nào đang chạy. Ctrl+C sẽ an toàn dừng command một cách an toàn, còn Ctrl+Z sẽ buộc dừng.

Nếu đột ngột terminal bị đóng băng, hãy gõ **Ctrl+S**, chỉ cần unfreeze bằng lệnh **Ctrl+Z**.

Ctrl+A chuyển đến đoạn đầu của dòng trong khi đó **Ctrl+E** chuyển tới cuối dòng.

Bạn có thể chạy nhiều command cùng lúc trong cùng một dòng lệnh bằng cách dùng dấu "**;**" để tắch chúng ra. Ví dụ **Command1; Command2; Command3.** hoặc dùng dấu **&&** nếu bạn muốn chạy các lệnh cơ bản trong Linux một cách tuần tự, lệnh sau chỉ chạy khi lệnh đầu thành công.

Linux Shell script

Định nghĩa: Shell script là một chuỗi các lệnh được viết trong plain text file.

Tạo và thực thi chương trình shell script:

B1: Tạo file hello.sh

```
#!/bin/bash
echo "hello world"
```

Dòng đầu tiên là bắt buộc của shell script, sau # được hiểu là comment, chú thích các đoạn mã.

B2: Để script có thể thực thi cần cấp quyền

```
chmod 0777 hello.sh
```

B3: Thực thi file

```
./hello.sh
```

Biến trong shell

Biến hệ thống:

Tạo và quản lý bởi Linux
Tên biến là chữ hoa

VD:

```
#!/bin/bash
```

```
echo "hello"
echo $BASH_VERSION
echo $BASH
echo $HOME
echo $PATH
```

Biến người dùng

```
tên_biến=value
```

Quy tắc đặt tên:

- Tên biến bắt đầu bằng ký tự
- Không có dấu cách 2 bên toán tử =
- Có phân biết chữ hoa, thường
- Biến không có giá trị khởi tạo thì có giá trị NULL
- Không được dùng dấu ?, * để đặt tên các biến

ECHO để in giá trị của biến

Cú pháp

```
echo $tên_biến
```

In một số ký tự đặc biệt với tham số tuỳ cho -e

```
//example
$ echo -e "Hello\tNam"
#output:  Hello   Nam
$ echo -e "Hello\nNam"
#output
Hello
Nam
```

Các phép toán số học

Cú pháp:

```
expr toán_hạng_1 toán tử toán_hạng_2
```

Phải có dấu cách trước và sau toán tử

Các dấu ngoặc

- Tất cả các ký tự trong dấu ngoặc kép “” đều không có ý nghĩa tính toán, trừ các ký tự sau \ hoặc $
- Dấu nháy ngược `` nghĩa là yêu cầu thực thi lệnh

Cấu trúc điều khiển trong shell script

1. If

```
if điều_kiện
then
câu lệnh 1
```

```
…
fi
```

2. If else

```
if điều_kiện then
      câu_lệnh_1
….
else
      câu_lệnh_2
fi
```

3. For

```
for { tên biến } in { danh sách }
do
…
done

for (( expr1; expr2; expr3 ))
do
…
done
```

4. While

```
while [Điều kiện]
do
…
done
```

Lệnh test

Kiểm tra một biểu thức là đúng hay không và trả lại 0 nếu biểu thức đúng, khác 0 là sai
Cú pháp:

```
test biểu_thức HOẶC  [biểu thức]
```

Các biểu thức so sánh như: -eq, -ne, -lt, -le, -gt, -ge

Git