

TRUNG TÂM ĐÀO TẠO NEWSTAR

Newstar

**QUẢN TRỊ
HỆ THỐNG**

LINUX

LPI 1



Biên soạn: HUỲNH THANH TÂM

TRUNG TÂM ĐÀO TẠO NEWSTAR



Quản trị hệ thống LINUX

(LPI 1)

Biên soạn: Huỳnh Thanh Tâm

8-2017

Mục Lục

A. LPI-1:

Chương 1: TỔNG QUAN	2
I. Lịch sử phát triển	2
II. Các distro phổ biến của Linux	2
III. Đặc điểm của Linux	4
IV. Giới thiệu Centos 7	5
Chương 2: CÀI ĐẶT LINUX SERVER	7
Chương 3: CẤU TRÚC HỆ ĐIỀU HÀNH LINUX	10
Chương 4: CÁC LỆNH CƠ BẢN	14
Chương 5: BOOT AND RUN LEVELS	22
Chương 6: QUẢN TRỊ USER, GROUP	25
Chương 7: QUYỀN TRÊN HỆ THỐNG TỆP TIN, THƯ MỤC	32
Chương 8: TRÌNH SOẠN THẢO VI & CÁC LỆNH LỌC DỮ LIỆU	38
Chương 9: NETWORK CONFIGURATION	43
Chương 10: INSTALL SOFTWARE	48
Chương 11: LINUX FILE SYSTEMS, DEVICES	52
Chương 12: MỞ RỘNG HDD	54
Chương 13: LẬP SHELL SCRIPT	62

Chương 1: TỔNG QUAN

I. Lịch sử phát triển

Linux là hệ điều hành (HĐH) tương tự với Microsoft Windows, nhưng Linux được tự do phát triển theo nhu cầu sử dụng và hoàn toàn miễn phí, Linux được công bố dưới bản quyền của GPL (General Public Licence).

Unix một hệ điều hành được viết trên assembler cho máy PDP-11/20 ra đời vào năm 1970. Unix được xây dựng bởi Ken Thompson và Dennis Richie trong dự án Multics viết lại hệ điều hành đa-bài toán trên máy PDP-7.

Ngày 5/4/1991, Linus Torvalds, chàng sinh viên 21 tuổi của trường Đại học Helsinki - Phần Lan đã bắt tay vào viết những dòng lệnh đầu tiên của Linux. Linux được xây dựng trên cơ sở cải tiến một phiên bản UNIX có tên Minix do Giáo Andrew S. Tanenbaum xây dựng và phổ biến.

Sau ba năm nhân Linux ra đời, đến ngày 14-3-1994, hệ điều hành Linux phiên bản 1.0 được phổ biến. Thành công lớn nhất của Linux 1.0 là nó đã hỗ trợ giao thức mạng TCP/IP chuẩn UNIX, sánh với giao thức socket BSD- tương thích cho lập trình mạng. Trình điều khiển thiết bị đã được bổ sung để chạy IP trên một mạng Ethernet hoặc trên tuyến đơn hoặc qua modem. Hệ thống file trong Linux 1.0 đã vượt xa hết hổng file của Minix thông thường, ngoài ra đã hỗ trợ điều khiển SCSI truy nhập đĩa tốc độ cao. Điều khiển bộ nhớ ảo đã được mở rộng để hỗ trợ điều khiển trang cho các file swap và ánh xạ bộ nhớ của file đặc quyền Vào tháng 3-1995, nhân 1.2 được phổ biến. Điều đáng kể của Linux 1.2 so với Linux 1.0 ở chỗ nó hỗ trợ một phạm vi rộng và phong phú phần cứng, bao gồm cả kiến trúc tuyến phần cứng PCI mới.

Đến thời điểm này có hơn 200 Distro của Linux. Các bạn có thể xem các distro này tại website <http://distrowatch.com>

II. Các distro phổ biến của Linux

1) Debian:

Debian là một distro chứa số lượng các phần mềm rất lớn, Debian được xây dựng bởi một tổ chức nguyên tình nguyện công hiến để phát triển phần mềm tự do và đẩy mạnh những lý tưởng của cộng đồng phần mềm tự do. Dự án Debian xây dựng vào năm 1993. Debian có tiếng về mối liên kết chặt chẽ với triết lý Unix và phần mềm tự do. Nó cũng có tiếng về sự phong phú cho các chọn lựa: phiên bản phát hành hiện tại có hơn 29000 gói phần mềm cho 11 kiến trúc máy tính, từ kiến trúc ARM thường gặp ở các hệ thống nhúng và kiến trúc máy tính lớn s390 của IBM cho đến các kiến trúc thường gặp trên máy tính cá nhân hiện đại như x86 và PowerPC.



debian

Debian được hỗ trợ nhờ các khoản quyên góp thông qua tổ chức Software in the Public Interest, một tổ chức bảo trợ phi lợi nhuận cho các dự án phần mềm tự do.

2) Ubuntu:

Hệ điều hành Ubuntu là một trong những bản phân phối (distro) Linux phổ biến nhất hiện nay do Mark Shuttleworth sáng lập và công ty Canonical của ông tài trợ. Hệ điều hành này được sử dụng phổ biến và ưa chuộng vì có giao diện đẹp, thân thiện, dễ sử dụng, kho phần mềm ứng dụng rất phong phú đáp ứng được hầu hết yêu cầu của người dùng, các version được cập nhật liên tục.



ubuntu

Ubuntu được chia làm 2 loại:

- ✓ Ubuntu Desktop: Sử dụng cho người dùng cuối (end user). Nó tương tự như windows XP, Win 7, win 8.
- ✓ Ubuntu Server: Sử dụng để cài các dịch vụ phục vụ cho người dùng cuối. Giống như Windows Server.

3) RedHat Enterprise Linux:

Red Hat Enterprise Linux (RHEL) là một bản phân phối Linux được phát triển bởi Red Hat và mục tiêu hướng tới thị trường thương mại.

Red Hat Enterprise Linux là một trong những hệ điều hành an toàn nhất hiện có, sẵn sàng đáp ứng khối lượng tính toán cường độ cao out-of-the-box. Ngoài ra còn cung cấp bộ các tùy chỉnh tối ưu hiệu suất giúp bạn điều



redhat

chính và sắp xếp hành vi hệ thống tùy theo khối lượng công việc cụ thể. Được triển khai trong các cơ quan chính phủ, tài chính - ngân hàng, nơi mà bảo vệ dữ liệu là việc quan trọng nhất.

4) Centos

CentOS là viết tắt của Community Enterprise Operating System. Centos là một bản phân phối hệ điều hành tự do dựa trên Linux kernel. Nó có nguồn gốc hoàn toàn từ bản phân phối Red Hat Enterprise Linux (RHEL). Những người phát triển CentOS đã tận dụng mã nguồn của Red Hat để tạo ra một sản phẩm hoàn toàn tương tự tuy nhiên cái giá của nó là miễn phí. Ngày nay, CentOS thường được sử dụng trong các doanh nghiệp bởi tính ổn định vòng đời End-of-Life dài.



5) Kali Linux

Kali Linux là một distro của Linux, được sử dụng để kiểm tra, tấn công thử nghiệm vào các lỗ hổng của hệ thống công nghệ thông tin. Kali linux được viết dựa trên nền tảng của Debian được đồng bộ hóa với các Repository của Debian nên có thể dễ dàng có được các bản cập nhật vá lỗi bảo mật mới nhất và các cập nhật Repository. Đây là một phiên bản tiến hóa của Backtrack và distro này rất hữu ích đối với những chuyên gia đánh giá bảo mật.



III. Đặc điểm của Linux

Tuy Linux có nhiều Distro nhưng chúng đều có những đặc điểm chung như sau:

- ✓ Linux tương thích với nhiều hệ điều hành như DOS, MicroSoft Windows ...
- ✓ Cho phép cài đặt Linux cùng với các hệ điều hành khác trên cùng một ổ cứng. Linux có thể truy nhập đến các file của các hệ điều hành cùng một ổ đĩa. Linux cho phép chạy mô phỏng các chương trình thuộc các hệ điều hành khác.
- ✓ Do giữ được chuẩn của UNIX nên sự chuyển đổi giữa Linux và các hệ UNIX khác là dễ dàng.

- ✓ Linux là một hệ điều hành UNIX tiêu biểu với các đặc trưng là đa người dùng, đa chương trình và đa xử lý.
- ✓ Linux có giao diện đồ họa (GUI) thừa hưởng từ hệ thống X-Window. Linux hỗ trợ nhiều giao thức mạng, bắt nguồn và phát triển từ dòng BSD. Thêm vào đó, Linux còn hỗ trợ tính toán thời gian thực.
- ✓ Linux khá mạnh và chạy rất nhanh ngay cả khi nhiều quá trình hoặc nhiều cửa sổ.
- ✓ Linux ngày càng được hỗ trợ bởi các phần mềm ứng dụng bổ sung như soạn thảo, quản lý mạng, quản trị cơ sở dữ liệu, bảng tính ...
- ✓ Linux hỗ trợ tốt cho tính toán song song và máy tính cụm (PC-cluster) là một hướng nghiên cứu triển khai ứng dụng nhiều triển vọng hiện nay.

IV. Giới thiệu Centos 7

Centos được phát triển dựa trên nhân RedHat Enterprise Linux. Centos có nhiều Version, các version ra sau đều kế thừa, khắc phục, bổ sung và tối ưu các tính năng của các phiên bản trước. Mỗi version đều có thời hạn sử dụng riêng, cụ thể như sau:

- ✓ *CentOS 5.x – End of Life 03/31/2017*
- ✓ *CentOS 6.x – End of Life 11/30/2020*
- ✓ *CentOS 7.x – End of Life 06/30/2024*

Version mới nhất của Centos là version 7 (64 bit) được phát hành vào tháng 07/2014. Những thay đổi nổi bật của Centos 7:

- ✓ *Updated Kernel to 3.10.0*
- ✓ *Added support for Linux Containers*
- ✓ *Open VMware Tools & 3D graphics drivers out of the box*
- ✓ *OpenJDK-7 as default JDK*
- ✓ *Upgrade from 6.5 to 7.0 using preupg command*
- ✓ *LVM-snapshots with ext4 and XFS*
- ✓ *Switch to grub2, systemd and firewalld*
- ✓ *Default XFS file system*

- ✓ *iSCSI and FCoE in kernel space*
- ✓ *Support for PTPv2*
- ✓ *Support for 40G Ethernet Cards*
- ✓ *Supports installations in UEFI (Unified Extensible Firmware Interface) Secure Boot form on compatible hardware*

newstar.vn

Chương 2: CÀI ĐẶT LINUX SERVER

Linux có nhiều distro, mỗi distro có nhiều version. Việc chọn lựa distro nào, version để cài đặt và phát triển ứng dụng phải dựa vào các yếu tố:

- ✓ Lựa chọn những distro nổi tiếng
- ✓ Dựa vào mục đích sử dụng
- ✓ Chọn distro mà người quản trị biết sử dụng
- ✓ Chọn version cho distro với thời hạn *end of life* phù hợp

Trong chương trình Linux 1 sẽ sử dụng Centos 7 để hướng dẫn các học viên cách quản lý và phát triển ứng dụng. Bởi vì, Centos là một distro nổi tiếng, được sử dụng rộng rãi tại các doanh nghiệp.

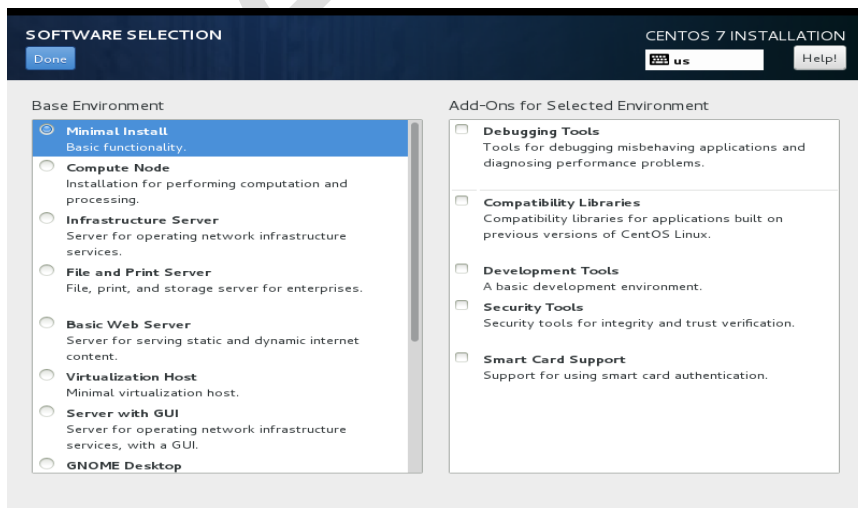
Trong chương này sẽ hướng dẫn cách cài đặt Centos 7 trên VMware Workstation.

Để tiến hành cài đặt các học viên cần chuẩn bị:

- File cài đặt *CentOS-7.0-1406-x86_64-DVD.iso* hoặc đĩa CDROM.
- Cài đặt chương trình VMware Workstation trên Windows.

Chi tiết các bước cài đặt xem tại **phụ lục I**.

Một vài điểm chú ý khi cài đặt:



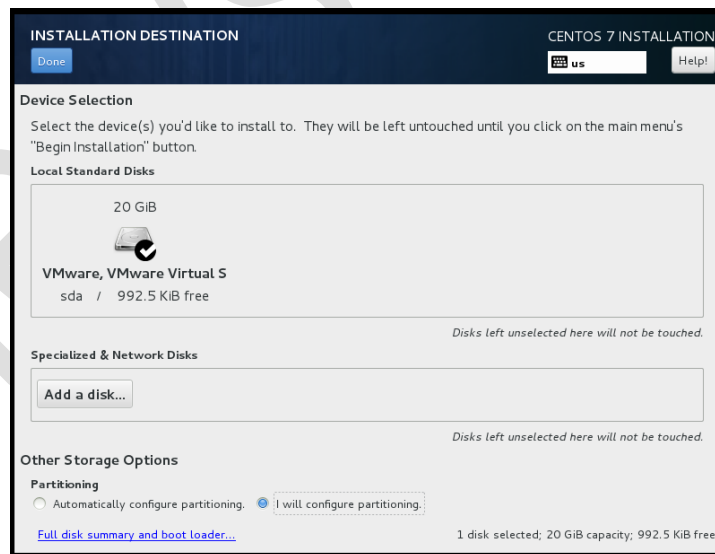
Software Selection:

- ✓ **Minimal Install:** Chỉ cài đặt các gói cần thiết cho HĐH, đây được xem như bản rút gọn của Centos. Sau khi cài xong người quản trị chỉ làm việc với giao diện dòng lệnh (giống MS DOS).
- ✓ **GNOME Desktop:** cung cấp giao diện đồ họa trực quan cho người dùng tương tác với HĐH.

Các options khác như: *Compute Node, Infrastructure Server, File and Print Server, Basic Web Server, Virtualization Host, Server with GUI...* khi cài đặt HĐH sẽ cài các thư viện tương ứng với các ứng dụng lựa chọn. ví dụ: bạn chọn *Basic Web Server* thì việc cài đặt sẽ thực hiện cài các thư viện dành cho dịch vụ Web.

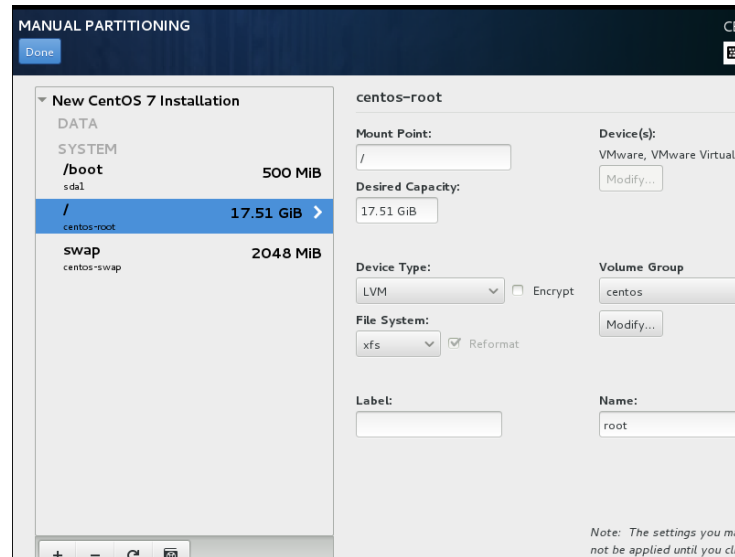
Tuy nhiên, các chương trình ứng dụng này ta có thể cài đặt sau khi cài đặt xong một HĐH với bất kỳ lựa chọn nào trong **Software Selection**.

Đối với các học viên học chương trình Linux 1, các bạn nên chọn chế độ cài GNOME Desktop.



Installation Destination:

- ✓ **Automatically configure partitioning:** chế độ chia partition tự động
- ✓ **I will configure partitioning:** chế độ chia partition thủ công.



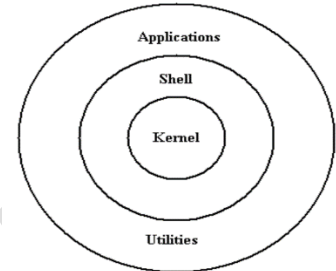
Manual Partitioning: Thực hiện chia 3 partition

- ✓ */boot*: dùng để chứa các thư viện cho việc khởi động HĐH, dung lượng 500MB
- ✓ *Swap*: phân dùng hoán đổi không gian địa chỉ vật lý. Dùng để tính toán và lưu kết quả tạm thời, dung lượng = RAM *2.
- ✓ */*: thư mục gốc, chứa các thành phần còn lại, dung lượng còn lại của ổ đĩa.

Chương 3: CẤU TRÚC HỆ ĐIỀU HÀNH LINUX

I. Kiến trúc tổng quan hệ thống Linux:

Kiến trúc của HĐH Linux chia làm 3 thành phần: Kernel, Shell, Applications



1) **Kernel (nhân):** Đây là phần quan trọng và được ví như trái tim của HĐH, Phần kernel chứa các module, thư viện để quản lý và giao tiếp với phần cứng và các ứng dụng. Kernel trên Centos 7 có version 3.10.0.

2) **Shell:** Shell là một chương trình. Có chức năng thực thi các lệnh (command) từ người dùng hoặc từ các ứng dụng – tiện ích yêu cầu chuyển đến cho Kernel xử lý. Bên cạnh đó, shell còn có khả năng bảo vệ kernel từ các yêu cầu không hợp lệ.

Các loại shell:

- ✓ *Sh (the Bourne Shell):* đây là shell nguyên thủy của UNIX được viết bởi Stephen Bourne vào năm 1974. Đến nay shell sh vẫn sử dụng rộng rãi.
- ✓ *Bash(Bourne-again shell):* đây là shell mặc định trên linux.
- ✓ *csH (the C shell):* shell được viết bằng ngôn ngữ lập trình C, được viết bởi Bill Joy vào năm 1978.
- ✓ Ngoài ra còn có các loại shell khác như: *ash* (the Almquist shell), *tsh* (the TENEX C shell), *zsh* (the Z shell).

Dấu nhắc Shell thay đổi tùy thuộc vào tài khoản user đang làm việc. Khi làm việc với tài khoản user root, dấu nhắc shell có dạng:

```
[root@localhost root]#
```

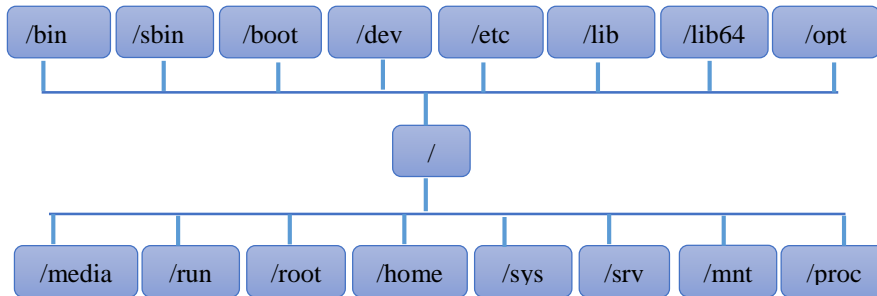
Khi làm việc với tài khoản user thường, dấu nhắc shell có dạng:

```
[linux@localhost linux]$
```

3) **Applications:** Là các ứng dụng và tiện ích mà người dùng cài đặt trên Server. Ví dụ: ftp, samba, Proxy, ...

II. Cấu trúc hệ thống File:

Cấu trúc hệ thống file trên Centos được bố trí theo dạng hình cây (tree) như sau:



Bắt đầu là thư mục gốc “/”, sau đó là các thư mục con (hay còn gọi là nhánh): /bin, /sbin, /home, /mnt ...

Mỗi thư mục con của thư mục gốc có các chức năng khác nhau.

- ✓ **/bin:** Chứa các file binary của các tập lệnh trong Linux.
- ✓ **/sbin:** Tương tự như /bin, nhưng là những lệnh chỉ được dùng bởi quản trị hệ thống - tương đương root user.
- ✓ **/boot:** Chứa các thư viện cần thiết cho quá trình khởi động.
- ✓ **/dev:** Chứa thông tin chứa các file thiết bị. Trong Linux, mỗi thiết bị đều có file đại diện và được đặt tên theo 1 Logic nhất định:
 - cdrom : đĩa CDRom / DVD
 - fd* : Đĩa mềm
 - hd* : Đĩa cứng IDE
 - sd* : Đĩa cứng SCSI
 - st* : Băng từ
 - tty* : cổng giao tiếp (COM,...)
 - eth* : card ethernet
- ✓ **/etc:** Chứa file cấu hình hệ thống và ứng dụng.
- ✓ **/lib:** Chứa thư viện chia sẻ được dùng bởi các tiến trình, các lệnh boot, lệnh hệ thống như trong /bin và /sbin.
- ✓ **/lib64:** Tương tự như lib nhưng dành cho 64bit.
- ✓ **/opt:** Nơi dành riêng cho các tiện ích chương trình được cài đặt.
- ✓ **/media:** Thư mục có vai trò như đích đến của quá trình mount point. Khi gắn 1 thiết bị lưu trữ bên ngoài, để sử dụng, cần mount thiết bị này vào /media, từ đó,

các thư mục, tập tin sẽ được chuyển vào đây (lúc này /media có thể coi như ảnh chiều của thiết bị).

- ✓ **/run:**
- ✓ **/root:** Thư mục home của user root.
- ✓ **/home:** Thư mục chứa các thư mục home của các user được tạo.
- ✓ **/sys:**
- ✓ **/srv :** chứa dữ liệu, các file của các dịch vụ trên hệ thống.
- ✓ **/mnt:** Thư mục này được dùng để gắn các hệ thống tập tin tạm thời (mounted filesystems).
- ✓ **/proc:** Lưu các thông tin về tình trạng của hệ thống.

III. Các kiểu file:

Trên linux tất cả mọi thứ đều được xem dưới dạng là file. Có 3 loại file: file thông thường (Regular files), file thư mục (Directory files), file đặc biệt (Special files).

- ✓ **File thông thường:** một chương trình, file text, library, file nhạc ...
- ✓ **Thư mục:** thành phần dùng để chứa các file khác (container).
- ✓ **File đặc biệt:** (device, socket, pipe, symbolic links ...).

Symbol	Meaning
-	Regular file
d	Directory
l	Link
c	Special File
s	Socket
p	Named Pipe
b	Block Device

Các file ẩn thường bắt đầu bằng dấu “.”

IV. Đường dẫn:

Đường dẫn là một trong những phần quan trọng đối với các học viên đang làm quen với Linux, đây là thành phần xuyên suốt trong quá trình sử dụng các lệnh trên hệ thống. Hiểu rõ về cách sử dụng các loại đường dẫn sẽ giúp ích rất nhiều cho việc học tập và là quản trị hệ thống Linux.

Có 3 loại đường dẫn:

- ✓ **Đường dẫn tuyệt đối:** bắt đầu bằng “/”.

Ví dụ: `/etc/sysconfig`

- ✓ **Đường dẫn tương đối:** không bắt đầu bằng “/”.

Ví dụ: `etc/sysconfig`

- ✓ **Đường dẫn đặt biệt:** “..” Thư mục cha.

“.” Thư mục hiện tại.

newstar.vn

Chương 4: CÁC LỆNH CƠ BẢN

Khi làm việc với HĐH Linux hầu như tất cả người dùng đều sử dụng lệnh để làm việc, vì vậy các học viên cần nắm vững các lệnh cơ bản. Lệnh là một chương trình hoặc một script dùng để thực thi một nhiệm vụ nào đó. Lệnh được gõ sau dấu nhắc shell.

Dòng lệnh shell tổng quát có dạng như sau:

`command` `[options]`

Trong đó:

<code>command</code>	Lệnh
<code>options</code>	Tùy chọn, thường bắt đầu bằng - hoặc -- Nhiều tùy chọn có thể kết hợp bằng một ký hiệu - ví dụ: -lF thay vì -l -F
<code>arguments</code>	tham số lệnh

Chú ý: Dòng lệnh shell có phân biệt chữ thường và chữ hoa.

I. Lệnh trợ giúp: man

Mỗi lệnh trên Linux có rất nhiều options, mỗi options thực hiện các chức năng khác nhau, người quản trị cũng không cần nhớ hết các options của lệnh mà chỉ cần nhớ một vài options thông dụng. Để biết một lệnh có bao nhiêu options cũng như chức năng của từng options thì lệnh đầu tiên cần phải biết là lệnh: **man**

Cấu trúc lệnh:

```
man <tên_lệnh>
```

Ví dụ: Để xem hướng dẫn sử dụng lệnh cp (copy) có thể nhập lệnh

```
$man cp
```

Để thoát khỏi man ta bấm phím “q”

II. Các lệnh kiểm tra performance

Sau khi cài đặt xong Server Linux, người quản trị cần phải biết thông tin cấu hình của Server mình quản trị. Các thông cấu hình cần biết như: RAM, CPU, HDD, số serial, phiên bản Centos, số bit của OS, các tiến trình đang chạy ...

✓ **Lệnh xem thông tin RAM:** Xem tổng dung lượng, dung lượng hiện tại đang dùng, dung lượng còn trống. có 2 lệnh đó là:

- *cat /proc/meminfo*

Lệnh *cat*: Dùng để đọc nội dung của file text

/proc/meminfo: đây là đường dẫn (đường dẫn tuyệt đối) tới file chứa thông tin RAM có tên là *meminfo*

- *free*

các options:

-b	Hiển thị theo bytes
-k	Hiển thị theo kilobytes
-m	Hiển thị theo megabytes
-g	Hiển thị theo gigabytes
--tera	Hiển thị theo terabytes
-h	Hiển thị theo kiểu tự động gom theo khối dữ liệu

Ví dụ:

```
[root@localhost ~]$ free -k
              total        used         free       shared  buff/cache   available
Mem:          1017480      520256         81168          8336      416056      314520
Swap:          2097148           0       2097148

[root@localhost ~]$ free -m
              total        used         free       shared  buff/cache   available
Mem:           993          508           79           8           406          307
Swap:           2047           0          2047

[root@localhost ~]$ free -h
              total        used         free       shared  buff/cache   available
Mem:           993M          510M          76M          8.1M       406M          304M
Swap:           2.0G           0B          2.0G

[root@localhost ~]$
```

✓ **Lệnh xem thông tin CPU:** *cat /proc/cpuinfo*

✓ **Lệnh hiển thị thông tin kernel:** *uname -a*

options:

-a : all information

✓ **Lệnh hiển thị phiên bản Centos:** *cat /etc/redhat-release*

✓ **Lệnh xem dung lượng ổ cứng:** Xem dung lượng ổ cứng đã dùng và còn trống bao nhiêu: *df -h*

options:

-h : in kích thước mà người dùng có thể đọc

✓ **Xem thông tin model, serial,... phần cứng:** *dmidecode*

options:

-t : type

<i>Type Information</i>	<i>22 Portable Battery</i>
<i>0 BIOS</i>	<i>23 System Reset</i>
<i>1 System</i>	<i>24 Hardware Security</i>
<i>2 Baseboard</i>	<i>25 System Power Controls</i>
<i>3 Chassis</i>	<i>26 Voltage Probe</i>
<i>4 Processor</i>	<i>27 Cooling Device</i>
<i>5 Memory Controller</i>	<i>28 Temperature Probe</i>
<i>6 Memory Module</i>	<i>29 Electrical Current Probe</i>
<i>7 Cache</i>	<i>30 Out-of-band Remote Access</i>
<i>8 Port Connector</i>	<i>31 Boot Integrity Services</i>
<i>9 System Slots</i>	<i>32 System Boot</i>
<i>10 On Board Devices</i>	<i>33 64-bit Memory Error</i>
<i>11 OEM Strings</i>	<i>34 Management Device</i>
<i>12 System Configuration Options</i>	<i>35 Management Device Component</i>
<i>13 BIOS Language</i>	<i>36 Management Device Threshold</i>
<i>14 Group Associations</i>	<i>Data</i>
<i>15 System Event Log</i>	<i>37 Memory Channel</i>

16 <i>Physical Memory Array</i>	38 <i>IPMI Device</i>
17 <i>Memory Device</i>	39 <i>Power Supply</i>
18 <i>32-bit Memory Error</i>	40 <i>Additional Information</i>
19 <i>Memory Array Mapped Address</i>	41 <i>Onboard Devices Extended</i>
20 <i>Memory Device Mapped Address</i>	<i>Information</i> 42 <i>Management</i>
21 <i>Built-in Pointing Device</i>	<i>Controller Host Interface</i>

Ví dụ: `dmidecode -t 1`

(xem thông tin system)

- ✓ **Lệnh xem các tiến trình:** *top*
- ✓ **Lệnh xem dung lượng của thư mục:** *du*

options:

-s : xuất kết quả theo summarize (tổng dung lượng)

-h : in kích thước mà người dùng có thể đọc

Ví dụ: Xem dung lượng của thư mục /etc

`du -sh /etc`

- ✓ **Lệnh xem tên server:** *hostname*
- ✓ **Lệnh xem địa chỉ ip:** *ifconfig*

III. Các lệnh quản lý file và thư mục

- ✓ **Lệnh ls:** dùng để xem (liệt kê) nội dung thư mục

Cấu trúc lệnh:

```
ls [options] [Path]
```

Options:

-l : liệt kê chi tiết.

-a : liệt kê tất cả các file ẩn

Ý nghĩa các cột trong option -l của lệnh ls

1 st Column	File type and access permissions
2 nd Column	# of HardLinks to the File
3 rd Column	Owner and the creator of the file
4 th Column	Group of the owner
5 th Column	File size in Bytes
6 th Column	Date and Time
7 th Column	Directory or File name

Ví dụ: `ls -l /etc` : Liệt kê nội dung thư mục etc

`ls` : Liệt kê nội dung thư mục hiện hành

- ✓ **Lệnh cd:** dùng để chuyển thư mục

Cấu trúc lệnh::

`cd [Path]`

Ví dụ: `cd /etc` Chuyển đến thư mục /etc.

`cd usr` Chuyển vào thư mục usr là con của thư mục hiện hành.

`cd ..` Chuyển lên thư mục cấp cao hơn (cha)

`cd` Chuyển về thư mục home

`cd ~` Chuyển về thư mục home

- ✓ **Lệnh pwd:** cho biết thư mục hiện hành

Cấu trúc lệnh:

`pwd`

`[student]$ pwd`
`/home/student`

Ví dụ:

- ✓ **Lệnh passwd:** đổi mật khẩu đăng nhập của user đang login.

Cấu trúc lệnh:

`passwd`

Ví dụ:

```
[student]$ passwd
Changing password for student
(current) UNIX password:
New password:
Retype new password:
```

Mật khẩu phân biệt HOA – thường. user “root” có quyền thay đổi cho user bất kỳ

```
[root]# passwd student
```

- ✓ **Lệnh cp:** dùng để sao chép file

Cấu trúc lệnh:

```
cp [Options] Source Dest
```

Options:

-R, -r : Sao chép toàn bộ thư mục.

Source, Dest: Lần lượt là tên thư mục/tập tin nguồn, đích

Ví dụ:

Sao chép tập tin passwd vào thư mục hiện hành với cùng tên

```
#cp /etc/passwd passwd
```

Sao chép thư mục thư mục mydir_1 và đổi tên thành mydir_2 tại thư mục hiện hành

```
#cp -R mydir_1 mydir_2
```

- ✓ **Lệnh mv:** dùng để đổi tên / di chuyển thư mục hoặc file từ nơi này sang nơi khác

Cấu trúc lệnh:

```
mv [options] Source Dest
```

Options:

-i : Nhắc trước khi di chuyển với tập tin/thư mục đích đã có rồi.

-f: Ghi đè khi di chuyển với tập tin/thư mục đích đã có rồi.

Ví dụ:

Đổi tên thư mục **dir1** thành **dir2**:

```
#mv dir1 dir2
```

Di chuyển tập tin **myfile** vào thư mục mydir:

```
#mv myfile mydir
```

Di chuyển tập tin **myfile** vào thư mục **dir1** đồng thời đổi tên thành **newfile**:

```
#mv myfile dir1/newfile
```

- ✓ **Lệnh mkdir:** dùng để tạo thư mục

Cấu trúc lệnh: `mkdir [Options] Directory`

```
mkdir [options] directory
```

Options:

-p : Cho phép tạo thư mục con ngay cả khi chưa có thư mục cha.

Directory: Tên thư mục muốn tạo.

Ví dụ: Tạo thư mục `my_dir1`, `my_dir2`

```
#mkdir my_dir1 my_dir2
```

Tạo thư mục kể cả thư mục cha nếu chưa có

```
#mkdir -p dir3/dir4
```

- ✓ **Lệnh rmdir:** dùng để xóa thư mục rỗng. Thư mục rỗng là thư mục không chứa bất kỳ thành phần nào.

Cấu trúc lệnh:

```
rmdir [options] directory
```

Options:

-p : xóa thư mục và cả thư mục cha.

Directory: Tên thư mục muốn xóa.

Ví dụ: Xóa thư mục rỗng `my_dir1`, `my_dir2`

```
#rmdir my_dir1 my_dir2
```

Xóa thư mục `dir3/dir4` sau đó xóa `dir3`

```
#rmdir -p dir3/dir4
```

- ✓ **Lệnh rm:** dùng để xóa file/thư mục. Lệnh này được xem là một trong những lệnh nguy hiểm của Linux. Học viên cần phải chú ý khi sử dụng lệnh này.

Cấu trúc lệnh:

```
rm [options] file
```

Options:

-f: xóa không cần hỏi

-i: hỏi trước khi xóa

-R, -r: xóa toàn bộ thư mục, kể cả thư mục con

Mặc định tùy chọn -i được sử dụng

Ví dụ: Xóa tập tin myfile

```
#rm myfile
```

IV. Các lệnh hệ thống

✓ Lệnh shutdown:

Cấu trúc lệnh:

```
shutdown [option] [time] [wall]
```

Options:

- h: shutdown
- r: restart
- c: cancel pending shutdown

Time:

- now: thực hiện ngay lập tức
- hh:mm: ấn định thời gian
- +m: sau m phút sẽ thực hiện

Wall: Message thông báo.

Ví dụ: Thực hiện shutdown server sau 10 phút nữa với thông báo “Khởi động lại Server”

```
#shutdown -r +10 "Khởi động lại Server"
```

✓ Lệnh reboot: Khởi động lại Server

```
#reboot
```

✓ Lệnh init:

```
init [number]
```

Number:

- 3: restart
- 0: shutdown

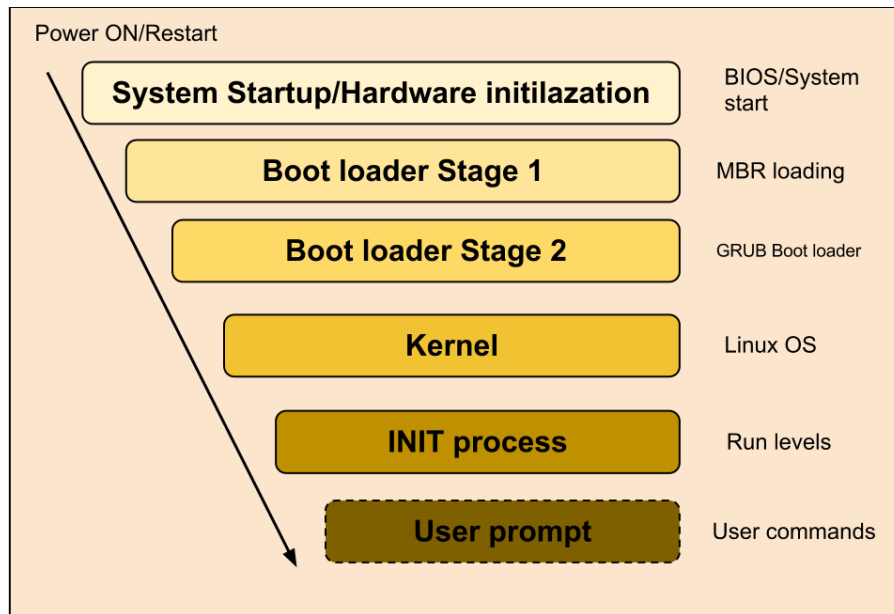
✓ Lệnh date: xem ngày giờ hệ thống

```
#date
```

Chương 5: BOOT AND RUN LEVELS

Trong chương này sẽ tìm hiểu quá trình khởi động của Server Linux, các level sử dụng và cách thiết lập run levels.

Quá trình khởi động



Quá trình khởi động của Linux sẽ thực hiện tuần tự các bước từ trên xuống dưới theo hình trên.

1) System Startup

Đây là bước đầu tiên của quá trình khởi động, ở bước này BIOS (**Basic Input/Output System**) thực hiện một công việc gọi là POST (Power-on Self-test). POST là một quá trình kiểm tra tính sẵn sàng phần cứng nhằm, kiểm tra thông số và trạng thái của các phần cứng máy tính như bộ nhớ, CPU, thiết bị lưu trữ, card mạng...

Nếu quá trình POST kết thúc thành công, BIOS sẽ cố gắng tìm kiếm và khởi chạy (boot) một hệ điều hành được chứa trong các thiết bị lưu trữ như ổ cứng, CD/DVD, USB....

Thông thường, BIOS sẽ kiểm tra ổ đĩa mềm, hoặc CD-ROM xem có thể khởi động từ chúng được không, rồi đến phân cứng. Thứ tự của việc kiểm tra các ổ đĩa phụ thuộc vào các cấu hình trong BIOS.

Nếu BIOS không tìm thấy boot device thì sẽ cảnh báo "**No boot device found**"

Nếu hệ điều hành Linux được cài trên ổ đĩa cứng thì nó sẽ tìm đến Master Boot Record (MBR) tại sector đầu tiên của ổ đĩa cứng đầu tiên.

2) MBR loading:

MBR (Master Boot Record) được lưu trữ tại Sector đầu tiên của một thiết bị lưu trữ dữ liệu, ví dụ /dev/hda hoặc /dev/dsa/. MBR rất nhỏ chỉ 512 byte.

MBR chứa thông tin:

- ✓ Primary boot loader code (446 Bytes): Cung cấp thông tin boot loader và vị trí boot loader trên ổ cứng
- ✓ Partition table information (64 Bytes): Lưu trữ thông tin của các partition
- ✓ Magic number (2 Bytes): được sử dụng để kiểm tra MBR, nếu MBR bị lỗi thì nó sẽ phục hồi lại.

3) Boot loader stage 2 (Grub Loader):

Sau khi xác vị trí BootLoader, bước này sẽ thực hiện loading BootLoader vào bộ nhớ và đọc thông tin cấu hình sau đó hiển thị GRUB boot menu để user lựa chọn. Nếu user không lựa chọn OS thì sau khoảng thời gian được định nghĩa GRUB sẽ load kernel default vào memory để khởi động.

4) Kernel:

Kernel của hệ điều hành sẽ được nạp vào trong RAM. Khi kernel hoạt động thì việc đầu tiên đó là thực thi quá trình INIT

5) INIT:

Đây là giai đoạn chính của quá trình BOOT. Quá trình này bắt đầu bằng việc đọc file /etc/inittab để xác định run-level. Sau đó sẽ thực thi các script tương ứng với run-level.

6) User prompt:

Người đăng nhập và sử dụng

7) Run Levels:

Run levels là chế độ sử dụng của Server. Mỗi chế độ có những module, chức năng hoạt động riêng. Trong chương trình LP1 các học viên chỉ cần chú ý đến 2 chế độ:

Multi-user.target: Chế độ dòng lệnh Command Mode(non-graphics) User chỉ sử dụng các lệnh (command) để thao tác. Ở chế độ này server sử dụng rất ít RAM.

Graphical.target: Chế độ đồ họa, mặc định khi Install OS ở chế độ GNOME là ta đang dùng Graphical.target.

Các lệnh thiết lập run levels:

- Thiết lập Multi-user.target mặc định khi khởi động:

```
#systemctl set-default multi-user.target
```

- Thiết lập Graphical.target mặc định khi khởi động:

```
#systemctl set-default graphical.target
```

- Chuyển đổi các levels:

Từ graphical sang command mode:

```
# systemctl isolate multi-user.target
```

Từ command mode sang graphical:

```
#systemctl isolate graphical.target
```

Chương 6: QUẢN TRỊ USER, GROUP

Chương này sẽ giới thiệu cấu trúc các file chứa các thông tin user và group. Giới thiệu các câu lệnh trong việc quản lý user, group với các options thông dụng, các options khác học viên sử dụng lệnh trợ giúp (*man*) để xem cách sử dụng.

I. Quản trị User:

Trên linux có hai loại tài khoản user đó là: tài khoản user hệ thống và tài khoản user người dùng.

User hệ thống: dùng để thực thi các module, script cần thiết phục vụ cho HĐH.

User người dùng: là những tài khoản để login để sử dụng HĐH. Trong các tài khoản user thì tài khoản user root (superuser) là tài khoản quan trọng nhất. Tài khoản này được tự động tạo ra khi cài đặt linux. Tài khoản này không thể đổi tên hoặc xóa bỏ. User root còn được gọi là superuser vì có toàn quyền trên hệ thống. Chỉ làm việc với tài khoản user root khi muốn thực hiện công tác quản trị hệ thống, trong các trường hợp khác chỉ nên làm việc với tài khoản user bình thường.

Mỗi user có các đặc điểm sau:

- ✓ Tên mỗi user là duy nhất, chỉ có thể đặt tên chữ thường, chữ hoa.
- ✓ Mỗi user có một mã định danh duy nhất (uid).
- ✓ Mỗi user có thể thuộc về nhiều nhóm.
- ✓ Tài khoản superuser có uid = gid = 0.

1) File `/etc/passwd`:

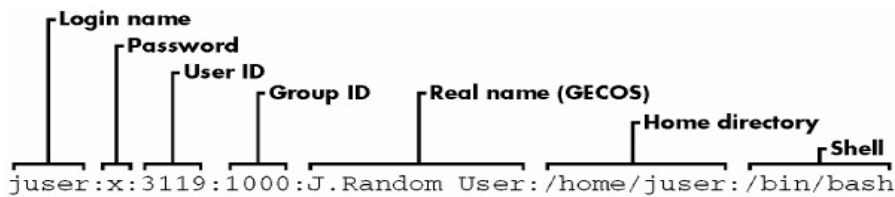
Là file văn bản chứa thông tin về các tài khoản user trên máy. Mọi user đều có thể đọc tập tin này nhưng chỉ có root mới có quyền thay đổi.

Để xem nội dung của file ta dùng lệnh:

```
#cat /etc/passwd
```

Cấu trúc của file gồm nhiều hàng, mỗi hàng là thông tin của 1 user. Dòng đầu tiên của tập tin mô tả thông tin cho user root (có ID = 0), tiếp theo là các tài khoản khác của hệ

thống, cuối cùng là các tài khoản người dùng thường. Mỗi hàng được chia làm 7 cột cách nhau bằng dấu “:”.



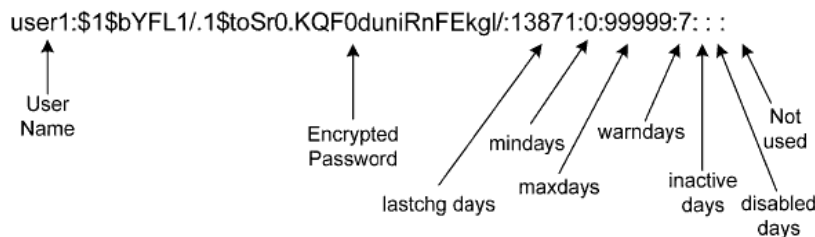
Ý nghĩa của các cột:

Cột 1	Tên người sử dụng
Cột 2	Mã liên quan đến mật khẩu cho Unix chuẩn và ‘x’ đối với Linux. Linux lưu mã này trong một tập tin khác <code>/etc/shadow</code> mà chỉ có root mới có quyền đọc
Cột 3	user ID
Cột 4	group ID
Cột 5	Tên mô tả người sử dụng (Comment)
Cột 6	Thư mục home của user. Thường sẽ nằm trong <code>/home/”tên_tài_khoản”</code>
Cột 7	Shell sẽ hoạt động sau khi user login, thường là <code>/bin/bash</code>

2) File `/etc/shadow`

Là tập tin văn bản chứa thông tin về mật khẩu của các tài khoản user trên máy. Chỉ có root mới có quyền đọc tập tin này. User root có quyền reset mật khẩu của bất kỳ user nào trên máy.

Mỗi dòng trong tập tin chứa thông tin về mật khẩu của user, định dạng của dòng gồm nhiều cột giá trị, dấu “:” được sử dụng để phân cách các cột.



Ý nghĩa các cột giá trị như sau:

Cột 1	Tên người sử dụng, tên này cũng giống với tên trong /etc/passwd
Cột 2	Mật khẩu đã được mã hóa. Để trống – không có mật khẩu, Dấu “*” – tài khoản bị tạm ngưng (disable).
Cột 3	Số ngày kể từ lần cuối thay đổi mật khẩu (tính từ 1/1/1970).
Cột 4	Số ngày trước khi có thể thay đổi mật khẩu, giá trị 0 có nghĩa có thể thay đổi bất kỳ lúc nào.
Cột 5	Số ngày mật khẩu có giá trị. 99999 có ý nghĩa mật khẩu có giá trị vô thời hạn.
Cột 6	Số ngày cảnh báo user trước khi mật khẩu hết hạn
Cột 7	Số ngày sau khi mật khẩu hết hạn tài khoản sẽ bị xóa. Thường có giá trị 7 (một tuần).
Cột 8	Số ngày kể từ khi tài khoản bị khóa (tính từ 1/1/1970).

3) Các lệnh quản lý user

- ✓ **Lệnh useradd:** Tạo tài khoản user

Cấu trúc lệnh:

```
useradd [Options] login_name
```

options:

- c: comment, tạo bí danh.
- u: set user ID. Mặc định sẽ lấy số ID tiếp theo để gán cho user.
- d: chỉ định thư mục home.
- g: chỉ định nhóm chính.
- G: chỉ định nhóm phụ (nhóm mở rộng).
- s: chỉ định shell cho user sử dụng.

Ví dụ 1: Tạo user với tên mary và tên đầy đủ Mary Smith (tham số -c)

```
#useradd -c "Mary Smith" mary
```

```
#passwd mary
```

```
New UNIX password: *****
```

```
Retype new UNIX password: *****
```

Giải thích: User được tạo sẽ thuộc về nhóm mary và thư mục của user là /home/mary được tạo ra tự động

Ví dụ 2: Tạo user với tên mary và tên đầy đủ Mary Smith (tham số -c), user thuộc về nhóm users và các nhóm wheel, sales.

```
#useradd -g users -G wheel,sales -c "Mary Smith" mary
```

```
$passwd mary
```

```
New UNIX password: *****
```

```
Retype new UNIX password: *****
```

✓ **Lệnh usermod:** Sửa thông tin tài khoản

Cấu trúc lệnh:

```
usermod [Options] login_name
```

Options:

-c: comment, tạo bí danh.

-l -d: thay đổi thư mục home.

-g: chỉ định nhóm chính.

-G: chỉ định nhóm phụ (nhóm mở rộng).

-s: chỉ định shell cho user sử dụng.

-L: Lock account

Ví dụ: Đổi tên tài khoản mary thành Jenny (tham số -l) với thư mục của user là /home/jenny (tham số -d)

```
#usermod -l jenny -c "Jenny Barnes" -m -d /home/jenny mary
```

✓ **Lệnh userdel:** Xóa tài khoản user

Cấu trúc lệnh:

```
userdel [Options] login_name
```

Options:

-r: xóa thư mục home của user

Ví dụ: Xóa tài khoản user mary.

#userdel mary

Thư mục home của user không bị xóa khi sử dụng lệnh userdel, để xóa cả thư mục home của user, sử dụng tham số -r.

Ví dụ: Xóa tài khoản user mary và thư mục home của user.

#userdel -r mary

Khi xóa tài khoản user bằng lệnh userdel, dòng mô tả tương ứng của user trong các tập tin /etc/passwd và /etc/shadow cũng bị xóa.

- ✓ **Lệnh chage:** Dùng để thiết lập các chính sách (policy) cho user.

Cấu trúc lệnh:

chage [options] login_name

Options:

-l : xem chính sách của 1 user

-E: thiết lập ngày hết hạn cho account. Vd: chage -E 6/30/2014 a1

-I: thiết lập số ngày bị khóa sau khi hết hạn mật khẩu

-m: thiết lập số ngày tối thiểu được phép thay đổi password

-M: thiết lập số ngày tối đa được phép thay đổi password

-W: Thiết lập số ngày cảnh báo trước khi hết hạn mật khẩu.

Ví dụ: chage -E 2015-04-30 -m 5 -M 90 -I 30 -W 14 newstar1

Lệnh trên sẽ thiết lập mật khẩu hết hạn vào ngày 30/04/2015. Ngoài ra, số ngày tối thiểu/tối đa giữa các lần thay đổi mật khẩu được thiết lập để 5 và 90 tương ứng. Các tài khoản sẽ bị khóa 30 ngày sau khi mật khẩu hết hạn, và một tin nhắn cảnh báo sẽ được gửi ra 14 ngày trước khi hết hạn mật khẩu.

II. Quản trị Group:

Nhóm là tập hợp của nhiều user. Mỗi nhóm có tên duy nhất, và có một mã định danh duy nhất (gid). Khi tạo một user (không dùng option -g) thì mặc định một group được tạo ra.

1) File /etc/group:

Là tập tin văn bản chứa thông tin về nhóm user trên máy. Mọi user đều có thể đọc tập tin này nhưng chỉ có root mới có quyền thay đổi.



Mỗi dòng trong tập tin chứa thông tin về các nhóm user trên máy, định dạng của dòng gồm nhiều cột giá trị, dấu “:” được sử dụng để phân cách các cột.

Ý nghĩa các cột giá trị như sau:

Cột 1	Tên nhóm
Cột 2	Mật khẩu đã được mã hóa. <ul style="list-style-type: none">- Để trống “ ”: không có mật khẩu,- Dấu “*”: tài khoản bị tạm ngưng (disable).
Cột 3	Mã nhóm (gid)
Cột 4	Danh sách các user thuộc nhóm

2) Các lệnh quản lý group

✓ **Lệnh groupadd:** Tạo nhóm

Cấu trúc lệnh:

```
groupadd [Options] group
```

Options:

- g GID: Định nghĩa nhóm với mã nhóm GID

Group: Tên nhóm định nghĩa

Ví dụ: Tạo nhóm users

```
#groupadd users
```


Tạo nhóm accounting với GID = 200

```
#groupadd -g 200 accounting
```

- ✓ **Lệnh groupmod:** Sửa thông tin nhóm

Cấu trúc lệnh:

```
groupmod [options] group
```

Options:

- g GID: Sửa mã nhóm thành GID
- n group_name: Sửa tên nhóm thành group_name

Group: Tên nhóm cần chỉnh sửa.

Ví dụ: Sửa gid của nhóm users thành 201

```
#groupmod -g 201 users
```

Đổi tên nhóm accounting thành accountant

```
#groupmod -n accountant accounting
```

- ✓ **Lệnh groupdel:** dùng để xóa nhóm

Cấu trúc lệnh:

```
groupdel group
```

Ví dụ: xóa nhóm testgroup

```
#groupdel testgroup
```

Chương 7: QUYỀN TRÊN HỆ THỐNG TẬP TIN, THƯ MỤC

Phân quyền truy xuất đến các tài nguyên Server Linux là một vấn đề quan trọng. Phân quyền giúp tăng mức độ an toàn, đảm bảo đúng trách nhiệm – quyền hạn của từng user khi sử dụng tài nguyên trên Server. Chương này sẽ giới thiệu các quyền truy cập, cách thiết lập quyền trên hệ thống tập tin và thư mục.

1) Quyền truy xuất:

Quyền truy xuất trên thư mục và tập tin được trình bày khi thực hiện lệnh `ls -l`

Ví dụ: `#ls -l`

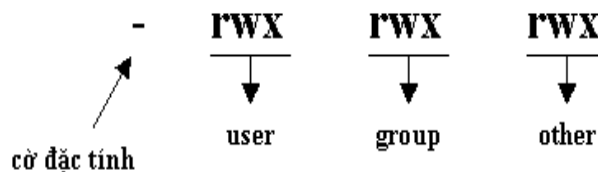
Kết quả

```
drwxr-xr-x 2      john  admin 30196 Jun  4 19:50 application
-rwxr-x--x 1      john  admin  155 Jul  7 14:45 hello
-rwxr-xr-x 1      john  admin   78 Mar 15  8:45 read_me
```

Danh sách quyền truy xuất trình bày ở cột đầu tiên trong kết quả. Các loại quyền truy xuất gồm:

- ✓ **Đọc (read):** Cho phép đọc nội dung tập tin và xem nội dung thư mục bằng lệnh `ls`.
- ✓ **Ghi (write):** Cho phép thay đổi nội dung hoặc xóa tập tin. Đối với thư mục, quyền này cho phép tạo, xóa hoặc đổi tên tập tin mà không phụ thuộc vào quyền sở hữu trên tập tin chứa trong thư mục.
- ✓ **Thực thi (execute):** Cho phép thực thi chương trình, đối với thư mục, quyền này cho phép chuyển vào thư mục bằng lệnh `cd`.

Quyền truy xuất gồm 3 nhóm :



Kí hiệu	Kiểu file
-	Regular file
d	Directory
l	Symbolic link
b	Block special file
c	Character special file
p	Named pipe
s	Socket

- ✓ Quyền của người sở hữu (owner hoặc user) ký hiệu bằng ký tự u: Người tạo ra thư mục/tập tin hoặc được gán quyền sở hữu.
- ✓ Quyền của nhóm (group) ký hiệu bằng ký tự g: Nhóm người sử dụng được gán quyền
- ✓ Quyền của những người dùng khác (others) ký hiệu bằng ký tự o: Là những người sử dụng khác không thuộc về 2 loại trên.

2) Biểu diễn quyền truy xuất

Biểu diễn quyền truy xuất theo 2 cách

- ✓ **Bằng chữ:** Trong cách biểu diễn này, quyền truy xuất được viết bằng các ký tự:

r : read
w : write
x : execute
- : không có quyền

Ví dụ:

- rwX :Có toàn quyền
- r-- :Chỉ có quyền đọc
- rw- :Chỉ có quyền đọc và ghi
- :Không có quyền gì

Quyền hạn trên 1 file sẽ gồm cả 3 nhóm quyền (owner, group, others) nên danh sách quyền sẽ gồm 9 ký tự.

Ví dụ:

rwXrw---	Người sở hữu có toàn quyền, các user cùng nhóm chỉ có quyền đọc/ghi còn mọi người khác không có quyền truy xuất
rw-r-----	Người sở hữu có quyền đọc/ghi, các user cùng nhóm chỉ có quyền đọc còn mọi người khác không có quyền truy xuất
rwXr-xr--	Người sở hữu có toàn quyền, các user cùng nhóm chỉ có quyền đọc và thực thi chương trình còn mọi người khác chỉ có quyền đọc

✓ **Bảng số:** Trong cách biểu diễn này, mỗi quyền được gán cho một giá trị số theo bảng sau:

Quyền	Giá trị
r	4
w	2
x	1

Mỗi nhóm quyền truy xuất là tổng của các loại quyền trên.

Ví dụ:

Quyền	Ý nghĩa	Biểu diễn bằng số
rwx	Có toàn quyền	7
r--	Chỉ có quyền đọc	4
rw	Chỉ có quyền đọc và ghi	6
---	Không có quyền gì	0

Vì quyền thực sự gồm cả 3 nhóm quyền (owner, group, others) nên danh sách quyền biểu diễn dạng số sẽ gồm 3 chữ số.

Dạng chữ	Dạng số	Ý nghĩa
rwxrw----	760	Người sở hữu có toàn quyền, các user cùng nhóm chỉ có quyền đọc/ghi còn mọi người khác không có quyền truy xuất
rw-r--r--	644	Người sở hữu có quyền đọc/ghi, các user cùng nhóm chỉ có quyền đọc còn mọi người khác không có quyền truy xuất
rwxr-xr--	754	Người sở hữu có toàn quyền, các user cùng nhóm chỉ có quyền đọc và thực thi chương trình còn mọi người khác chỉ có quyền đọc

Ví dụ:

Lưu ý: Người sử dụng có quyền đọc thì có quyền sao chép tập tin và tập tin sau khi sao chép sẽ thuộc sở hữu người thực hiện sao chép

3) Các lệnh về quyền:

✓ **Lệnh chmod:** Thay đổi quyền truy xuất trên thư mục/tập tin

Cấu trúc lệnh:

```
chmod [Options] Mode file
```

Options:

-R: Áp dụng đối với thư mục làm cho lệnh chmod có tác dụng trên cả các thư mục con (đệ quy).

Mode: Quyền truy xuất mới trên tập tin

Quyền truy xuất mới có thể gán cho từng nhóm quyền bằng cách sử dụng ký tự u đại diện cho quyền của người sở hữu (owner), g đại diện cho quyền của nhóm (group) và o đại diện cho quyền của mọi người dùng khác (others). Ký tự “+” có ý nghĩa gán thêm quyền, “-” có ý nghĩa rút bớt quyền và “=” có nghĩa là gán.

Ví dụ 1:

g+w	thêm quyền ghi cho nhóm
o-rwx	loại bỏ tất cả các quyền của mọi người dùng khác
u+x	thêm quyền thực thi cho người sở hữu
+x	thêm quyền thực thi cho tất cả (mọi người)
a+rw	thêm quyền ghi đọc cho tất cả
ug+r	thêm quyền đọc cho owner và group
o=x	chỉ cho phép thực thi với mọi người

Ví dụ 2:

```

[linux@Fedora linux]$ chmod a-x passwd
[linux@Fedora linux]$ ls -l
total 4
-rw-rw-r-- 1 linux  linux  1576 May  1 02:40 passwd
[linux@Fedora linux]$ chmod ug+x passwd
[linux@Fedora linux]$ ls -l
total 4
-rwxrwxr-- 1 linux  linux  1576 May  1 02:40 passwd
[linux@Fedora linux]$ chmod o-rwx passwd
[linux@Fedora linux]$ ls -l
total 4
-rwxrwx--- 1 linux  linux  1576 May  1 02:40 passwd
[linux@Fedora linux]$

```

- ✓ **Lệnh chown:** Thay đổi người sở hữu thư mục/tập tin.

Cấu trúc lệnh:

```
chown [Options] Owner file
```

Options:

-R: Áp dụng đối với thư mục làm cho lệnh chmod có tác dụng trên cả các thư mục con (đệ quy).

Owner: Người sở hữu mới trên tập tin.

Ví dụ:

```

[root@Fedora linux]# ls -l
total 8
-rw-rw-r-- 1 linux  linux  6 May  1 03:26 hello
-rw-r--r-- 1 linux  linux 1617 May  1 03:22 passwd
[root@Fedora linux]# chown fedora hello
[root@Fedora linux]# ls -l
total 8
-rw-rw-r-- 1 fedora  linux  6 May  1 03:26 hello
-rw-r--r-- 1 linux  linux 1617 May  1 03:22 passwd
[root@Fedora linux]#

```

- ✓ **Lệnh chgrp:** Thay đổi nhóm sở hữu thư mục/tập tin.

Cấu trúc lệnh:

```
chgrp [Options] Group file
```

Options:

-R: Áp dụng đối với thư mục làm cho lệnh chmod có tác dụng trên cả các thư mục con (đệ quy).

Group: Nhóm sở hữu mới trên tập tin.

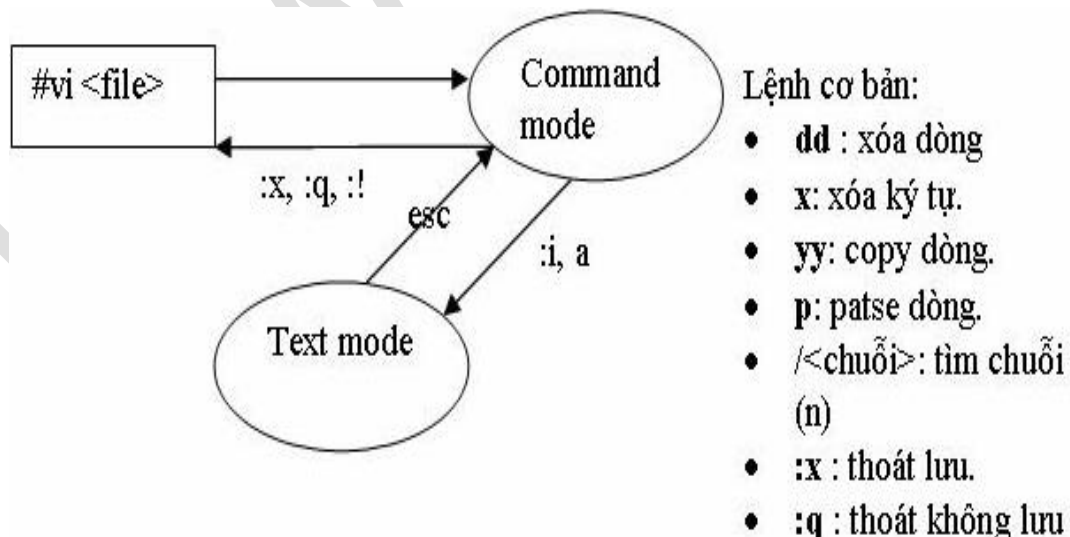
Ví dụ:

```
[root@Fedora linux]# ls -l
total 8
-rw-rw-r--  1 fedora  linux    6 May  1 03:26 hello
-rw-r--r--  1 linux   linux  1617 May  1 03:22 passwd
[root@Fedora linux]# chgrp fedora hello
[root@Fedora linux]# ls -l
total 8
-rw-rw-r--  1 fedora  fedora    6 May  1 03:26 hello
-rw-r--r--  1 linux   linux  1617 May  1 03:22 passwd
[root@Fedora linux]#
```

Chương 8: TRÌNH SOẠN THẢO VI & CÁC LỆNH LỌC DỮ LIỆU

Vi là trình soạn thảo có mặt trên các hệ điều hành Unix/Linux. Vi thường được sử dụng trong việc tạo và chỉnh sửa các file cấu hình trên Linux. Biết cách sử dụng trình soạn thảo Vi sẽ giúp ích rất nhiều cho người quản trị trong việc vận hành và phát triển dịch vụ trên Linux.

1) Các chế độ làm việc của Vi



Để Tạo hoặc chỉnh sửa file ta thực hiện lệnh vi theo cấu trúc lệnh như sau:

```
#vi <file>
```


Khi thực hiện lệnh này ta sẽ vào chế độ **Command mode**. Ở chế độ này ta sẽ sử dụng các lệnh để làm việc. Nếu muốn chỉnh sửa văn bản ta phải chuyển sang chế độ **Text mode**.

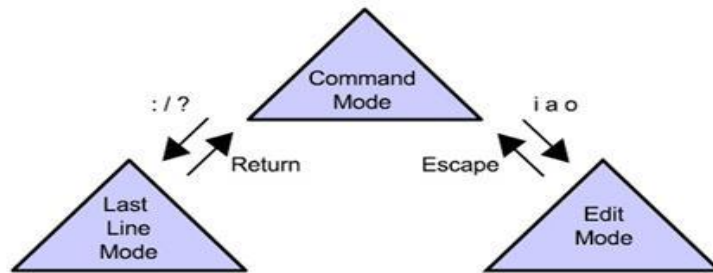
Để chuyển từ chế độ **Command mode** sang **Text mode** ta bấm phím “i” hoặc “a” hoặc “insert”.

Để chuyển từ chế độ **Text mode** sang **Command mode** ta bấm phím “esc”

Để thoát khỏi vi ta phải ở chế độ **command mode** và thực hiện lệnh thoát:

- Thoát và không lưu các thay đổi ta dùng lệnh: “:q!”
- Thoát và lưu: “:x” hoặc “:wq”

2) Chế độ command (command mode)



Search Function

/exp Go forward to exp
 ?exp Go backward to exp

Move and Insert Text

:3,8d Delete line 3-8
 :4,9m 12 Move lines 4-9 to 12
 :2,5t 13 Copy lines 2-5 to 13
 :5,9w file Write lines 5-9 to file

Save Files and Exit

:w Write to disk
 :w newfile Write to newfile
 :w! file Write absolutely
 :wq Write and quit
 :q Quit editor
 :q! Quit and discard
 :e! Re-edit current file,
 Discard buffer

Control Edit Session

:set nu Display line number
 :set nonu Turn off line number
 :set all Show all settings
 :set list Display invisible
 Characters
 :set wm=5 Wrap lines 5 spaces
 From right margin

Screen/Line Movement

j Move cursor down
 k Move cursor up
 h Move cursor left
 l Move cursor right
 o Go to line start (zero)
 \$ Go to line end
 G Go to last file line

Word Movement

w Go forward 1 word
 b Go backward 1 word

Search Function

n Repeat previous search
 N Reverse previous search

Delete Text

x Delete 1 character
 dw Delete 1 word
 dd Delete 1 line
 D Delete to end of line
 d0 Delete to start of line
 dG Delete to end of file

Cancel Edit Function

u Undo last change
 . Do last change again

Add/Append Text

a Append after cursor
 A Append at line end
 i Insert before cursor
 Si Insert text 5 times
 I Insert at beginning of line

Add New Lines

o Open a line below cursor
 O Open a line above cursor

Change Text

cw Change a word
 3cw Change 3 words
 C Change line
 r Replace one character
 R Replace/type over a line

Copy and Insert Text

Y Yank a copy lines
 5Y Yank a copy of 5
 p Put below cursor
 P Put above cursor



Như ta thấy trên hình, có rất nhiều lệnh trong chế độ command, việc ghi nhớ các lệnh này sẽ rất khó khăn. Chúng ta sẽ chỉ cần nhớ một vài lệnh cơ bản sau:

- “:set nu” : hiển thị số dòng của file (set number).
- “:set nonu: không hiển thị số dòng của file.
- “dd”: Xóa dòng ngay vị trí con trỏ.
- dG: Xóa từ vị trí con trỏ đến cuối file (G: tổ hợp phím shift + g).
- Sử dụng các phím mũi tên để di chuyển con trỏ chuột.
- “:n”: di chuyển con trỏ đến dòng số n
Ví dụ: di chuyển con trỏ đến dòng số 10 ta đánh lệnh :10
- “u”: undo
- “/text”: Tìm kiếm từ text trong file, dùng phím “n” (next) để xem kết quả kế tiếp.

3) Đường ống pipe

Ý nghĩa: Sử dụng đầu ra của lệnh này làm đầu vào của lệnh khác. Sử dụng đường ống pipe sẽ giúp kết hợp các lệnh 1 cách linh hoạt và mang lại nhiều lợi ích cho việc lọc dữ liệu kết quả.

Cấu trúc lệnh:

```
command 1 | command 2 |
```

Ví dụ: Liệt kê chi tiết nội dung thư mục /etc/ và xem theo dạng từng trang 1

```
# ls -al / | more
```

Có thể nhóm nhiều lệnh bằng dấu ()

```
# (cal 2009;cal 2010)/more
```

4) Chuyển hướng nhập/xuất

Mục đích: dùng để đưa nội dung từ 1 file sang file khác; Nhập nội dung từ màn hình sang 1 file; Xuất nội dung 1 file ra màn hình.

Dấu ‘<’ : Dùng cho chuyển hướng nhập.

Dấu ‘>’: Dùng cho chuyển hướng xuất.

Ví dụ: Nhập nội từ màn hình shell vào file bai1 (bai1 lưu tại thư mục hiện hành)

```
# cat > bai1
```

```
Chao cac ban
```

Copy nội dung trong file bai1 sang file bai2

```
# cat bai1 > bai2
```

```
# tr 'A-Z' 'a-z' < .bash_profile
```

5) Các lệnh tìm kiếm

- Lệnh **grep**: Tìm theo keyword trong 1 file

Cấu trúc lệnh:

```
grep [options] 'string' file
```

Options thông dụng:

- c: đếm số lần xuất hiện của string (count)
- n: hiển thị số dòng của từ tìm kiếm trong file

Ví dụ: tìm kiếm từ **root** trong file **passwd**

```
#grep 'root' /etc/passwd
```

- Lệnh **find**: Là lệnh tìm kiếm mạnh nhất với nhiều tham số nhưng thường tốn thời gian hơn các lệnh khác.

Cấu trúc lệnh:

```
find [vị trí] [tiêu chuẩn tìm]
```

Tiêu chuẩn tìm kiếm	Tùy chọn
Tìm theo tên	-name pattern
Tìm theo timestamp	-atime, -mtime, -ctime
Tìm theo quyền	-perm mode
Tìm theo kích thước	-size n
Tìm theo group	-gid GID
Tìm theo user ID	-uid UID
Tìm theo cấp của thư mục con	-maxdepth levels

Ví dụ: tìm kiếm file có tên passwd:

```
#find / -name 'passwd'
```

6) Lệnh cut

Ý nghĩa: dùng để lọc dữ liệu.

Cấu trúc lệnh:

```
cut [options] file
```

Options thông dụng:

-d: Dấu hiệu nhận biết sự ngăn cách giữa các cột

-f: vị trí của trường muốn lấy.

Ví dụ: Lấy cột thứ 1 trong /etc/passwd

```
# cut -d: -f 1 /etc/passwd
```

Lấy cột thứ 7 của file /etc/passwd có chứa từ root

```
# grep root /etc/passwd|cut -d: -f7
```

Lấy 10 kí tự đầu tiên ở 5 dòng cuối file /etc/passwd

```
# tail -5 /etc/passwd|cut -c1-10
```

Chương 9: NETWORK CONFIGURATION

1) Các file cấu hình network

- ✓ */etc/hosts*: Dùng phân giải hostname không thể phân giải được. Có thể dùng thay DNS trong hệ thống mạng LAN

```
127.0.0.1          localhost.localdomain
::1               localhost.localdomain
```

- ✓ */etc/resolv.conf*: Chỉ định DNS server để phân giải tên miền.
- ✓ */etc/sysconfig/network-scripts/ifcfg-<tên card mạng>*: trong thư mục này chứa thông tin cấu hình của từng card mạng.

Ví dụ: eno16777736

Định dạng tên card mạng:

```
Two character prefixes based on the type of interface:
en -- ethernet
sl -- serial line IP (slip)
wl -- wlan
ww -- wwan

Type of names:
b<number>                -- BCMA bus core number
ccw<name>                -- CCW bus group name
o<index>                 -- on-board device index number
s<slot>[f<function>][d<dev_port>] -- hotplug slot index number
x<MAC>                   -- MAC address
[P<domain>]p<bus>s<slot>[f<function>][d<dev_port>] -- PCI geographical location
[P<domain>]p<bus>s<slot>[f<function>][u<port>][..]1[i<interface>] -- USB port number chain
```

Đối với Centos 6.x tên card mạng là ethx (x: là chỉ số card mạng). Đây là cách định dạng tên để nhớ.

2) Các lệnh network

- Xem địa chỉ ip:
#ifconfig
- Tắt mở card mạng:
#ifup têncard: bật card mạng
#ifdown têncard: tắt card mạng
- Khởi động dịch vụ:

#service network restart

Hoặc

#/etc/init.d/network restart

Hoặc

#systemctl restart network.service

- Xem thông tin gateway

#route

3) Cấu hình ip

a) Cấu hình bằng lệnh

#ifconfig <ten_card_mạng> <ip> netmask <netmask> up

Chú ý: IP sẽ mất mỗi khi tắt mở card mạng hay restart dịch vụ network. Đây còn gọi là cách đặt ip tạm thời, thường dùng để test.

b) Cấu hình bằng file

Thay đổi nội dung trong file cấu hình của card mạng

#vi etc/sysconfig/network-scripts/ifcfg-<tên card mạng>

Ý nghĩa các thông số cần chú ý trong file:

DEVICE	Tên của card mạng
ONBOOT	yes: Khởi động khi restart dịch vụ network no: không khởi động khi restart dịch vụ network
BOOTPROTO	static: dùng địa chỉ IP tĩnh dhcp: nhận IP từ DHCP server
IPADDR:	Địa chỉ IP của card mạng (dùng khi cấu hình ip tĩnh)
NETMASK	Địa chỉ subnet mask (dùng khi cấu hình ip tĩnh)
GATEWAY	Địa chỉ gateway (dùng khi cấu hình ip tĩnh)
DNS{1,2}:	Tên DNS server
USERCTL	yes: cho phép user thường cấu hình card mạng này no: không cho phép user thường cấu hình card mạng này

- **Cấu hình địa chỉ IP tĩnh:**

Ví dụ: thiết lập ip tĩnh cho Server với ip 10.10.10.10/24, gateway 10.10.10.254, DNS1=8.8.8.8

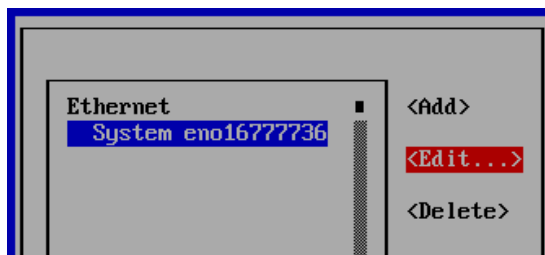
```
DEVICE=eno16777736
ONBOOT=yes
BOOTPROTO=static
IPADDR=10.10.10.10
NETMASK=255.255.255.0
GATEWAY=10.10.10.254
DNS1=8.8.8.8
USERCTL=no
```

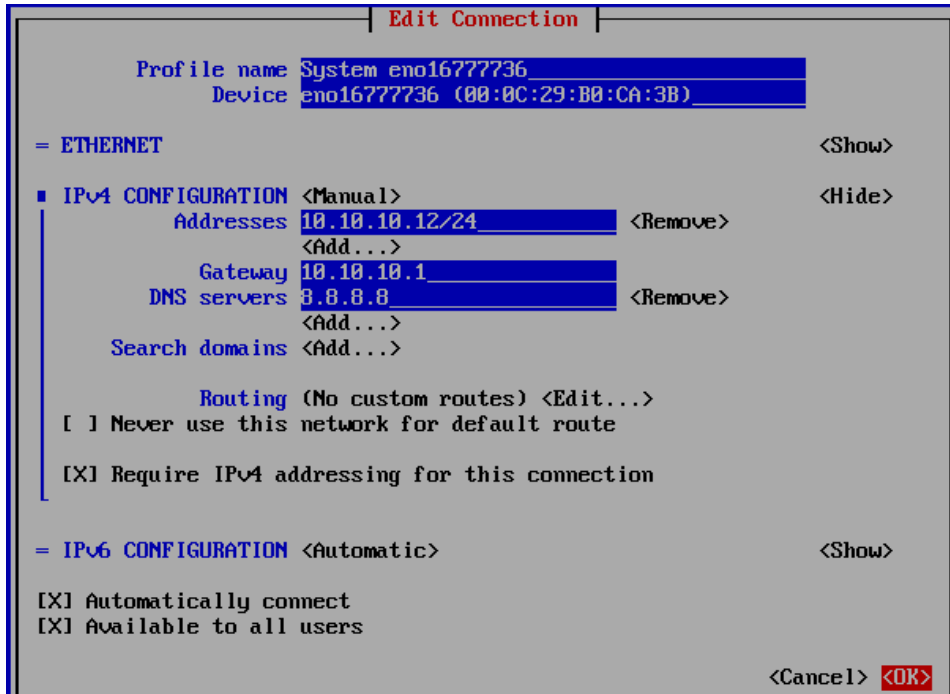
- **Cấu hình địa chỉ IP động**

```
DEVICE=eno16777736
ONBOOT=yes
BOOTPROTO=dhcp
USERCTL=no
```

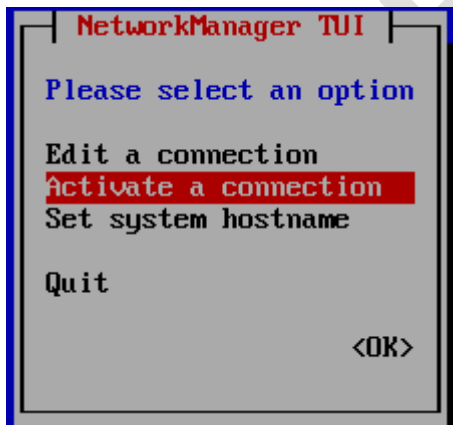
c) **Cấu hình ip bằng giao diện**

#nmtui





Sau khi chỉnh sửa xong ta thực hiện active card mạng



Hoặc

`#service network restart`

4) Thay đổi tên card mạng về dạng ethx

Bước 1: Kiểm tra thông tin interface hiện tại

`#ifconfig`

```
eno16777736: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>    mtu
1500
inet 192.168.35.51 netmask 255.255.255.0 broadcast 192.168.35.255
```



```
inet6 fe80::20c:29ff:feb0:ca3b prefixlen 64 scopeid 0x20<link>
ether 00:0c:29:b0:ca:3b txqueuelen 1000 (Ethernet)
RX packets 4964 bytes 547675 (534.8 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 409 bytes 36902 (36.0 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Bước 2: edit “/etc/sysconfig/grub

```
# vi /etc/sysconfig/grub
```

Chỉnh sửa tại dòng (thêm vào **net.ifnames=0 biosdevname=0**) (Predictable Network Interface Names)

```
GRUB_CMDLINE_LINUX="rd.lvm.lv=centos/swap vconsole.font=latarcyrheb-
sun16 rd.lvm.lv=centos/root crashkernel=auto vconsole.keymap=us rhgb quiet
net.ifnames=0 biosdevname=0"
```

Bước 3: biên dịch lại file grub để apply cấu hình mới

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

Bước 4: đổi tên file cấu hình card mạng

```
# cd /etc/sysconfig/network-scripts/
```

```
# mv ifcfg-eno16777736 ifcfg-eth0
```

```
# vi ifcfg-eth0
```

```
DEVICE=eth0
```

Bước 5: reboot Server và kiểm tra lại kết quả

```
# reboot
```

```
# ifconfig
```

Chương 10: INSTALL SOFTWARE

1) Các lệnh nén và giải nén

Cài đặt một chương trình lên Linux là công việc rất khó khăn so với môi trường Windows. Tùy thuộc vào các chương trình cài đặt mà độ phức tạp, thời gian cài đặt và cấu hình khác nhau. Các bước cài đặt một chương trình còn tùy thuộc vào Version của các Disktro.

Các chương trình cài đặt thường được nén lại và upload trên internet. Do vậy, các bạn phải biết cách giải nén và nén file. Trên Linux có nhiều định dạng file nén như: .gz, .tar, .tar.gz, bz2, .tar.bz2. Mỗi loại file nén đều có cách nén và giải nén khác nhau.

- ✓ Nén file “.gz”

```
#gzip /etc/passwd
```

- ✓ Giải nén “.gz”

```
#gzip -d /etc/passwd
```

```
#gunzip /etc/passwd
```

- ✓ Nén file “.tar”

```
#tar -cvf /home/adminbackup.tar /home/abc
```

```
#tar -czf /home/adminbackup.tar.gz /home/abc
```

- ✓ Giải nén file “.tar”

```
#tar -xvf /home/adminbackup.tar
```

```
#tar -xzf /home/adminbackup.tar.gz
```

- ✓ Nén file “.bz2”

```
#tar -cjf backup_eiciel.tar.bz2 eiciel-0.9.6.1
```

- ✓ Giải nén file “.bz2”

```
#tar -xjf eiciel-0.9.6.1.tar.bz2
```

2) Cài đặt từ rpm

RedHat Package Manager (RPM) là một hệ thống quản lý những package được Linux hỗ trợ cho người dùng.

Đặc tính của RPM :

- Khả năng nâng cấp phần mềm.
- Truy vấn thông tin hiệu quả.

Những package được đóng gói có dạng:

Tênpackage-phiên bản-số hiệu.kiến trúc.rpm

Ví dụ: foo-1.0-1.i386.rpm

Cú pháp rpm:

```
rpm -options <tênphầnmềm>
```

Options

- i: cài đặt*
- e: xóa*
- U: update*
- nodeps: không kiểm tra các gói phụ thuộc*
- qa: tìm phần mềm*
- ql: tìm nơi cài đặt*

Cài đặt phần mềm bằng rpm:

Cú pháp:

```
rpm -ivh package
```

Một số trường hợp lỗi

- Package đã cài rồi: package is already installed
- Xung đột tập tin: ...package...conflicts with...
- Phụ thuộc vào package khác : failed dependencies

Ví dụ : Bạn cài package foo-1.0-1.i386.rpm

```
#rpm -ivh foo-1.0-1.i386.rpm
```

```
foo package foo-1.0-1.i386.rpm is already installed
```

3) Cài đặt từ source

Bước 1: Giải nén file (tar -xzf file)

Bước 2: Đọc file hướng dẫn cài đặt (INSTALL, README) trong thư mục vừa giải nén.

Bước 3: Biên dịch và cài đặt

*./configure --option
make & make install*

4) Cài đặt từ yum

yum (yellowdog updater modified): Cài đặt, xóa, truy vấn các phần mềm từ các repositories trên internet hay local một cách tự động. với phương thức yum các phần mềm phụ thuộc sẽ tự động cài đặt, tự động cập nhật các phần mềm.

Chú ý: yum phải chạy dưới quyền của root user.

repository: Là nơi lưu trữ các phần mềm trên internet

Cú pháp:

yum options <tên/nhóm phần mềm>

Options:

- *Install: cài đặt phần mềm*
- *Remove: xóa phần mềm*
- *List installed: xem các phần đã cài*
- *Groupinstall: Cài 1 nhóm phần mềm*
- *Groupremove: Gỡ bỏ 1 nhóm phần mềm*
- *Clean: Xóa các cache, plugin, meta-data...*

Một số lỗi thường gặp:

```
Trying other mirror.  
http://mirrors.btte.net/centos/6.2/updates/i386/repodata/8b5041e3b52e06d1fc2243a82e7341e0c61943ad574863913edc3fa26611f8d6-primary.sqlite.bz2: [Errno 12] Timeout on http://mirrors.btte.net/centos/6.2/updates/i386/repodata/8b5041e3b52e06d1fc2243a82e7341e0c61943ad574863913edc3fa26611f8d6-primary.sqlite.bz2: (28, 'Operation too slow. Less than 1 bytes/sec transferred the last 30 seconds')  
Trying other mirror.
```



Đường truyền mạng yếu, đứt mạng
Không phân giải được tên miền
Tường lửa..

```
[root@server1 x-unikey-1.0.4]# ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for gawk... gawk
checking whether make sets $(MAKE)... no
checking build system type... i686-pc-linux-gnu
checking host system type... i686-pc-linux-gnu
checking for style of include used by make... none
checking for gcc... no
checking for cc... no
checking for cc... no
checking for cl... no
configure: error: no acceptable C compiler found in $PATH
See `config.log' for more details.
[root@server1 x-unikey-1.0.4]#
```



Thiếu bộ thư viện “Development Tools” và các trình biên dịch gcc...

newstar.vn

Chương 11: LINUX FILE SYSTEMS, DEVICES

1) File System

File system là những thành phần vô cùng quan trọng trong 1 hệ điều hành, được sử dụng để điều khiển cách dữ liệu lưu trữ và truy vấn... . Linux là hệ điều hành có khả năng hỗ trợ nhiều loại file system nhất hiện nay.

Các Kiểu File System trên Centos 7:

xfs: đây là file system mặc định trên centos 7. Được phát triển bởi Silicon Graphics từ năm 1993. Có các đặc điểm:



- Hạn chế được tình trạng phân mảnh dữ liệu.
- Hỗ trợ file system lên tới 6 exabyte tương đương 18 triệu terabyte.
- Hỗ trợ file lên đến 8 exabyte.
- Cấu trúc thư mục chứa hàng chục triệu đối tượng.
- Kích thước tối đa của xfs partition là 500TB.

ext4: file system này ra đời từ phiên bản 2.6.28 của nhân Linux (25/12/2008). Nó kế thừa và phát huy các điểm mạnh của ext3, đồng thời ext4 có thể giảm bớt hiện tượng phân mảnh dữ liệu trong ổ cứng, hỗ trợ file system và file có dung lượng lớn và tốc độ hoạt động nhanh. Hỗ trợ kích thước tối đa của 1 file system ext4 trên Centos 7 là 50TB.

Vfat: Đây là file system FAT trên Microsoft windows.

Swap: partition swap được sử dụng để hỗ trợ virtual memory. Trong trường hợp không đủ RAM để xử lý thì data sẽ lưu vào swap và xử lý. Swap còn gọi là không gian hoán đổi bộ nhớ vật lý. Kích thước để swap hoạt động hiệu quả nhất là bằng 2 lần bộ nhớ RAM.

BIOS Boot: được sử dụng cho việc khởi động một thiết bị với một GUI partition table (GPT). Kích thước của partition này rất nhỏ (khoảng 1 MB). Tuy nhiên bạn không cần phải tạo partition này nếu đã sử dụng Master Boot Record (MBR).

2) Kết nối các thiết bị (Devices)

Hệ điều hành Linux coi tất cả (các thiết bị, ổ đĩa) đều là các tệp tin (file). Trong phần này sẽ hướng dẫn các bạn kết nối các thiết bị thông dụng vào Centos 7 như: USB, CDROM.

Khi gắn các thiết bị usb hay cdrom vào server nếu trong chế độ đồ họa (graphic) thì Server sẽ tự động nhận và hiển thị lên Desktop. Trong trường hợp sử dụng chế độ command line (minimal) thì phải thực hiện việc **mount** các thiết bị đó vào một mount point (thư mục chứa).

Cấu trúc lệnh mount:

```
mount <ten_thiet_bi> <mount_point>
```

<ten_thiet_bi>: Nếu là cdrom thông thường sẽ là /dev/cdrom (hoặc cdrom1)

Nếu là usb thì cần xác định tên thông qua lệnh `#fdisk -l`

<mount_point>: Là vị trí gắn kết với thiết bị trên cây thư mục. Thư mục thường được sử dụng làm mount point là /media. Bạn có thể tạo ra 1 thư mục để làm mount point.

Sau khi thiết bị được mount, việc thao tác với usb tương ứng với thao tác trên thư mục mount point.

Chương 12: MỞ RỘNG HDD

Thiếu hụt dung lượng ổ đĩa thường xuyên xảy ra trên hệ thống của bạn, có nhiều nguyên nhân: có thể do mở rộng cơ sở dữ liệu lưu trữ, số lượng người dùng tăng lên hay số lượng các tác vụ trên hệ thống linux của bạn lớn hơn dự kiến ban đầu.

Có 2 cách để giải quyết:

- Thêm một ổ đĩa mới và di chuyển thư mục trên ổ đĩa bị full đến ổ cứng đó, sau đó liên kết cấu trúc thư mục trên hai ổ đĩa cùng nhau.
- Kết hợp các phân vùng lại với nhau bằng cách sử dụng Linux Logical Volume Manager (LVM)

1) Thêm ổ cứng vào Server

Các lệnh sử dụng:

- fdisk: để phân chia partition của ổ cứng
- mkfs: để format
- mount: để gắn một partition đã format vào một mount point.
- Chỉnh sửa fstab để Linux có thể tự động mount khi boot.

Cấu hình:

- ✓ **Bước 1:** Gắn ổ cứng mới vào Server Linux
- ✓ **Bước 2:** Kiểm tra bằng lệnh fdisk

```
[root@localhost ~]# fdisk -l
Disk /dev/sdb: 5368 MB, 5368709120 bytes
255 heads, 63 sectors/track, 652 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Disk /dev/sdb doesn't contain a valid partition table

Disk /dev/sda: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00070fa7

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1 *          1           64     512000    83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2            64        1306     9972736    8e  Linux LVM
```

Device	Capacity
Memory	300 MB
Processors	1
Hard Disk (SCSI)	10 GB
Hard Disk 2 (SCSI)	5 GB
CD/DVD (IDE)	Using file D:\Cai ...
Network Adapter	Custom (VMnet2)
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

- ✓ **Bước 3:** Định dạng HDD sdb

#fdisk /dev/sdb

```
[root@localhost ~]# fdisk /dev/sdb
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel with disk identifier 0x9caaca08.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
switch off the mode (command 'c') and change display units to
sectors (command 'u').

Command (m for help): █
```

Chọn m để xem menu

Bước 4: Chọn n để thêm partition mới

```
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
█
```

Bước 5: Chọn p để tạo primary partition

Bước 6: Chọn 1 để tạo partition số 1

Bước 7: Chọn default cho cylinder

```
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-652, default 1):
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-652, default 652):
Using default value 652
```

Bước 8: Chọn w để write tables to disk

Bước 9: Kiểm tra partition vừa tạo

```
[root@localhost ~]# fdisk -l

Disk /dev/sdb: 5368 MB, 5368709120 bytes
255 heads, 63 sectors/track, 652 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x9caaca08

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1            1           652     5237158+  83  Linux
```

Bước 10: Định dạng phân vùng

```
#mkfs.ext4 /dev/sdb1
```

Bước 11: Tạo thư mục gắn kết với hdd mới

```
# mkdir /mnt/sdb1
```

Bước 12: Chỉnh sửa file fstab

(Thêm vào dòng sau)

```
/dev/sdb1      /mnt/sdb1      ext4  defaults      0 0
```

Bước 13: Restart Server sau đó kiểm tra bằng lệnh *df*

Giới thiệu file Fstab:

Là một tập tin cấu hình chứa các thông tin về các phân vùng trên ổ cứng cũng như các thiết bị lưu trữ khác trong máy tính của bạn.

Cấu trúc: gồm 6 cột

Cột 1 và cột 2: Thiết bị cần mount và nơi mặc định để mount.

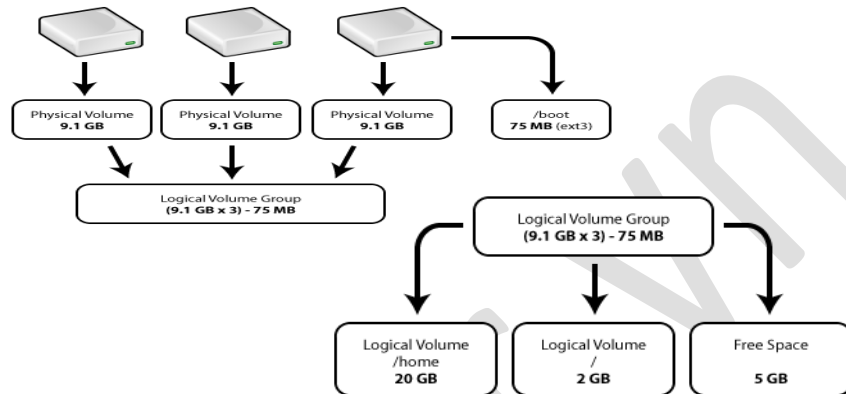
Cột 3: loại file system (ext3,ext4, swap, vfat, xfs ...)

Cột 4: Các tùy chọn mount (auto, noauto, user, ...)

Cột 5: Các tùy chọn cho lệnh dump và fsck

2) Kỹ thuật Logical volume manager (LVM)

LVM là một phương pháp cho phép ấn định không gian đĩa cứng thành những Logical Volume khiến cho việc thay đổi kích thước trở nên dễ dàng.



Các khái niệm

- **Physical Volume:** Là một cách gọi khác của partition trong kỹ thuật LVM, nó là những thành phần cơ bản được sử dụng bởi LVM. Một Physical Volume không thể mở rộng ra ngoài phạm vi một ổ đĩa.
- **Logical Volume Group:** Nhiều Physical Volume trên những ổ đĩa khác nhau được kết hợp lại thành một Logical Volume Group, với LVM Logical Volume Group được xem như một ổ đĩa ảo.
- **Logical Volumes:** Logical Volume Group được chia nhỏ thành nhiều Logical Volume, mỗi Logical Volume có ý nghĩa tương tự như partition. Nó được dùng cho các mount point và được format với những định dạng khác nhau như ext3, ext4 ... Khi dung lượng của Logical Volume được sử dụng hết ta có thể đưa thêm ổ đĩa mới bổ sung cho Logical Volume Group và do đó tăng được dung lượng của Logical Volume.
- **Physical Extent:** là một đại lượng thể hiện một khối dữ liệu dùng làm đơn vị tính dung lượng của Logical Volume.

Một số lệnh của LVM

	create	view	remove	Resize	Change	scan	...
PV	pvcreate	pvs, pvdisplay	pvremove		pvchange	pvscan	...
VG	vgcreate	vgs, vgdisplay	vgremove	vgextend, vgreduce	vgchange	vgscan	...
LV	Lvcreate	lvs, lvdisplay	lvremove	lvextend, lvreduce	lvchange	lvscan	...

Thực hành: Add thêm 2 HDD vào Server Linux

Sau khi thực hiện kết nối vật lý với 2 HDD mới ta thực hiện các bước như sau

Bước 1: kiểm tra 2 hdd vừa add vào

```
#fdisk -l
```

Bước 2: Khởi tạo các phân vùng trên sdb, sdc

```
#fdisk /dev/sdb
```

n, p, 1, enter, enter

Chuyển kiểu partition thành Linux LVM (kiểu 8e): t, 8e

Lưu lại partition table và thoát khỏi fdisk: w

Lặp lại các thao tác trên với sdc

Kiểm tra lại thông tin vừa cấu hình

```
#fdisk -l
```

```
Device Boot      Start          End      Blocks   Id  System
/dev/sdb1        1              652     5237158+ 8e  Linux LVM

Disk /dev/sdc: 5368 MB, 5368709120 bytes
255 heads, 63 sectors/track, 652 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0xf8d6527e

Device Boot      Start          End      Blocks   Id  System
/dev/sdc1        1              652     5237158+ 8e  Linux LVM

Disk /dev/mapper/VolGroup-lv_root: 8564 MB, 8564768768 bytes
255 heads, 63 sectors/track, 1041 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000
```

Bước 3: Tạo Physical Volume

```
[root@localhost ~]# pvcreate /dev/sdb1 /dev/sdc1
Writing physical volume data to disk "/dev/sdb1"
Physical volume "/dev/sdb1" successfully created
Writing physical volume data to disk "/dev/sdc1"
Physical volume "/dev/sdc1" successfully created
[root@localhost ~]#
```

Kiểm tra PV vừa tạo

#pvdisk

```
--- NEW Physical volume ---
PV Name           /dev/sdb1
VG Name
PV Size           4.99 GiB
Allocatable       NO
PE Size           0
Total PE          0
Free PE           0
Allocated PE      0
PV UUID           xymNyH-Gzqp-DbOe-Ehzh-ntYH-EcUN-Ngsrlg

"/dev/sdc1" is a new physical volume of "4.99 GiB"
--- NEW Physical volume ---
PV Name           /dev/sdc1
VG Name
PV Size           4.99 GiB
Allocatable       NO
PE Size           0
Total PE          0
Free PE           0
Allocated PE      0
PV UUID           IDghmW-3TIO-kKV0-NUjj-CfSG-6FuV-crqNaR
```

Bước 4: Tạo Volume Group tên là GAME

```
[root@localhost ~]# vgcreate GAME /dev/sdb1 /dev/sdc1
Volume group "GAME" successfully created
```

Kiểm tra thông tin volume group vừa tạo

```
#vgs
```

```
#vgdisplay
```

Bước 5: Tạo Logical Volume

```
[root@localhost ~]# lvcreate --name game1 --size 2G GAME
Logical volume "game1" created
[root@localhost ~]# lvcreate --name game2 --size 7.51G GAME
Rounding up size to full physical extent 7.51 GiB
Logical volume "game2" created
[root@localhost ~]#
```

Kiểm tra thông tin Logical Volume vừa tạo

```
#lvs
```

```
#lvdisplay
```

Bước 6: Tạo filesystem

```
#mkfs.ext4 /dev/GAME/game1
```

```
#mkfs.ext4 /dev/GAME/game2
```

Bước 7: Mount LV

Tạo thư mục mount

```
#mkdir /var/data1
```

```
#mkdir /var/data2
```

```
#mount /dev/GAME/game1 /var/data1/
```

```
#mount /dev/GAME/game2 /var/data2/
```

Chỉnh sửa file fstab

```
#vi /etc/fstab
```

(sau đó reset Server)

```
#init 6
```

Yêu cầu:

a) Tăng dung lượng cho LV thêm 5 G

```
#lvextend -L +5G /dev/GAME/game2
```

```
#resize2fs /dev/GAME/game2
```

(kiểm tra lại thông tin)

```
#df -h
```

b) Giảm dung lượng cho LV game2 còn 3G

```
#umount /var/data2
```

```
#resize2fs /dev/GAME/game2 3G
```

```
#e2fsck -f /dev/GAME/game2
```

```
#lvchange -an /dev/GAME/game2
```

```
#lvreduce -L 3G /dev/GAME/game2
```

```
#lvchange -ay /dev/GAME/game2
```

```
#mke2fs -t ext4 /dev/GAME/game2
```

```
#mount /dev/GAME/game2 /var/data2
```

Chương 13: LẬP SHELL SCRIPT

Mục tiêu của lập trình shell: Tạo ra 1 chương trình nhằm thực hiện một nhiệm vụ, chức năng nào đó để phục vụ cho việc quản trị cấu hình

1) Các loại shell thông dụng trên Unix/Linux

- a) **sh (Shell Bourne):** Shell nguyên thủy có mặt trên hầu hết các hệ thống Unix/Linux... Nó rất hữu dụng cho việc lập trình Shell nhưng nó không xử lý tương tác người dùng như các Shell khác...
- b) **bash (Bourne Again Shell):** Đây là phần mở rộng của sh, nó kế thừa những gì sh đã có và phát huy những gì sh chưa có. Đây là Shell được cài đặt mặc định trên các hệ thống Linux.
- c) **Csh, tcsh và zsh:** Dòng shell sử dụng cấu trúc lệnh của C. Là shell thông dụng thứ 2 sau bash shell

2) Cách viết shell script

Dùng lệnh **vi/gedit** để viết shell

Nên dùng gedit để viết shell vì nó thể hiện màu sắc để kiểm soát lỗi

Thực thi script:

- ✓ Gán quyền thực thi cho script

```
#chmod a+x ten_script
```

- ✓ Thực thi

```
#!/ten script
```

Cấu trúc một chương trình shell script

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

```
command ....
```

```
exit 0
```


3) Biến trong shell

Trong linux shell thì có 2 kiểu biến:

✓ **Biến hệ thống (system variable):** được tạo bởi Linux. Kiểu biến này thường được viết bằng ký tự in hoa.

✓ **Biến do người dùng định nghĩa**

Định nghĩa biến :

Cú pháp : tên biến=giá trị

Một số quy định về biến trong shell :

- (1) Tên bắt đầu bằng ký tự hoặc dấu gạch chân (_).
- (2) Không được có khoảng trắng trước và sau dấu bằng khi gán giá trị cho biến
- (3) Biến có phân biệt chữ hoa chữ thường
- (4) Bạn có thể khai báo một biến có giá trị NULL như sau: `var01=` hoặc `var01=""`
- (5) Không dùng ?, * để đặt tên biến.

✓ Để truy xuất giá trị biến, dùng cú pháp sau: \$tên_biến

ví dụ :

```
n=10
```

```
echo $n
```

4) Lệnh trong Shell

a) **Lệnh echo:** xuất nội dung ra màn hình

Ví dụ: vi bai1.sh

(có nội dung như sau)

```
#!/bin/sh
```

```
# chương trình xuất ra màn hình một chuỗi
```

```
echo "chào các bạn"
```

Chạy chương trình:

```
#chmod a+x bai1.sh
```

```
#!/bin/sh
```

b) **Lệnh read:** đọc một số hay chuỗi từ bàn phím và gán cho tham_số
read tham_số

c) **Tính toán trong Shell**

✓ **Sử dụng expr**

cú pháp: *expr op1 phép_toán op2*

Ví dụ: *expr 1 + 3*
 expr 2 - 1
 expr 10 / 2
 expr 20 % 3
 *expr 10 * 3*
 echo `expr 6 \+ 3`

✓ **Sử dụng let**

Ví dụ:

let "z=\$z+3"
let "z += 3"
let "z=\$m\$n"*

✓ **Sử dụng \$(...)**

Ví dụ :

z=\$(z+3)
*z=\$((m*n))*

Chú ý khi dùng dấu nháy:

“ :Nháy kép bất cứ gì nằm trong đây nháy kép được xem là những ký tự riêng biệt.

‘ :Nháy đơn những gì nằm trong đây nháy đơn có ý nghĩa không đổi.

` :Nháy ngược thực thi lệnh

d) **Tham số lệnh**

Tham số lệnh: Là tham số được truyền khi thực hiện script.

Ví dụ: *./bai2.sh phepcong 12 24*

\$1, \$2, \$3 ... : vị trí và nội dung của các tham số trên dòng lệnh theo thứ tự từ trái sang phải.

\$@, \$*: danh sách tất cả các tham số trên dòng lệnh.

Điều kiện:

Nhận 1 trong 2 giá trị đúng hoặc sai

Sử dụng trong các câu lệnh điều khiển (if, while, ..)

Lệnh *test* hoặc []

Cấu trúc: *test dieu_kien* hoặc [*dieu_kien*]

Kiểm tra điều kiện trên tập tin

-d file	<i>true</i> nếu file là thư mục
-e file	<i>true</i> nếu file tồn tại trên đĩa
-f file	<i>true</i> nếu file là tập tin thông thường
-g file	<i>true</i> nếu set-group-id được thiết lập trên file
-r file	<i>true</i> nếu file cho phép đọc
-s file	<i>true</i> nếu kích thước file khác 0
-u file	<i>true</i> nếu set-ser-id được áp đặt trên file
-w file	<i>true</i> nếu file cho phép ghi
-x file	<i>true</i> nếu file được phép thực thi

So sánh chuỗi

So sánh	Kết quả
string1 = string2	<i>true</i> nếu 2 chuỗi bằng nhau (chính xác từng ký tự)
string1 != string2	<i>true</i> nếu 2 chuỗi không bằng nhau
-n string1	<i>true</i> nếu string1 không rỗng
-z string1	<i>true</i> nếu string1 rỗng (chuỗi null)

Các toán tử so sánh số học

-eq : bằng

-ge : lớn hơn hoặc bằng

-gt : lớn hơn

-le : nhỏ hơn hoặc bằng

-lt : nhỏ hơn

-ne : khác

Lệnh if:

Cấu trúc:

if condition

then

statements

else

statements

fi

Lệnh for

for *variable in values*

do

statements

done

Lệnh while

while *condition do*

statements

done

Lệnh until

until *condition*

do

statements

done