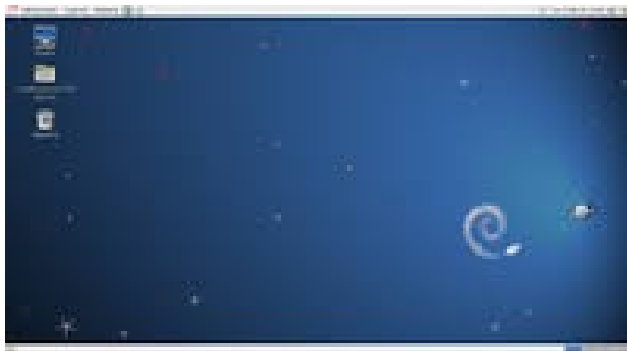


## I, LINUX

Đây là một hệ điều hành nhân Linux kernel mã nguồn mở, được ra mắt lần đầu vào năm 1991, do Linus Torvalds tạo ra. Linux kiểu như 1 hệ điều hành Unix đã được mô-đun hóa. Sau một khoảng thời gian ngắn phát triển, từ phiên bản Linux đầu tiên, hiện tại đã có nhiều phiên bản Linux (Distro) mà phổ biến nhất có thể nhắc tới : ubuntu, fedora, debian, linux mint, ... dưới đây là 1 vài hình ảnh về giao diện của những hệ điều hành này:



### 1. Tổng quan , cài đặt Linux

Có nhiều lý do để nhiều người chọn Linux trở thành hệ điều hành chính trên máy tính của mình, và đặc biệt là những lập trình viên, có thể kể đến những lý do cơ bản sau:

- + , về vấn đề bản quyền, chi phí: Linux thuộc phần mềm mã nguồn mở, nên người sử dụng không cần phải lo lắng nhiều về vấn đề bản quyền, đồng thời chi phí cho những phần mềm kiểu này là ít hoặc free.

- + , Linux là 1 hệ điều hành linh hoạt, uyển chuyển. bởi nó là 1 hệ điều hành mã nguồn mở nên bản thân người dùng có thể tùy chỉnh, thay đổi, sửa chữa theo ý mình nếu có đủ khả năng. tính linh hoạt , uyển chuyển của hệ điều hành này còn thể hiện ở chỗ nó tương thích với nhiều môi trường : môi trường máy chủ, desktop, nhúng vào các thiết bị điều khiển,...

- + , Linux mang tính bảo mật, an toàn cao: Linux có một cơ cấu phân quyền người sử dụng hết sức chặt chẽ: chỉ những Root mới có quyền can thiệp vào hệ thống để cài đặt và thay

đổi. Bên cạnh đó Linux cũng có chế độ cho phép một người dùng bình thường có khả năng sử dụng quyền của Root trong một khoảng thời gian cho một số hành động. Cơ chế phân quyền như này giúp tránh khỏi những sai sót, đổ vỡ trong hệ thống.

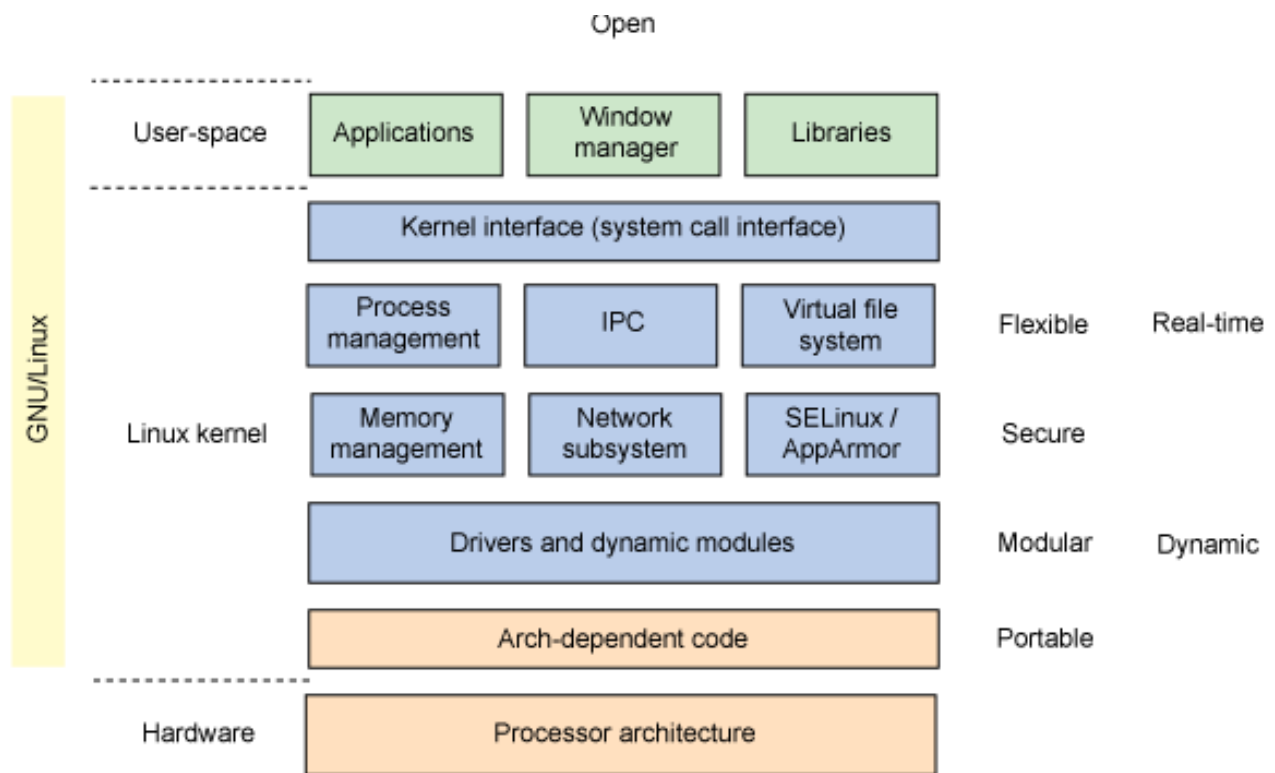
Tính chất mở một lần nữa thể hiện điểm mạnh của mình, khi nó giúp Linux tăng sự an toàn hệ thống. khi một lỗi hỏng bị phát hiện, thì nó sẽ được cả cộng đồng mã nguồn mở tham gia sửa chữa, trong một thời gian ngắn lỗi hỏng đó sẽ được fix. bên cạnh đó thì, vì người quản trị cũng “nhìn” thấy được từng ngõ ngách của hệ thống nên cũng phần nào đảm bảo sự an toàn.

+, thích hợp cho việc quản trị mạng: bởi vì ngay từ đầu được tạo ra cho chế độ đa người dùng, nên Linux được coi là hệ điều hành mạng rất có giá trị : tính bảo mật cao, chạy ổn định, cơ chế chia sẻ tài nguyên linh hoạt (TCP/IP)

+, chạy thống nhất trên các hệ thống phần cứng: mặc dù có rất nhiều phiên bản Linux khác nhau do các nhà phân phối khác nhau ban hành nhưng nhìn chung chúng đều chạy khá ổn định trên các thiết bị phần cứng.

Đây là 1 phần mềm mã nguồn mở nên việc cài đặt không có gì rắc rối lắm.

## 2. tiêu chuẩn phân cấp hệ thống file

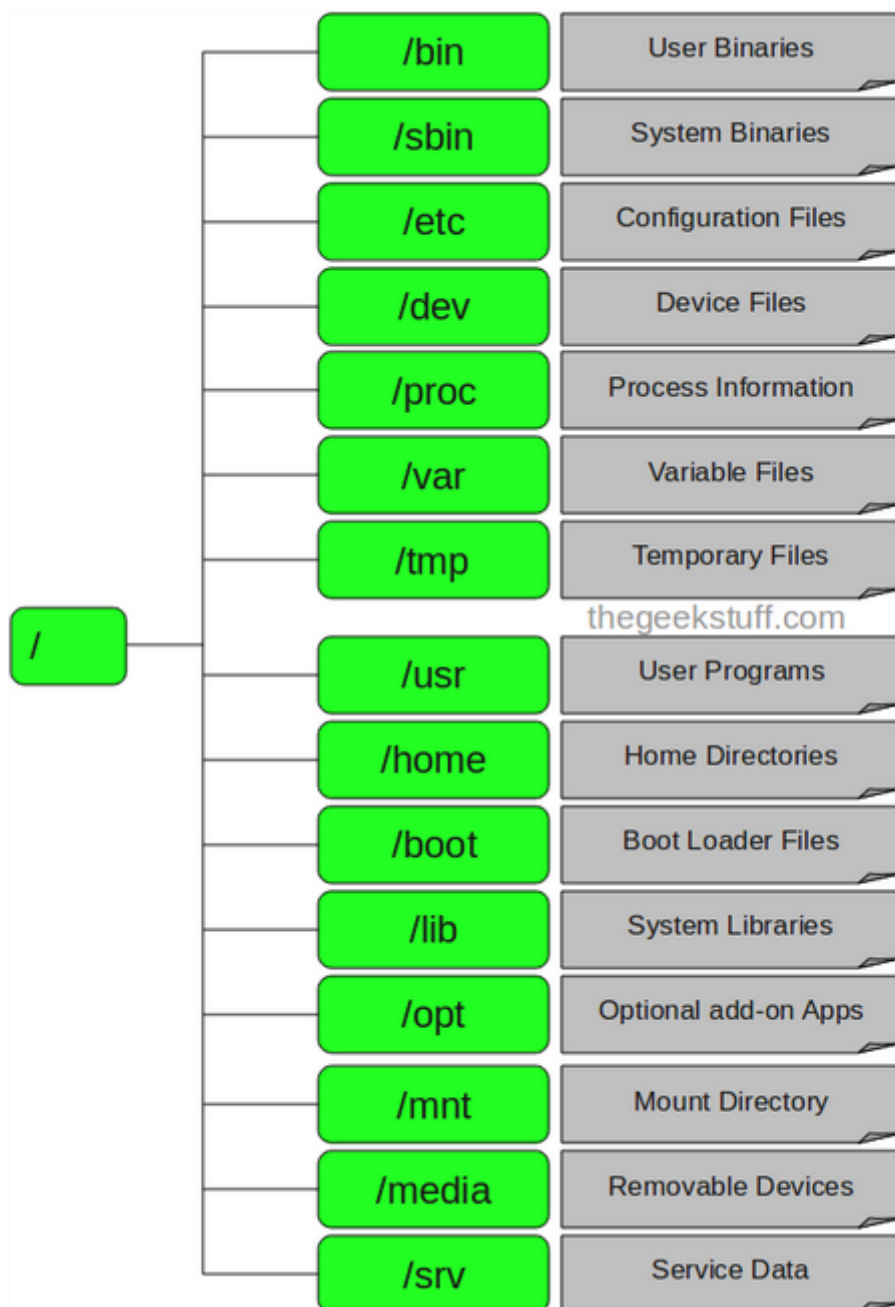


ảnh trên đây là hình ảnh Linux nhìn từ ngoài, có thể thấy rõ từ ảnh, Linux gồm một nhân kernel(mã cốt quản lí tài nguyên phần cứng, phần mềm) cùng một bộ sưu tập các ứng dụng của người dùng ở phía trên .

Tầng cuối cùng Portable là tập hợp mã kiến trúc giúp Linux có thể hỗ trợ đa nền tảng phần cứng.

Cấu trúc tập tin trong Linux được tổ chức theo một hệ thống phân bậc tương tự cấu trúc của một cây phân cấp, với bậc cao nhất là thư mục gốc , kí hiệu “/” ( root directory). Đối với hệ điều hành UNIX và LINUX tất cả các thiết bị kết nối với máy tính đều được nhận dạng như các tập tin, kể cả những linh kiện như : ổ đĩa cứng, các phân vùng đĩa cứng, các ổ usb. Điều này cũng có nghĩa tất cả các tập tin, thư mục đều nằm dưới thư mục gốc ( kể cả những tập tin biểu tượng cho các ổ đĩa cứng) .

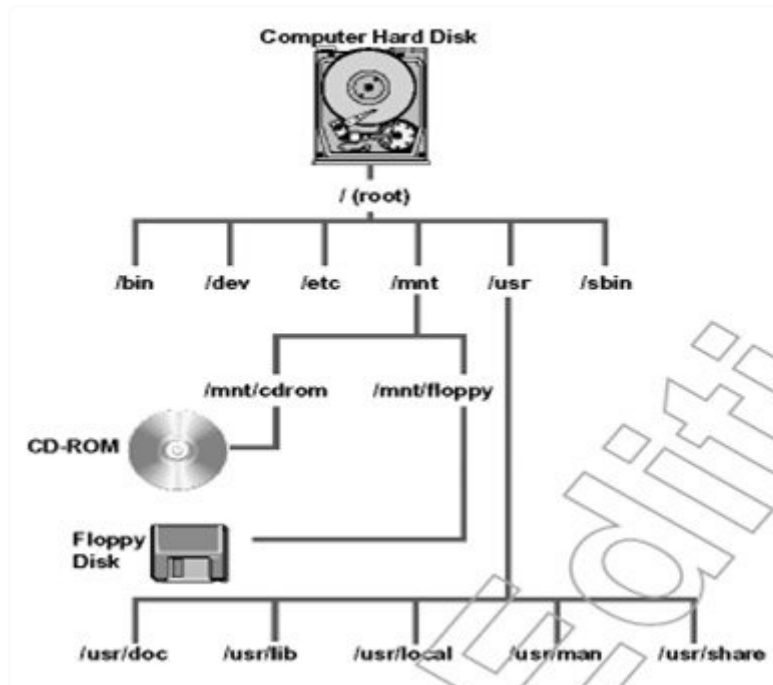
nằm dưới thư mục gốc (/) có một loạt các thư mục quan trọng của hệ thống tập tin . dưới đây là hình ảnh các thư mục đó:



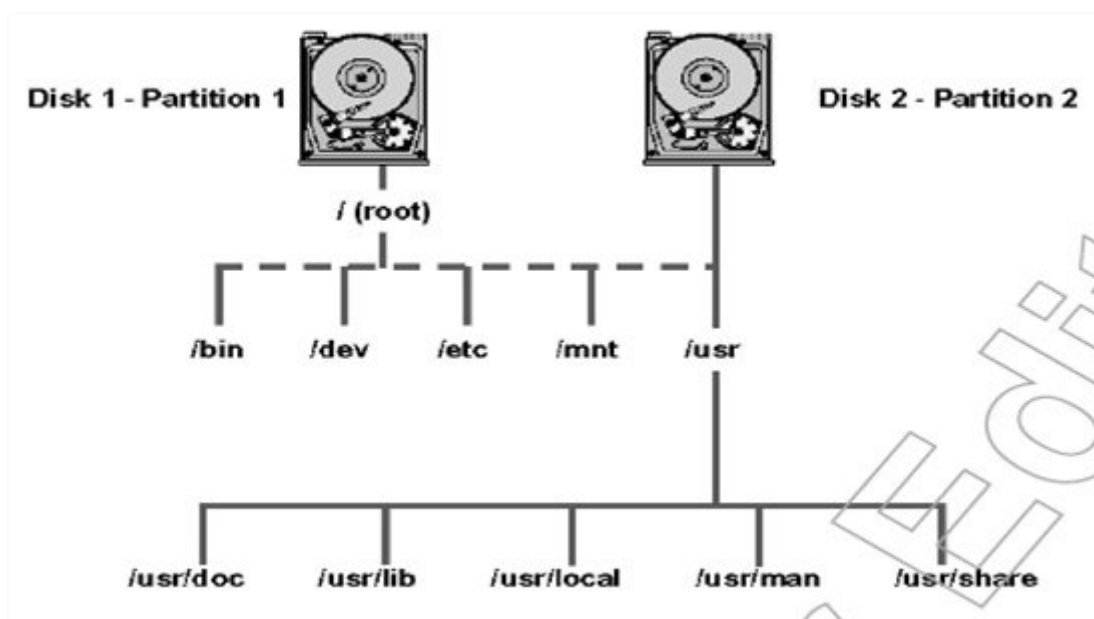
trong đó:

- / (root): thư mục gốc, chỉ có root user mới có quyền viết dưới thư mục này.
- /bin (user binaries): chứa các tập tin thực thi nhị phân, chứa các lệnh phổ biến sử dụng ở chế độ single-user mode, ví dụ các lệnh: ls, ping, ps, cp,...
- /sbin (system binaries): cũng là nơi chứa các tập tin thực thi nhị phân. các lệnh trong thư mục này được sử dụng bởi admin hệ thống nhằm mục đích duy trì hệ thống, ví dụ: iptables, reboot, ifconfig,...
- /etc: chứa các tập tin cấu hình hệ thống, tập tin lệnh để khởi động dịch vụ của hệ thống. bên cạnh đó còn có các shell scripts startup, shutdown để chạy, ngừng các chương trình các nhân, ví dụ: /etc/resolv.conf,...
- /proc: chứa thông tin về system process, hệ thống tập tin giả chứa thông tin về các quá trình đang chạy.
- /usr: chứa các ứng dụng, thư viện, tài liệu, mã nguồn các chương trình thứ cấp/
- /home: thư mục chính lưu trữ các tập tin cá nhân của tất cả user
- /boot: chứa các tập tin cấu hình cho quá trình khởi động hệ thống

- /dev: chứa tập tin để nhận biết các thiết bị của hệ thống bao gồm thiết bị đầu cuối, USB, thiết bị được gắn với hệ thống.  
dưới đây là hình ảnh về cây thư mục phân cấp này:



dưới đây sẽ là hình ảnh Ổ đĩa và các phân vùng:



### 3. bash shell

Shell là một chương trình đặc biệt, chấp nhận câu lệnh tiếng anh theo một cú pháp cho trước và chuyển nó đến kernel để xử lý tiếp, sau đó trả lại kết quả cho người dùng. Có thể nói nó như một chương trình tương tác giữa người dùng và máy tính, thông dịch các lệnh của người dùng nhập vào hoặc từ các file. Nhưng shell thực tế không phải là một phần của hạt nhân hệ thống, nó chỉ dùng hạt nhân để thực thi các chương trình, tạo ra các tập tin. Các loại shell script:

- + , BASH(shell thông dụng nhất trên linux)
- + , CSH
- + , KSH
- + , TCSH

mỗi loại shell sẽ có cú pháp, cách sử dụng, công dụng khác nhau. chú ý trong báo cáo này, em sẽ tìm hiểu về BASH

sử dụng câu lệnh : “echo \$SHELL” để biết được hệ thống đang chạy shell gì.

Nội dung chương trình bao gồm các câu lệnh được sử dụng trên dòng lệnh của Linux, các câu lệnh trên cùng 1 dòng phân cách nhau bằng dấu “;”

Một số câu lệnh thiết lập quyền, thực thi 1 chương trình Shell:

- + , chmod o+x tên file (câu lệnh thiết lập quyền)
- + , bash tên file
- sh tên file
- ./tên file

(các câu lệnh cho chức năng thực thi)

a, Biến trong shell

\*, Linux shell có 2 loại biến :

- Biến hệ thống: đây là biến được tạo ra và quản lý bởi Linux. tên biến viết hoa
- Biến do người dùng định nghĩa: đây là biến được tạo ra và quản lý bởi người dùng. tên biến viết thường

\*, Cách xem , truy nhập giá trị các biến:

Có \$ trước tên biến

VD: echo \$bien

dưới đây là bảng 1 số biến hệ thống thông dụng

System Variable	Meaning
BASH=/bin/bash	Our shell name
BASH_VERSION=1.14.7(1)	Our shell version name
COLUMNS=80	No. of columns for our screen
HOME=/home/vivek	Our home directory
LINES=25	No. of columns for our screen
LOGNAME=students	students Our logging name
OSTYPE=Linux	Our Os type
PATH=/usr/bin:/sbin:/bin:/usr/sbin	Our path settings
PS1=[\u@\h \W]\\$	Our prompt settings
PWD=/home/students/Common	Our current working directory
SHELL=/bin/bash	Our shell name
USERNAME=vivek	User name who is currently login to this PC

\*, Định nghĩa các biến của người dùng:

tên\_biến = giá trị

VD: n = 3

echo \$n

#in giá trị của biến n

bên cạnh đó cũng sẽ có các quy tắc khi đặt tên biến như: tên biến phải bắt đầu bằng kí tự, không được để dấu cách 2 bên toán tử bằng, phân biệt chữ hoa chữ thường, khi biến không được khởi tạo sẽ có giá trị mặc định là NULL, không có các kí tự “?”””\*” trong tên biến.

b. Các phép toán.

Để thực hiện các phép tính số học cần dùng câu lệnh: expr biểu\_thức

Trong đó có các toán tử: +,-,\\*,/,%

VD: expr 5\\*7

echo ‘expr 6+9’ # in ra giá trị biểu thức

Chú ý: đối với dấu “”, tất cả các kí tự trong cặp dấu này đều không có ý nghĩa tính toán, trừ những kí tự sau \ hoặc \$

đối với cặp dấu `` sẽ yêu cầu thực hiện lệnh

c. Kết thúc câu lệnh

Linux trả về mặc định 0 nếu câu lệnh kết thúc thành công

\$? : để kiểm tra trạng thái kết thúc 1 câu lệnh

d. Các cấu trúc rẽ nhánh

Cú pháp câu lệnh (if....then)

if điều kiện

then

câu lệnh

fi

(thường dùng cho trạng thái kết thúc câu lệnh là 0)

đối với câu lệnh if...then...else cũng tương tự

e. Các phép toán kiểm tra

\*, các phép toán kiểm tra

#### For Mathematics, use following operator in Shell Script

Mathematical Operator in Shell Script	Meaning	Normal Arithmetical/ Mathematical Statements	But in Shell	
			For test statement with if command	For [ expr ] statement with if command
-eq	is equal to	5 == 6	if test 5 -eq 6	if [ 5 -eq 6 ]
-ne	is not equal to	5 != 6	if test 5 -ne 6	if [ 5 -ne 6 ]
-lt	is less than	5 < 6	if test 5 -lt 6	if [ 5 -lt 6 ]
-le	is less than or equal to	5 <= 6	if test 5 -le 6	if [ 5 -le 6 ]
-gt	is greater than	5 > 6	if test 5 -gt 6	if [ 5 -gt 6 ]
-ge	is greater than or equal to	5 >= 6	if test 5 -ge 6	if [ 5 -ge 6 ]

**NOTE:** == is equal, != is not equal.

\*, Các phép so sánh xâu

### For string Comparisons use

Operator	Meaning
string1 = string2	string1 is equal to string2
string1 != string2	string1 is NOT equal to string2
string1	string1 is NOT NULL or not defined
-n string1	string1 is NOT NULL and does exist
-z string1	string1 is NULL and does exist

\*, Các phép kiểm tra file, thư mục

### Shell also test for file and directory types

Test	Meaning
-s file	Non empty file
-f file	Is File exist or normal file and not a directory
-d dir	Is Directory exist and not a file
-w file	Is writeable file
-r file	Is read-only file
-x file	Is file is executable

\*, Các phép toán logic

- not: !biểu\_thức

-and: -a

- or: -r

f. Các cấu trúc lặp

\*, Cấu trúc lặp for

*Cú pháp:*

for {tên biến} in {danh sách}

do

    các câu lệnh

done

<hoặc>

for ((expr1;expr2;expr3))

do

    các câu lệnh

done

VD:

for i in 1 2 3 4

do

    echo "Hello"

done

\*, Cấu trúc lặp while

*Cú pháp:*

```
while [điều kiện]
do
    các câu lệnh
done
```

g. Cấu trúc case

```
case $ten_biến in
pattern1) câu lệnh;;
pattern2) câu lệnh;;
*)câu lệnh ;;
esac
```

#### 4. init system:

Init process là tiến trình được khởi động lên đầu tiên trong hệ thống Linux. Tức là sau khi chọn hệ điều hành trong menu của Boot Loader, hệ điều hành bắt đầu được khởi động và tiến trình đầu tiên được khởi động đó chính là init. Nhiệm vụ của init là start và stop các process services. vì nó là tiến trình được khởi động đầu tiên trên hệ thống nên init process luôn có PID(Process ID) là 1.

Có 3 kiểu triển khai init chính trong hệ thống Linux là:

- + , system V (cũ)
- + , Upstart (cũ)
- + , Systemd (init system được phát triển khoảng năm 2010 dùng để thay thế các init system cũ)

Init Systemd không chỉ dừng lại ở việc start / stop các services nó còn có thể mount filesystems, quản lí network sockets,...và quản lí các công việc như thế, Systemd phân chia ra các đơn vị units:

- + , Service units(.service) để start và stop các unit
- + , Mount units(.mount) để quản lí các mount point
- + , Target unit(.target) để điều khiển các runlevels(sử dụng trong sysV init)

trong systemd khái niệm runlevels được thay thế bằng các targets để boot vào như:

- poweroff.target-shutdown system
- rescue.target-singer user mode
- multi-user.target-multiuser with networking
- graphical.target-multiuser with networking and GUI
- reboot.target-restart

trong khi ở SysV init system có các runlevels là:

- 0: halt
- 1: singer-user
- 2: multi-user
- 3: multi-user with networking
- 4: undefined ( user defined)
- 5: multi-user with display manager (graphical login)
- 6: reboot



## II, GIT

### 1. Cơ bản về Git

Git là một phần mềm quản lý mã nguồn phân tán được phát triển bởi Linus Torvalds năm 2005, ban đầu dành cho việc phát triển nhân Linux. Hiện nay, Git trở thành một trong các phần mềm quản lý mã nguồn phổ biến nhất. Nó có khả năng chạy trên nhiều hệ điều hành khác nhau: Linux, Windows, Mac OSX,...( trích wikipedia)

Các khái niệm cơ bản trong git:

+, **repository**: (repo) đây là nơi chứa tất cả thông tin cần thiết để duy trì và quản lý các sửa đổi và lịch sử dự án.

repo có 2 cấu trúc dữ liệu chính là object store và index. tất cả dữ liệu của repo đều được chứa trong thư mục đang làm việc dưới dạng folder ẩn có tên là .git

+, **Object store**: là trái tim của Git, nó chứa những dữ liệu gốc, các file log ghi chép quá trình sửa đổi, tên người tạo file, ngày tháng, ....Git có 4 loại objects: blobs, trees, commits, tags

. blobs: là file nhị phân có thể chứa mọi loại dữ liệu bất kể là dữ liệu của chương trình gì

. trees: lớp đại diện cho thông tin thư mục như thông tin định danh của blob, đường dẫn, metadata chứa thông tin cấu trúc và các thư mục nhỏ chứa trong thư mục đó.

. commits: chứa metadata có thông tin về mọi thứ như tên tác giả, người tải lên, ngày tải, thông tin log,...

. tags: đánh dấu

+, **index**: là file nhị phân động tạm thời miêu tả cấu trúc thư mục của toàn bộ repo và trạng thái của dự án được thể hiện thông qua commit và tree tại 1 thời điểm nào đó trong lịch sử phát triển. index không chứa nội dung file mà chỉ dùng để truy tìm những thứ mà bạn muốn commit.

Một số website và phần mềm cho phép lưu trữ Git:

+, GitHub

+, GitLab

+, GNU Savannah...

Cơ chế lưu trữ phiên bản của Git là nó sẽ tạo ra một “ảnh chụp” (snapshot) trên mỗi tập tin và thư mục sau khi commit, từ đó nó cho phép bạn tái sử dụng lại 1 ảnh chụp nào đó mà có thể hiểu đó như là 1 phiên bản. Đây cũng chính là lợi thế của Git so với các Distributed Version Control System (DVCS) khác khi nó không lưu cứng dữ liệu mà sẽ lưu dưới dạng snapshot.

### 2. Cài đặt Git

\*, Cài đặt Git trên Ubuntu

*Sử dụng lệnh sau để cài Git:*

```
sudo apt-get install git
```

*Thiết lập chứng thức cá nhân:*

```
$git config --global user.name "tên"
```

```
$git config --global user.email "tên email"
```

*Tạo repository:*

có 2 loại repository: local và remote , với kho chứa trên máy cá nhân và kho chứa trên máy chủ từ xa.

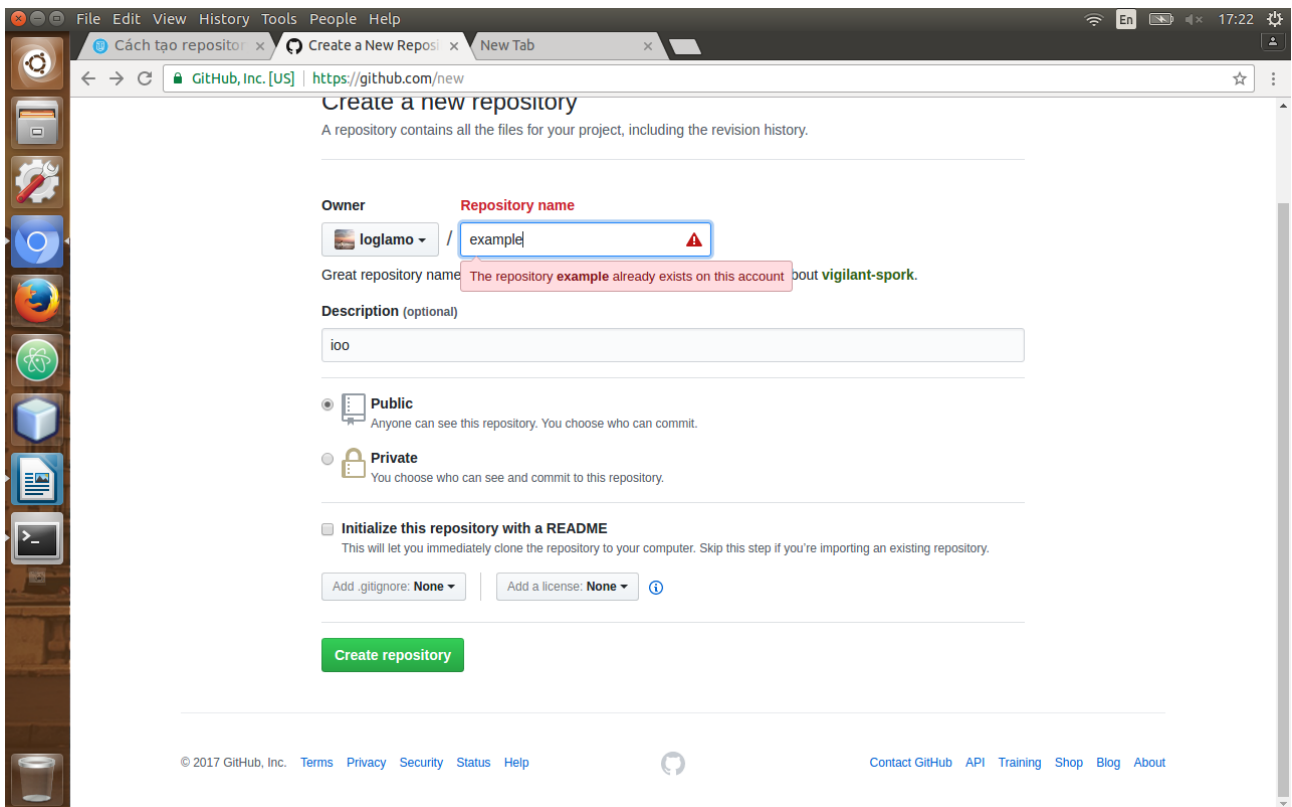
- tạo local: git init tên\_file

sau khi tạo ra file này sẽ hiện lên 1 đường dẫn, với cuối đường dẫn là .git/, đây là 1 thư mục ẩn, chứa các thiết lập về git, lưu lại tất cả thông tin về kho chứa, và không cần động chạm đến. Nếu kho đã có sẵn mã nguồn thì phải đưa tệp về tracked

git add tên\_file: tracked

git status : để xem danh sách tệp tin đã được tracked

- tạo remote: trên github



sau khi tạo xong 1 repo trên github sẽ có 1 đường dẫn đến repo này, và khi muốn clone kho về máy ta sử dụng câu lệnh : \$ git clone <địa chỉ>

sau khi đã clone về máy, ta có thể thay đổi tùy ý và sau đó có thể đẩy lên github bằng câu lệnh: \$git push origin master

### 3. SVN

Bên cạnh Git\_ 1 version control còn có SVN(subversion). Đây cũng là 1 trong những version control được sử dụng phổ biến hiện nay. Hiện tại kho plugin WordPress cũng đang sử dụng SVN.

Nó làm việc với 1 máy chủ trung tâm kho mã nguồn, những kho đó được chia vào 3 key chính là : Trunk, Branches, Tags. trong đó:

- trunk :

là nơi lưu trữ mã nguồn gốc, như khu vực trung tâm nơi bất cứ ai làm việc trên dự án sẽ lấy các cập nhật tại đây. khi bạn làm việc với 1 tính năng mới bạn sẽ branches mã nguồn từ trunk, sau khi tính năng đó được hoàn thành, nó sẽ được đưa vào trunk.

- Branches:

Sử dụng khi tạo ra tính năng mới, vì vậy sẽ branch mã nguồn từ trunk. Điều này có nghĩa là sẽ lấy ra được bản sao chép trunk và đặt trong 1 folder mới bên trong khu vực branch, và tạo những thay đổi trên folder đó. sau khi hoàn thành sẽ merger những thay đổi đó vào trunk.

- Tags:

Nó cũng là 1 phương án tạo mã nguồn, nhưng đối với một bản sao chép từ mã nguồn trunk và đặt nó vào bên trong 1 folder mới với thư mục mới. Sự khác nhau ở đây là tag sẽ không bao giờ sử dụng cho phát triển mã nguồn, chúng chỉ là phương án dễ dàng cho việc lấy lại mã nguồn. Thời điểm sử dụng tag là cho việc triển khai mã nguồn, khi hoàn thành những mã nguồn mới, merge vào trunk, test đầy đủ và sẵn sàng cho việc phát hành, khi đó sử dụng tag. Tag sẽ được tạo trên trunk và đánh dấu tính năng mới đó. Lợi ích làm việc này là nếu bản phát hành mới đổ vỡ, có thể lấy lại mã nguồn quay trở lại thời điểm tag trước đó.

Mỗi loại version control này có những đặc điểm, lợi ích riêng nên tùy từng yêu cầu , sở thích từng người để chọn cho mình những loại quản lí phiên bản phù hợp.