Cách viết Script cơ bản

1. **Khái niệm cơ bản về Bash Script:**  
   Trong những ngày đầu, máy tính chỉ hiểu được ngôn ngữ nhị phân, đó là điều khó khăn đối với người dùng và cả những nhà phát triển.  
   Theo nhu cầu đó,một chương trình đặc biệt ra đời, nó được gọi là Shell. Shell chấp nhận các câu lệnh tiếng Anh, theo một cú pháp cho trước và chuyển nó đển Kernel để xử lý tiếp, sau đó trả lại kết quả cho người dùng.  
   Shell là chương trình để tương tác giữa người dùng và máy tính. Thông dịch các lệnh của người dùng nhập vào hoặc từ các file.  
   Shell không phải là một phần của hạt nhân hệ thống, nhưng nó dùng hạn nhân để thực thi các chương trình, tạo ra các tập tin…

**Các loại Shell Script:**

Một số Shell có sẵn trên [Linux](http://www.gocit.vn/bai-viet/tag/linux/):

* BASH ( Bourne-Again SHell ) phát triển bởi Brian Fox và Chet Ramey. Đây là Shell thông dụng nhất trên Linux.
* CSH (C SHell) phát triển bởi Bill Joy tại University of California (dành cho BSD). Sử dụng cấu trúc lệnh giống C, rất thân thiện cho các lập trình viên C trên linux.
* KSH (Korn SHell) phát triển bởi David Korn tại AT & T Bell Labs.
* TCSH bạn có thể gọ lệnh #man tcsh để xem thông tin.

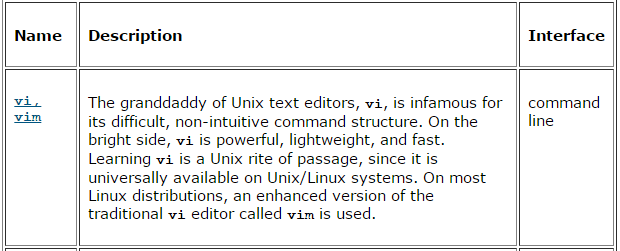
Để xem hệ thống của bạn hỗ trợ những loại shell nào, có thể dùng lệnh #cat /etc/shells. Mỗi loại shell có cú pháp, cách dùng khác nhau, cung cấp cho bạn các chức năng khác nhau.  
Trong MS-DOS, Shell là được cọi là [COMMAND](http://www.gocit.vn/bai-viet/tag/command/).COM, cũng được sử dụng cho mục đích tương tự, nhưng không mạnh mẽ bằng Shell Linux.  
Tất cả các loại Shell phía trên đều đọc các lệnh người dùng nhập và đưa đến Linux OS để nói với OS rằng người dùng đang muốn gì. Các dòng lệnh được gọi là command line.  
Để xem system đang chạy shell gì, bạn có thể dùng lệnh**#**echo $SHELL

**2. Chương trình Script đầu tiên:**

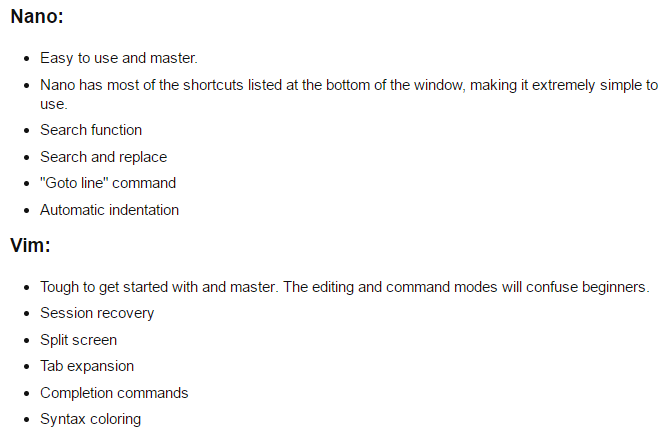
**A. Giới thiệu về trình soạn thảo script trên Linux**

- link tham khảo : <http://linuxcommand.org/wss0010.php>

- http://www.freeos.com/guides/lsst/ch02sec01.html



**\*** Ngoài ra còn có thêm **1 trình soạn thảo** phổ biến nữa trong Linux, dưới đây là sự so sánh giữa **Nano** và **Vim** :



- **Nano** thường được sử dụng cho những file script đơn giản còn **Vim** là trình soạn thảo nâng cao cùng nhiều chức năng hơn cũng như độ phức tạp cao hơn.

**B. Viết script đầu tiên “ Hello world “ :**

**B1. Tạo file :** bằng lệnh [ **vi filename** ]



**B2. Tạo các lệnh trong trình soạn thảo**

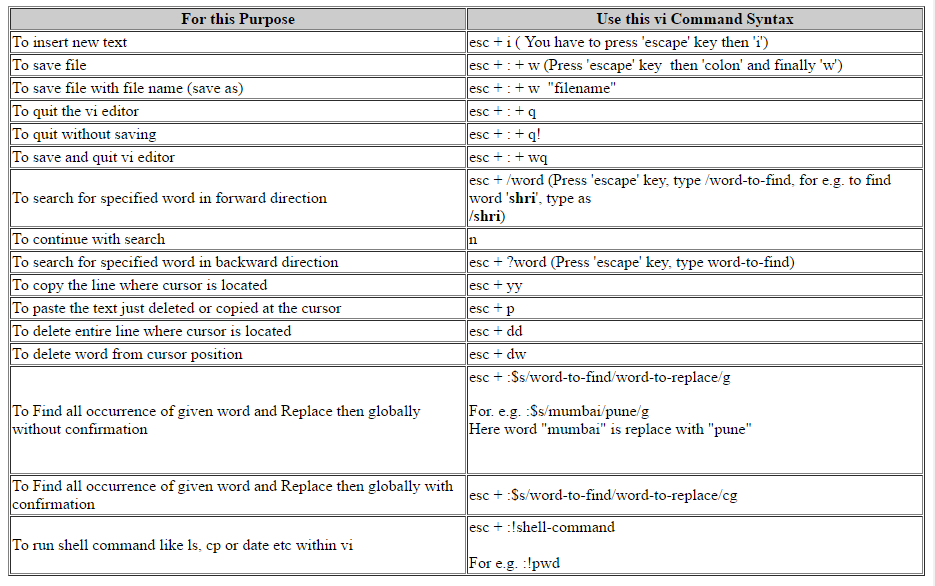
* ấn ESC + i để soạn thảo



**B3. Lưu file :**

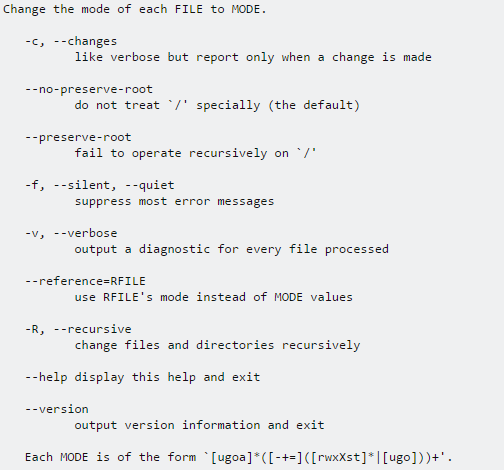
* C1: lưu file **:x**
* C2: lưu file và thoát **:wq**

Một số **Vi command cần thiết:**

****

**B4. Lưu file và cấp quyền cho file:**

* Lưu file bằng lệnh **chmod +x hello.sh**
  + Lệnh **chmod +x FILENAME**  có nghĩa là sẽ cấu hình phân quyền cho file để file có thể chạy được
  + Tham khảo thêm về cách mode của **chmod :**

****

* + Sau khi thực hiện lệnh để file có thể chạy được, thực hiện lệnh sau để phân quyền cho file script:
    - **Chmod [quyền] Filename**

Vd: 

* Chạy file bằng lệnh **./hello.sh**

****

Hoặc

**sh hello.sh**



(\*)Lưu ý : trong mục này, dấu “ ./ “ có nghĩa là ta đang chạy file trên thư đường dẫn thư mục hiện tại, và chỉ có mình dấu chấm (.) là mang ý nghĩa chạy file command trên chính shell đó mà không phải tạo ra một bản sao mới của shell. Vd : **$ . hello.sh**

Hoặc

**bash filename**

1. **Một số ví dụ đơn giản làm quen với Bash Shell Script:**

Dòng đầu tiên chúng ta luôn đặt **#!/bin/bash**, đây là cú pháp bắt buộc. Sau **#** được hiểu là comment, chú thích của các đoạn mã.

**Backup thư mục từ xa:**

**Step 1:** Tạo một đường dẫn với tên Zimbra\_Scripts và đi tới thư mục đó:

mkdir Zimbra\_Scripts; cd Zimbra\_Scripts

**Step 2**: tạo và chỉnh sửa file có tên là “ imap\_users” và điền tên người dung, mật khẩu của tất cả domains và người dung trên zimbra.server.com:

domains=(

domain1 \

domain2 \

users=(

user1@domain1 \

user2@domain1 \

passwords=(

u1d1passwd \

u2d1passwd \

)

Step 3: Tạo một tệp tin khác trong thư mục 'Zimbra\_Scripts' - gọi là 'zimbra\_migration' và nhập code chính vào.Mã này tái tạo các tên miền hiện đang có trên Zimbra.server.com lên backup.server.com, sẵn sàng nhận email trong bước tiếp theo. Đối với quản trị viên có nhiều tên miền, việc tạo lại từng nhóm một có thể là một quá trình tốn thời gian.

- Dòng đầu tiên bắt đầu với một khoảng thời gian. Và đảm bảo tập tin 'imap\_users' được bao gồm trong script 'zimbra\_migration'.

Điều này đã được thực hiện liên quan đến lập trình hướng đối tượng, cho tính di động và bảo mật.

nano zimbra\_migration

. imap\_users

for ((i = 0 ; i < ${#domains[@]} ; i++ ))

do

echo $i: /opt/zimbra/bin/zmprov cd ${domains[$i]}

done

for ((i = 0 ; i < ${#users[@]} ; i++ ))

do

echo $i: /opt/zimbra/bin/zmprov ca ${users[$i]}

/opt/zimbra/bin/zmprov ca ${users[$i]} ${passwords[$i]}

done

Chúng ta sử dụng hai vòng lặp, lặp lại đầu tiên thông qua các mảng của các domains. Một số code để gỡ lỗi, giúp ta bằng cách hiển thị số id của tên miền trong vòng lặp và sau đó tạo ra tên miền bằng cách gọi :

zmprov cd

Giá trị miền được lấy từ mảng 'domains' nằm trong file 'imap user'.

- Bạn có thể chạy lại tập lệnh này bất cứ lúc nào, nếu tên miền đã tồn tại - nó sẽ KHÔNG được ghi đè, nhưng một thông báo cho bạn biết rằng miền đang tồn tại sẽ xuất hiện ở đây.

Trong vòng lặp thứ hai, một lần nữa một mã gỡ lỗi nhỏ cho ta biết ta đang ở đâu trong danh sách người dùng và người dùng nào đang được tạo.

zmprov ca

Tạo người dùng, lấy các đối số của nó từ mảng 'user' trong tệp 'imap\_users' và mật khẩu từ cùng một vị trí mảng của mảng 'passwords', cùng nằm trong tệp 'imap\_users'.

**Step 4:**

Thiết lập quyền trên tất cả các tệp để có thể thực hiện các tệp tin :

chmod +x \*

Chạy file zimbra\_migration:

./zimbra\_migration

Sử dụng **LDAP :**

Đăng nhập vào Zimbra vs tài khoản người dùng

su - zimbra

Xuất dữ liệu LDAP ra một .ldif file:

./openldap/sbin/slapcat -F /opt/zimbra/data/ldap/config -b "" -l /tmp/zimbra-ldap.ldif

Đăng nhập vào Zimbra vs tài khoản người dùng

su - zimbra

Tắt dịch vụ zimbra server:

zmcontrol stop

Kiểm tra thủ công xem có bất kỳ dịch vụ imbra nào đang chạy không - (tắt nếu cần thiết):

ps auxx | grep slapd

Xóa các dữ liệu hiện tại :

rm -f openldap-data/\*

Nhập tài khoản người dùng và mật khẩu từ tệp LDAP zimbra-ldap.ldif

rm -f data/ldap/hdb/db/\_\_db.\* data/ldap/hdb/db/\*.bdb data/ldap/hdb/db/alock data/ldap/hdb/logs/\*

**IMAPSYNC & Cronjob IMAPSYNC**

Tạo tệp script cho công việc định kỳ 'imapsync\_crontab' – vẫn trong thư mục 'Zimbra\_Scripts'.

nano imapsync\_crontab

. imap\_users #if in the same folder as imap\_users else full-path to imap\_users (e.g = " . /path/to/imap\_users")

src\_srv = zimbra.server.com

dest\_srv = backup.server.com

for ((i = 0 ; i < ${#users[@]} ; i++ ))

do

/usr/bin/imapsync --noauthmd5 --syncinternaldates --subscribe \

--host1 $src\_srv --ssl1 --user1 ${users[$i]} --password1 ${passwords[$i]} \

--host2 $dest\_srv --ssl2 --user2 ${users[$i]} --password2 ${passwords[$i]}

done

Ta thiết lập địa chỉ máy chủ nguồn là 'src\_srv' và địa chỉ máy chủ đích là 'dest\_srv' - để sử dụng trong các truy cập đến imapsync.

Một vòng lặp duy nhất lặp qua mảng có chứa danh sách 'người dùng và mật khẩu của chúng.

Các tùy chọn imapsync sau được sử dụng:

* Noauthmd5 - không sử dụng md5
* Synacterternaldates - để giữ đúng ngày và giờ email được gửi / nhận
* Subscribe - tự động tạo bất kỳ thư mục người dùng bổ sung nào trên máy đích (giữ nguyên cấu trúc thư mục)

Host1 - đây là máy chủ nguồn 'zimbra.server.com'

Host2 - đây là máy chủ đích 'backup.server.com'

Ssl1 / 2 - sử dụng kết nối an toàn trên cả hai máy

Password1 / 2 - có thể là giống nhau trên cả hai máy

**Tạo Crontab:**

* **Tham khảo : https://hocvps.com/tong-quat-ve-crontab/**

export EDITOR=$(type -P nano);crontab -e

Bây giờ, thiết lập việc đồng bộ hóa hàng giờ với máy chủ đích. Điều này giả định thư mục 'Zimbra Scripts' nằm trong '/ opt / Zimbra Scripts'.

Đầu ra được gửi qua email tới [admin@zimbra.server.com](mailto:admin@zimbra.server.com).

MAILTO=admin@zimbra.server.com

@hourly /opt/Zimbra\_Scripts/imapsync\_crontab

Điều này đảm bảo rằng email người dùng được sao lưu hàng giờ.