PHÁT TRIỂN PHẦN MỀM NGUỒN MỞ

ThS. Trần Văn Hùng

Mail: hung.tranvan@stu.edu.vn

2024

NỘI DUNG CHÍNH

Chương 1: Tổng quan về phát triển phần mềm mã nguồn mở

Chương 2: Mô hình phát triển phần mềm mã nguồn mở

Chương 3. Hạt nhân Linux (Linux Kernel)

Chương 4. Công cụ và kỹ thuật hỗ trợ phát triển

Chương 5. Qui trình phát triển phần mềm mã nguồn mở

Chương 1:

TỔNG QUAN VỀ PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN MỞ

Nội dung

- Phần mềm và mã nguồn phần mềm
- Chủ sở hữu phần mềm
- Giấy phép sử dụng phần mềm (License)
- Phần mềm tự do
- Hệ điều hành Linux
- Phần mềm mã nguồn mở (Open Source Software)

Phần mềm và mã nguồn

Một chương trình viết bằng C++, có tên file nguồn là hello.cpp và có nội dung

```
#include<iostream>
using namespace std;
int main()
{
  cout << "Hello World!";
  return 0;
}
```

File nguồn được viết bởi tác giả A

File nguồn này được biên dịch sang file thực thi. Hello.exe

Phần mềm và mã nguồn

- Tập tin hello.exe được gọi là một phần mềm, hay một chương trình phần mềm. Nội dung của hello.exe bao gồm các mã máy, mã thực thi hay mã nhị phân, là các chỉ thị mà máy tính phải thực hiện,.
- Tập tin hello.cpp được gọi là mã nguồn của phần mềm hello.exe
- A: chủ sở hữu của phần mềm (cả 2 tập tin hello.exe và hello.cpp)

Chủ sở hữu phần mềm

- Khi một phần mềm được tạo ra nó thuộc một chủ sở hữu nào đó.
- Chủ sở hữu có thể là một cá nhân (lập trình viên viết ra phần mềm) hoặc là một công ty phần mềm (người bỏ tiền ra thuê mướn lập trình viên trực tiếp viết phần mềm cho công ty).

Chủ sở hữu phần mềm

- Chủ sở hữu phần mềm có toàn quyền trên phần mềm mà họ là chủ sở hữu, và sẽ quyết định mức độ sử dụng và khai thác của những người khác trên phần mềm mà họ là chủ sở hữu.
- Khi muốn sử dụng một phần mềm đó, người sử dụng phải xin phép chủ sở hữu phần mềm thông qua một giấy phép được cập bởi chủ sở hữu phần mềm.

Giấy phép sử dụng phần mềm

- Giấy phép sử dụng phần mềm (License) được chủ sở hữu phần mềm cấp cho người muốn sử dụng phần mềm.
- Nó là một bản hợp đồng gồm các điều khoản và điều kiện, mô tả những gì mà chủ sở hữu phần mềm cho phép khai thác phiên bản phần mềm liên quan. Nó qui định về những khả năng mà ta có thể có được trên phần mềm được cấp giấy phép sử dụng.
-

Giấy phép sử dụng phần mềm

- Giấy phép của một số loại phần mềm phổ biến
  - Phần mềm thương mại
  - Phần mềm miễn phí (freeware) và phần mềm trả một phần (shareware)
  - Phần mềm mã nguồn mở

Phần mềm thương mại

- Chỉ cho phép người sử dụng khai thác phần mềm theo những ràng buộc đã ghi rõ trong giấy phép.
- Bản quyền loại này rất bị hạn chế. Trong trường hợp có những lỗi phần mềm được phát hiện hay một số chức năng hoạt động không tốt thì người sử dụng phải chờ cho đến khi chủ sở hữu phần mềm sửa đổi chúng. ..

Phần mềm miễn phí...

- Là các phần mềm có chủ sở hữu. Được phân phối một cách tự do.
- Phần mềm miễn phí không đòi hỏi tiền bản quyền sử dụng phần mềm.
- Phần mềm trả một phần thì sau một khoản thời gian đã định người sử dụng phải trả tiền nếu như muốn được phép sử dụng tiếp.
- Cả hai loại phần mềm này đều không cho phép người sử dụng truy cập vào mã nguồn của phần mềm.

Phần mềm mã nguồn mở

- Giấy phép phần mềm này qui định rằng nó được phân phối đến người sử dụng cùng với mã nguồn của nó mà chúng có thể bị sửa đổi.
- Nó có thể được phân phối lại mà không bị một ràng buộc nào khác.
- Chúng ta có thể phân phối cả những thay đổi mà chúng ta đã thực hiện trên mã nguồn gốc

Phần mềm mã nguồn mở

- Các điều khoản, điều kiện trong các giấy phép sử dùng phần mềm khác nhau là khác nhau. Có 3 khả năng đối với người sử dụng :
  - Khả năng phân phối lại (Distribution Possibility): Quyền được phép sao chép và phân phối lại phiên bản phần mềm hay không ?
  - Khả năng truy cập vào mã nguồn (Accessibility to source code): cho phép xem mã nguồn, sử dụng, sửa đổi mã nguồn phần mềm của họ cho mục đích của bạn hay không ?
  - Phí sử dụng phần mềm (Free).

So sánh giấy phép phần mềm

	Khả năng phân phối lại	Truy cập vào mã nguồn	Miễn phí
Phần mềm thương mại (Commercial Software)	Không	Không	Không
Phần mềm miễn phí (Freeware)	Đa số	Không	Có
Phần mềm trả một phần (Shareware)	Đôi khi	Không	Không
Phần mềm mã nguồn mở (Open Source Software)	Được phép	Được phép	Đa số

Phần mềm tự do

- Phong trào phần mềm tự do
- Phần mềm tự do (Free Software)
- Giấy phép sử dụng phần mềm GPL (General Public License)
- GNU GPL V2
- LGPL

Phong trào phần mềm tự do

- Nhằm tạo ra những Phần mềm tự do (free software) là những phần mềm mà người dùng có thể tự do chia sẻ, nghiên cứu và sửa đổi chúng.
- Được khởi xướng bởi Richard M. Stallman vào năm 1983 khi ông bắt đầu dự án GNU ("GNU is Not UNIX") nhằm thay thế hệ điều hành Unix với tính năng tự do
- Thành lập quỹ phần mềm tự do (FSF - Free Software Foundation) năm 1985

Phần mềm tự do

- Phần mềm tự do (Free Software) PMTD) đề cập đến sự do, không đề cập đến vấn đề chi phí/giá cả.
- Một phần mềm tự do cho phép người sử dụng phần mềm có 4 khả năng tự do sau:
  - Tự do thực thi chương trình cho bất kỳ mục đích gì
  - Tự do nghiên cứu cách thực thi của chương trình và sửa đổi chúng cho mục đích của bạn. Truy cập vào mã nguồn chương trình là tiền đề
  - Tự do phân phối phần mềm cho người khác
  - Tự do cải tiến chương trình và phân phối cải tiến của bạn cho cộng đồng.

Giấy phép sử dụng phần mềm GPL

- GPL (General Public License)
- Thông thường, các phần mềm đều được copyright nhằm bảo về quyền tác giả.
- Copyleft: là một phương pháp tổng quát nhằm làm cho một chương trình tự do và yêu cầu tất cả những phiên bản sửa đổi hay mở rộng của chương trình cũng phải tự do.
- Khái niệm Copyleft được cụ thể hóa trong giấy phép GNU General Public License (GNU GPL).
- Đây là giấy phép cho phần mềm tự do, được phát hành cho phần lớn các sản phẩm của dự án GNU.

Các phiên bản của GPL

- Version 1 – General Public License – GPL v1, 1989
- Version 2 – Library General Public License – LGPL v2, 1991
- Version 3 – GPLv3, 2007

Phần mềm mã nguồn mở

- Từ free trong tên gọi của phần mềm tự do (Free Software) làm liên tưởng đến một loại phần mềm miễn phí hơn là 4 yếu tố tự do của nó và phần mềm tự do thì không thể làm thương mại được.
- Để tránh sự hiểu lầm này Eric Raymond and Bruce Perens đề xuất một tên gọi khác là Phần mềm mã nguồn mở (Open Source Software).
- Có 10 tiêu chí để đánh giá xem một giấy phép sử dụng phần mềm có đạt chuẩn là một phần mềm mã nguồn mở hay không.

Phần mềm mã nguồn mở

- Tiêu chí (1): Tự do phân phối lại (Free Redistribution): Bản quyền sẽ không hạn chế bất cứ ai bán hoặc cho phần mềm; và không đòi hỏi tiền bản quyền hay một chi phí nào cho thương vụ này.
- Tiêu chí (2): Mã nguồn (Source Code) phải được phân phối cùng với mã nguồn được công bố bằng những phương tiện công cộng mà người ta có thể lấy được mã nguồn với một chi phí sao chép hợp lý nhất

Phần mềm mã nguồn mở

- Tiêu chí (3): Sản phẩm kế thừa (Derived Works): Giấy phép phải công nhận những sửa đổi và những sản phẩm kế thừa; và phải cho phép chúng được phân phối với cùng những điều khoản như giấy phép của phần mềm ban đầu
- Tiêu chí (4) Tính toàn vẹn của mã nguồn của tác giả: Giấy phép có thể ngăn cản việc phân phối mã nguồn dưới dạng bị sửa đổi chỉ khi giấy phép chấp nhận sự phân phối các tập tin vá lỗi (patch file) với mã nguồn vì mục đích sửa đổi chương trình tại thời điểm xây dựng (built time) chương trình. Giấy phép phải cho phép một cách tường minh việc phân phối phần mềm tạo ra từ mã nguồn bị sửa đổi. Giấy phép có thể yêu cầu những sản phẩm kế thừa phải mang một cái tên khác hoặc số phiên bản khác so với phần mềm gốc.

Phần mềm mã nguồn mở

- Tiêu chí (5): Không phân biệt đối xử giữa các cá nhân và các nhóm (No Discrimination Against Persons or Groups)
- Tiêu chí (6): Không phân biệt đối xử với mục đích sử dụng (No Discrimination Against Fields of Endeavor)
- Tiêu chí (7): Phân phối giấy phép (Distribution of License): Những quyền được kèm với chương trình phải được áp dụng đối với tất cả những người mà sau đó chương trình được phân phối lại mà không cần thiết phải thực thi thêm những giấy phép phụ của những thành phần này
-

Phần mềm mã nguồn mở

- Tiêu chí (8): Giấy phép không được dành riêng cho một sản phẩm (License Must Not Be Specific to a Product)
- Tiêu chí (9): Giấy phép không được cản trở phần mềm khác (License Must Not Restrict Other Software): không được đặt những hạn chế lên những phần mềm khác cùng được phân phối với phần mềm của giấy phép này.
- Tiêu chí (10): Giấy phép phải trung lập về mặt công nghệ (License Must Be Technology- Neutral): Không có sự dự trù nào của giấy phép dành cho một công nghệ riêng hay một kiểu giao diện nào đó
-

Lợi ích phần mềm mã nguồn mở

- Được phát triển bởi một cộng đồng nhiều người nhờ đó có thể tìm ra các lỗi một cách dễ dàng và update bản mới nhanh chóng.
- Cách phân phối của phần mềm mã nguồn mở giúp nhiều người có điều kiện tiếp cận với chúng hơn. Nhất là đối với các nước đang phát triển, nơi mà giá phần mềm dành cho phần bảo trì, bảo hành luôn là gánh nặng

Phần mềm mã nguồn mở thông dụng

- Linux, Ubuntu,..
- office: OpenOffice,..
- IDE
- Webserver: apache, ngnix,
- Ngôn ngữ lập trình và các sản phẩm: php, Python, java,
  - CMS
  - Framework
- Quản lý version
- Quản lý project

Chương 2:

MÔ HÌNH PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN MỞ

Nội dung

- Giới thiệu
- Mô hình phát triển phần mềm truyền thống
- Mô hình phát triển PMMNM
- Sự khác biệt giữa mô hình phát triển phần mềm truyền thống và PMMNM
- Động cơ của người phát triển PMMNM
- Môi trường phát triển PMMNM
- Cộng đồng mã nguồn mở

Giới thiệu

- Một PMMNM là một phần mềm, vì thế nó được phát triển trong một dự án phát triển phần mềm, với một ngoại lệ: Là một dự án nhóm mà các thành viên của nhóm có thể chưa bao giờ gặp nhau.
- Các qui trình công nghệ phần mềm (CNPM) hay qui trình phát triển phần mềm truyền thống có ứng dụng được vào cho phát triển PMMNM hay không ?
- Có những mô hình phát triển PMNM nào?

Giới thiệu

- Mô hình xây dựng phần mềm nguồn mở là một mô hình độc đáo và nó được hiện thực hoá với sự ra đời của Internet và sự bùng nổ thông tin do Internet mang lại.
- Phép so sánh nhà thờ và chợ trời thường được dùng để đối lập mô hình phát triển phần mềm nguồn mở với các

Mô hình phát triển PM truyền thống

- Là mô hình xây dựng nhà thờ thời trung cổ.
- Đòi hỏi tính chặt chẽ trong các công đoạn quản lý, thiết kế và xây dựng Có sự quản lý chặt chẽ như:
  - Quản lý ai là người viết các phần mã lệnh, phương pháp mà họ tích hợp các gói mã lệnh
  - Định nghĩa rõ ràng một cấu trúc quản lý
  - Xây dựng một kế hoạch chính xác về lịch phát hành mã lệnh
  - Các nhóm lập trình làm việc riêng rẽ, theo sự quản lý và kế hoạch chi tiết, cho đến khi sản phẩm được hoàn thành và chương trình phần mềm công bố.
  - Khi đã phát hành, phần mềm được coi là hoàn chỉnh và không cỏ nhiều công sức để chỉnh sửa nó về sau

Mô hình phát triển PMMNM

- Là mô hình xây dựng chợ: Không có một thiết kế ban đầu rõ ràng, không có một qui trình quản lý chính thức.
- Sử dụng một chính sách lỏng lẻo trong việc:
  - Phát hành mã nguồn
  - Quản lý ai là người viết mã nguồn cho việc sửa lỗi và cho các chức năng mới Nguyên tắc căn bản: Viết mã lệnh thường xuyên, phát hành thường xuyên.
- Đây là mô hình tăng trưởng: Tự phát triển khi phần mềm đạt đến một số chức năng cơ bản nào đó. Mô hình phát triển gồm 2 giai đoạn:

Giai đoạn khởi đầu:

- Phần mềm chưa đủ các chức năng để có thể hấp dẫn các lập trình viên khác
- Cần một số tài trợ về tài chánh để có thể đạt đến điểm có thể sử dụng được, sẽ chuyển sang giai đoạn tăng trưởng

Giai đoạn tăng trưởng: Nhận được thêm nhiều chức năng mới và các gói sửa lỗi từ cộng đồng

Khác biệt giữa mô PMTT & PMNM

- Có sự khác biệt về tài nguyên cho việc phát triển phần mềm trong 2 mô hình: Nhân lực, Máy tính, Kênh phân phối, ...
- Đối với CNPM truyền thống: Khan hiếm và tốn kém, vì thế cần quản lý nhân lực chặt chẽ và cần xây dựng môi trường để bảo vệ tài nguyên này
- Đối với PMMNM: Lập trình viên là tình nguyện. Sử dụng hạ tầng cơ sở (ví dụ máy tính) sẵn có. Phân phối qua Internet

Động cơ của người phát triển PMMNM

- Phần lớn PMMNM là kết quả của
  - Niềm đam mê lập trình,
  - Kết quả của một số bài tập trong các chương trình đại học,
  - Vì Lợi ích cộng đồng
- Một số công ty dùng như mô hình kinh tế để:
  - Thâm nhập thị trường đã bị thống trị bởi công ty khác
  - Phát hành sản phẩm nhanh hơn nhờ sử dụng lại PMMNM

Môi trường phát triển PMMNM

- Môi trường phát triển PMMNM cần cung cấp các chức năng sau:
  - Các kênh truyền thông (communication channel)
  - Các cơ sở dữ liệu về lỗi (Bug database)
  - Hệ thống quản lý mã nguồn (Version control)

Các kênh truyền thông

- Gồm các thành phần như: Website, Mailing list, Bug Tracker, IRC, Wiki, Newsletters, Files bundled with code
- Cung cấp các thông tin như:
  - Mô tả và mục tiêu dự án
  - Tin tức và bản phân phối mới nhất
  - Tài liệu người dùng
  - Tài liệu thiết kế
  - Vật phẩm quảng cáo
  - Kế hoạch và lịch trình tương lai
  - Chuẩn lập trình
  - Quyền sở hữu tập tin/môđun
  - Danh sách lỗi đang mở (và đóng)
  - Cách thức để lấy mã; đóng góp vào mã nguồn
  - Liên kết tới những kênh giao tiếp khác

Quản lý lỗi

- Lỗi (bugs) là không tránh khỏi, cần có phương tiện để người dùng thông báo lỗi.
- Có nhiều cách thức quản lý lỗi: mailing list, lưu trong cơ sở dữ liệu.
- Sử dụng mailing list có hạn chế:
  - Dễ bị mất, khó quản lý
  - Lập trình viên mới không biết các lỗi trước đây
- Sử dụng lưu trữ lỗi trong cơ sở dư liệu:
  - Dễ dàng trong tìm kiếm lỗi
  - Dùng cho các mục đích khác nữa: yêu cầu tính năng, cải tiến, bản vá lỗi
  - Ví dụ: Bugzilla, Mantis, Trac, Google Code

Hệ thống quản lý mã nguồn

- Hệ thống quản lý mã nguồn (Version control System) sử dụng
  - Lưu trữ mã nguồn trực tuyến
  - Theo dõi vết thay đổi trên mã nguồn
  - Trộn những đụng độ trên một tập tin

Cộng đồng mã nguồn mở

- Có 4 dạng đại diện cho các mẫu tổ chức cơ bản trong cộng đồng mã nguồn mở: nhà cung cấp duy nhất, cộng đồng lập trình viên, cộng đồng người sử dụng, các trung tâm
- Các dự án nguồn mở là một nhà cung cấp duy nhất
  - Các công ty thường phát triển phần mềm trong nội bộ và cuối cùng, vì bất kỳ lý do gì, phát hành sản phẩm dưới một giấy phép nguồn mở.
  - Mục tiêu ban đầu của các dự án nguồn mở đó là bán các giấy phép sở hữu độc quyền ở phạm vi toàn cầu.
  - Một số người chỉ trích nói rằng các dự án nguồn mở một nhà cung cấp duy nhất đó không thực sự là các dự án nguồn mở vì có một công ty duy nhất đứng đằng sau chúng kiểm soát toàn bộ cộng đồng.

Cộng đồng mã nguồn mở

Các cộng đồng các lập trình viên

- Các dự án nguồn mở được các cộng đồng các lập trình viên quản lý đại diện cho dự án nguồn mở điển hình. Chúng hoặc được một cá nhân hoặc một tổ chức như một công ty phần mềm, một cơ quan nhà nước hoặc một trường đại học khởi xướng.
- Những người đóng góp tham gia vào qui trình phát triển vì những động lực đa dạng và đi theo các mục tiêu khác nhau
- Một số lập trình viên đã tham gia vì các động lực về lý tưởng hoặc khác của bản thân. Họ tham gia trong cộng đồng vì họ được thuê trong một công ty nguồn mở hoặc vì họ sở hữu một doanh nghiệp CNTT đang phục vụ các khách hàng.

Cộng đồng mã nguồn mở

Các cộng đồng người sử dụng

- Các cộng đồng người sử dụng là tương tự với các cộng đồng các lập trình viên theo cách là sự kiểm soát được phân tán giữa những người tham gia đóng góp khác nhau.
- Tuy nhiên, trong các cộng đồng người sử dụng và các tổ chức của người sử dụng đầu cuối sản phẩm phần mềm đó sở hữu bản quyền của phần mềm.
  - Phần mềm hoặc từng được phát triển nội bộ ở một người sử dụng phần mềm và sau đó nó đã được phát hành như là phần mềm nguồn mở.
  - Hoặc phần mềm đó đã được tạo ra dưới một giấy phép nguồn mở như là công việc theo hợp đồng của một nhà cung cấp bên ngoài.
  - Những người sử dụng phần mềm đó là những người chủ sở hữu sản phẩm phần mềm đó. Việc sở hữu bản quyền mã đó cho phép những người sử dụng định nghĩa dạng giấy phép nguồn mở.
- Trong vai trò của họ như là những người sử dụng phần mềm thì hoạt động chính của cộng đồng đó là xác định các yêu cầu chung và triển khai chúng hoặc bằng sự phát triển trong nội bộ, hoặc bằng việc hợp đồng với các công ty bên ngoài.

Cộng đồng mã nguồn mở

Các trung tâm năng lực nguồn mở

- Các tổ chức nguồn mở đang tồn tại với nhiều dạng khác nhau như
- các trung tâm năng lực nguồn mở. Chúng hành động như là các nền tảng trung lập về sản phẩm và nhà cung cấp với mục tiêu tạo thuận lợi cho sử dụng phần mềm nguồn mở trong các cơ quan nhà nước, các công ty tư nhân, các tổ chức phi chính phủ (non-governmental organization NGO), …
- Các trung tâm năng lực nguồn mở lấp khoảng trống giữa những người sử dụng phần mềm, các nhà cung cấp CNTT và các thành viên cộng đồng nguồn mở độc lập
- Các trung tâm năng lực nguồn mở được quản lý thành công tồn tại khắp thế giới. Tổ chức khởi xướng, cấu trúc thành viên, các hoạt động, kích cỡ và việc cấp vốn … có thể khác nhau. Tuy nhiên, chúng tất cả đều làm việc với mục tiêu chung để cải thiện môi trường và các điều kiện chung để sử dụng và phát triển các phần mềm nguồn mở

Chương 3:

HẠT NHÂN LINUX

Nội dung

- Hệ điều hành Unix
- Lịch sử của hệ điều hành Linux
- Hạt nhân Linux (Linux Kernel)
- Hệ điều hành Linux (Linux Operating System)
- Các thành phần của một hệ điều hành Linux
- Kiến trúc hạt nhân Linux
- Những khác biệt của Hạt nhân Linux so với Unix
- Phiên bản hạt nhân Linux (Linux Version)
- Mã nguồn của hạt nhân Linux
- Những lý do các công ty hỗ trợ cho việc phát triển Linux Kernel

Hệ điều hành Unix

UNIX®

Hệ điều hành Unix

- Unix là một hệ điều hành ra đời từ phòng thí nghiệm Bell Labs của AT&T.
- Dự án được dẫn dắt bởi Ken Thompson và Dennis Ritchie, 2 nhà khoa học máy tính nổi tiếng.
- Phiên bản đầu tiên của Unix được ra đời vào tháng 3 năm 1971.
- AT&T chia sẻ Unix cho những tổ chức giáo dục, hay tổ chức thương mại bên ngoài, từ đó dẫn đến sự ra đời của nhiều phiên bản Unix khác nhau.
- Nổi bật nhất trong số đó là phiên bản giáo dục được biết đến rộng rãi với cái tên Berkeley Software Distribution, hay BSD.

Hệ điều hành Unix

- Unix và các phiên bản rẽ nhánh từ Unix ban đầu đều là close source và bị ràng buộc bản quyền
- Mặc dù phiên bản chính thức của Unix, BSD đã dừng phát triển từ lâu, thế nhưng những di sản mà chúng để lại là rất lớn cho đến ngày hôm nay.
- Rất nhiều hệ điều hành, từ close source cho đến open source đều dựa trên 2 nhánh này.
- Phiên bản thương mại, close source nổi tiếng, thành công nhất là MacOS của Apple.
- MacOS cũng như các hệ điều hành khác của Apple hiện nay là iOS, watchOS, và tvOS đều được dựa trên nền tảng của BSD

Lịch sử hệ điều hành Linux

- Linux là tên gọi của một hệ điều hành máy tính tương tự như Unix. Linux cũng là tên hạt nhân của hệ điều hành.
- Phiên bản Linux đầu tiên do Linus Torvalds viết vào năm 1991.
- Một cách chính xác, thuật ngữ "Linux" được sử dụng để chỉ nhân Linux, nhưng tên này được sử dụng một cách rộng rãi để miêu tả tổng thể một hệ điều hành giống Unix (còn được biết đến dưới tên GNU/Linux) được tạo ra bởi việc đóng gói nhân Linux cùng với các thư viện và công cụ GNU (tập hợp một số lượng lớn các phần mềm như máy chủ web, các ngôn ngữ lập trình, các hệ quản trị cơ sở dữ liệu, các môi trường làm việc desktop, và các ứng dụng thích hợp cho công việc văn phòng như OpenOffice...)

Linux Kernel

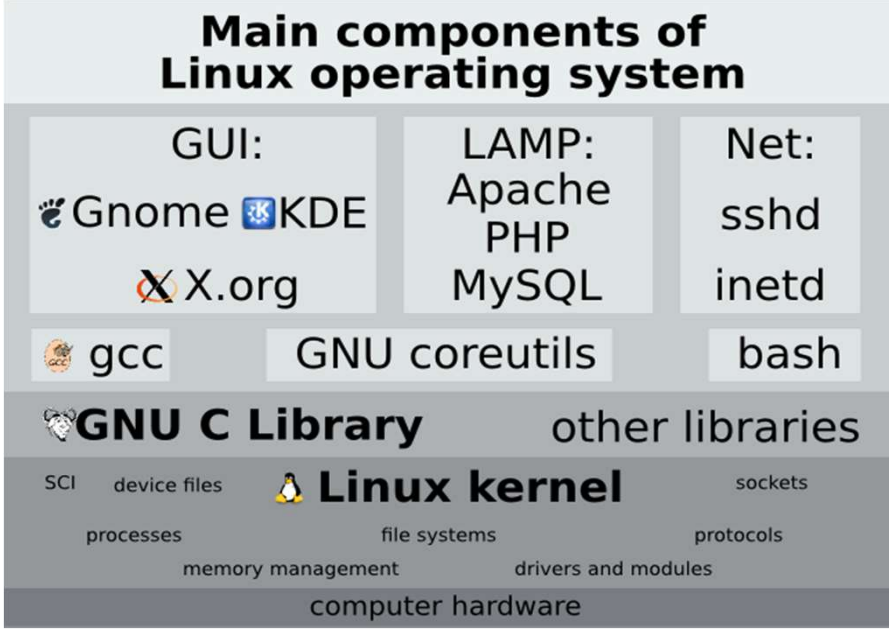
- Là phần cốt lõi nhất của một hệ điều hành được tạo ra bởi Linus Torvald, 1991, phát hành dưới license GPL.
- Linux chỉ là một thành phần của hệ điều hành, thành phần hạt nhân (Kernel) cốt lõi nhất. Có nhiệu vụ: Trừu tượng hóa các thiết bị phần cứng, giới thiệu một máy ảo cho các chương trình người dùng; Hỗ trợ đa nhiệm (multi tasking) và hỗ trợ giao tiếp liên quá trình.

Linux Operating System

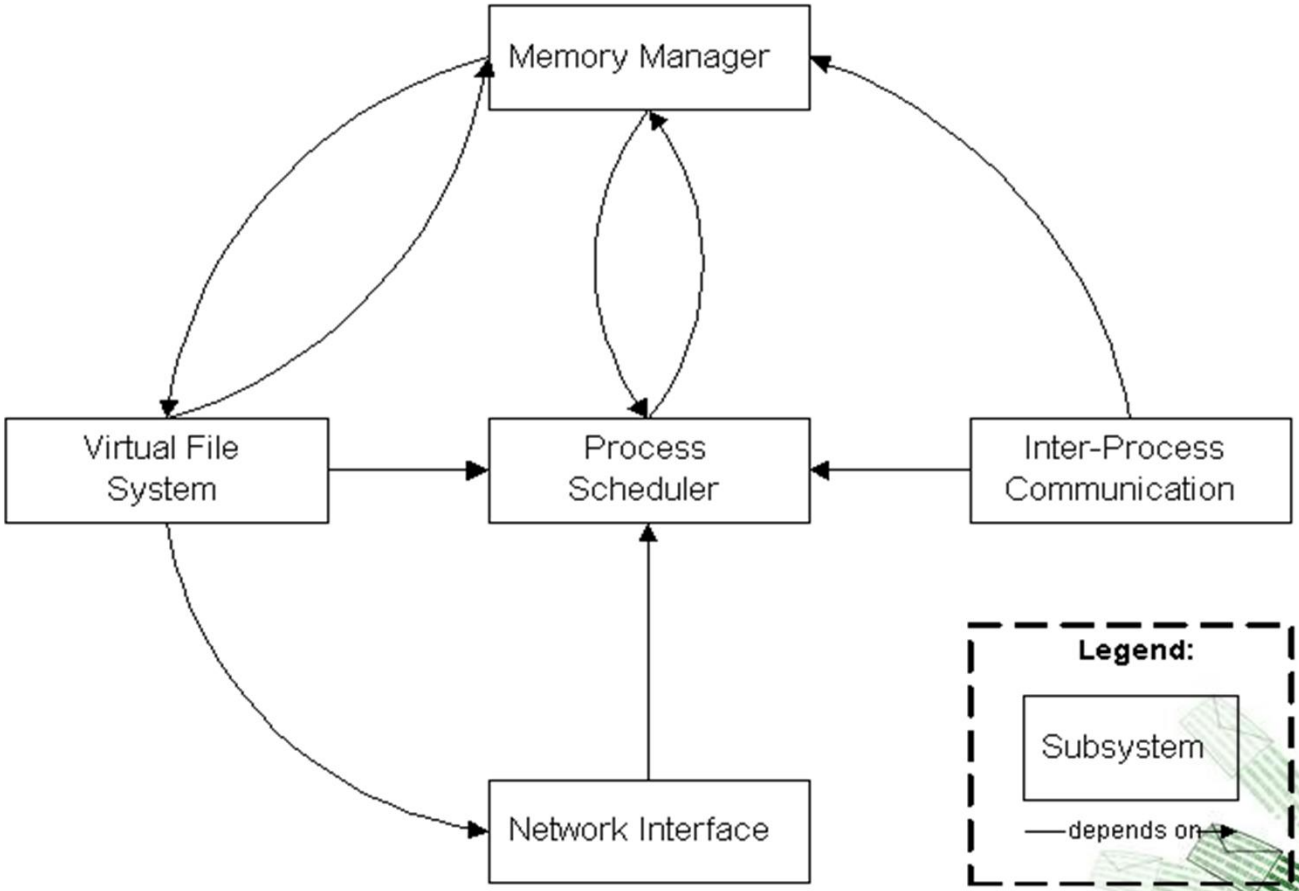
- Là các hệ điều hành sử dụng hạt nhân Linux
- Được gọi với tên Bản phân phối Linux (Linux Distribution), gọi tắt là Linux Distro
- Được phát hành bởi các nhà phân phối hệ điều hành (Linux Distributor)
- Có hơn 500 bản phân phối Linux. Các bản phân phối phổ biến nhất gồm Ubuntu, Fedora, OpenSuSe, Debian, Mandriva, LinuxMint, PCLinuxOS, Slackware, Gentoo Linux, CentOS

Các thành phần của Linux

- Hạt nhân Linux.
- Trình điều khiển thiết bị.
- Bộ khởi động.
- Cửa sổ lệnh hoặc giao diện người dùng đồ họa.
- Các tiện ích về tập tin và hệ thống...



Kiến trúc hạt nhân Linux



Kiến trúc hạt nhân Linux

Hạt nhân Linux gồm 5 thành phần cơ bản sau:

- Bộ định thời : Điều khiển việc truy cập đến CPU
- Bộ quản lý bộ nhớ: Đảm bảo nhiều tiến trình cùng sử dụng bộ nhớ máy tính một cách an toàn; cung cấp cơ chế bộ nhớ ảo
- Hệ thống tập tin trừu tượng: Trừu tượng hóa những chi tiết khác biệt của các loại thiết bị bằng cách giới thiệu một giao diện tập tin chung cho tất cả các thiết bị
- Giao diện mạng: Cung cấp truy cập đến nhiều chuẩn mạng và những loại thiết bị mạng khác nhau.
- Giao tiếp liên quá trình: Hộ trợ cơ chế giao tiếp giữa các tiến trình trên cùng một máy tính

Những khác biệt của Hạt nhân Linux so với Unix

- Hỗ trợ nạp động các mođun của kernel
- Hỗ trợ đa bộ xử lý đồng bộ (Symetrical MultiProcessor)
- Là kernel theo kiểu trưng dụng (Preemptive)
- Hỗ trợ đa luồng
- Hỗ trợ mô hình thiết bị hướng đối tượng, gắn nóng, hệ thống tập tin trên không gian người dùng
- Tự do (Free)

Phiên bản hạt nhân Linux (Linux Version)

- Có hai loại loại phiên bản Linux kernel: Phiên bản ổn định (Stable) và phiên bản phát triển (Development).
- Stable (ổn định) là phiên bản ở mức sản phẩm phù hợp cho việc triển khai rộng rãi.
- Development (phát triển) là phiên bản thử nghiệm với nhiều cải tiến được đưa vào.

Phiên bản hạt nhân Linux (Linux Version)

- Tiến trình phát triển các phiên bản diễn ra như sau:
- Đầu tiên các tính năng mới được tạo ra và thêm vào phiên bản Development của Linux kernel.
- Qua thời gian phiên bản development này được trưởng thành, và đến thời điểm tuyên bố đóng băng các tính năng: không cho thêm mới tính năng, chỉ cho chỉnh sửa tính năng đã có.
- Khi phiên bản development được xem là ổn định mã nguồn sẽ được đóng băng: chỉ chấp nhận các hiệu chỉnh lỗi.
- Phiên bản phát triển sẽ được phát hành như phiên bản stable đầu tiên của chuỗi phiên bản stable mới..

Tình hình phát triển hạt nhân Linux

- Về bức tranh tổng thể: Từ năm 2005, hơn 5000 các nhà phát triển của gần 500 công ty tham gia vào việc xây dựng Linux kernel.
- Cộng đồng phát triển vững mạnh về cả số lượng và năng suất. Số lượng mã nguồn thêm vào kernel tăng từng ngày.
- Về mô hình phát triển: Dựa trên « loose, time-based release model», Nhanh chóng đưa các tính năng mới đến cho người dùng, giảm sự cách biệt giữa các phiên bản.
- Rất nhiều cá nhân, công ty tham gia việc phát triển mã nguồn cho Linux kernel

Những lý do các công ty hỗ trợ cho việc phát triển Linux Kernel

- Để Linux có thể chạy trên phần cứng của họ và thu hút người dùng Linux: IBM, Intel, SGI, MIPS, Freescale, HP, Fujitsu, etc...
- Để chứng tỏ khả năng của họ để thu hút khách hàng sử dụng bản phân phối của họ: Red Hat, Novell, và MontaVista,...
- Linux là một thành phần trong các sản phẩm (video, tele set, mobilphone) của họ: Sony, Nokia, and Samsung
- Để xây dựng ứng dụng trên nền Linux và họ muốn phiên bản mới tiếp tục hỗ trợ ứng dụng của họ

Chương 4:

HỆ ĐIỀU HÀNH LINUX

Nội dung

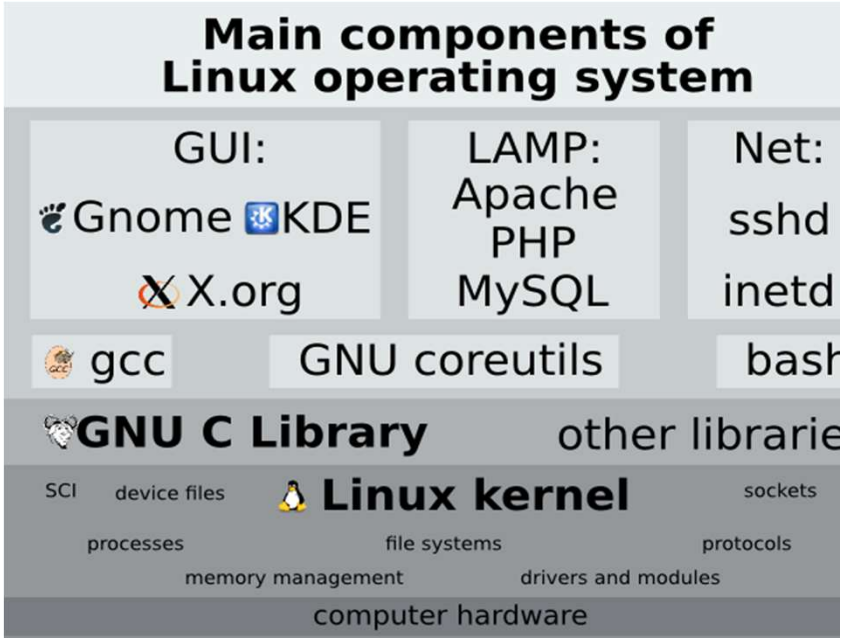
- Hệ điều hành Linux
- Các thành phần một hệ điều hành Linux
- Lý do để chọn hệ điều hành Linux
- Làm việc trên một hệ điều hành Linux
- Các loại tập tin
- Chuẩn phân cấp hệ thống tập tin (FHS-Filesystem Hierarchy Standard)
- Đường dẫn (path)
- Một số thư mục đặc biệt
- Các lệnh cơ bản
- Bộ thông dịch lệnh

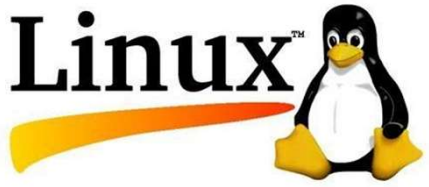
Hệ điều hành Linux

- Là các hệ điều hành sử dụng hạt nhân Linux.
- Được gọi với tên Bản phân phối Linux (Linux Distribution), gọi tắt là Linux Distro
- Được phát hành bởi các nhà phân phối hệ điều hành (Linux Distributor)
- Có nhiều bản phân phối Linux: Ubuntu, Fedora, OpenSuSe, Debian, Mandriva, LinuxMint, PCLinuxOS, Slackware, Gentoo Linux, CentOS

Thành phần hệ điều hành Linux

- Hạt nhân Linux.
- Trình điều khiển thiết bị.
- Bộ khởi động.
- cửa sổ lệnh hoặc giao diện người dùng đồ họa
- Các tiện ích về tập tin và hệ thống...





Lý do để chọn Linux OS

- Ứng dụng: Nhiều ứng dụng sẵn dùng trên Linux (miễn phí lẫn thương mại): văn bản, đồ họa, đa phương tiện, Internet, bảo mật, quản trị, máy chủ …
- Ngoại vi: Hỗ trợ nhiều chủng loại thiết bị ngoại vi, hỗ trợ nhanh chóng các thiết bị ngoại vi mới
- Phần mềm: Tồn tại một lượng lớn các phần mềm dưới dạng mã nguồn hoặc mã thực thi
- Nền: Hỗ trợ nhiều kiến trúc máy tính: Intel, Alpha, MIPS, Motorola, 64bits system, IBM S/390, SMPs
- Bộ giả lập: Cho phép chạy các ứng dụng của các hệ điều hành khác như MS-DOS, Windows, Macintosh
- Máy ảo: Bộ quản lý máy ảo cho phép chạy nhiều máy ảo với những hệ điều hành khác nhau trên cùng một máy tính thật (máy chủ)

Lý do để chọn Linux OS

- Hệ điều hành chuẩn: Dùng như hệ điều hành cho những nhà sản xuất phần cứng khác nhau.
- Đa người dùng & Đa tác vụ
- Tương thích: Hơn 95% mã nguồn được viết bằng C , độc lập thiết bị, nên có thể dịch để dùng cho nhiều loại máy khác nhau: Máy chủ, máy để bàn, di động,
- POSIX (Portable Operating System Interface for Computer Environments): Cho phép ứng dụng phát triển trên Linux có thể dùng trên các hệ thống khác như UNIX
- Miễn phí, mã nguồn mở & tự do: Tiết kiệm chi phí, không phụ thuộc nhà phát triển ứng dụng

Làm việc trên HĐH Linux

- Cần được nhà quản trị máy tính Linux cung cấp một tài khoản biểu hiện bằng một tên đăng nhập (login name/username) và một mật khẩu (password)
- Thực hiện thao tác đăng nhập (login/logon) vào máy tính Linux bằng giao diện đồ họa hoặc dòng lệnh.

Các loại tập tin

- Tập tin là một khái niệm trừu tượng để chỉ các thiết bị có thể ghi hoặc đọc dữ liệu vào/ra như đĩa cứng, màn hình, con chuột, …
- Có 3 loại tập tin dước Linux:
  - Tập tin bình thường: là các tập tin chương trình hoặc tập tin chứa dữ liệu, văn bản
  - Thư mục
  - Các tập tin là các thiết bị ngoại vi

Chuẩn phân cấp hệ thống tập tin

- Chuẩn phân cấp hệ thống tập tin (FHS-Filesystem Hierarchy Standard) Là một tài liệu mô tả cách sắp xếp các thư mục trên hệ thống Linux. FHS được phát triển để cấp một khuôn mẫu chung nhằm giúp cho việc phát triển các ứng dụng mà không phụ thuộc vào bản phân phối Linux. FHS mô tả các thư mục sau:

  / : Thư mục gốc
  /boot: Các tập tin tĩnh cần thiết cho tiến trình khởi động
  /dev : Các tập tin thiết bị
  /etc : Các tập tinh cấu hình hệ thống và các ứng dụng
  /lib : Các thư viện chia sẻ và các môdule của hạt nhân
  /mnt : Điểm gắn nối các hệ thống tập tin một cách tạm thời
  /opt : Nơi tích hợp các gói chương trình ứng dụng
  /sbin: Các tập tin thực thi cần thiết cho hệ thống
  /tmp : Nơi chứa các tập tin tạm
  /usr : Hệ phân cấp thứ cấp
  /var : Dữ liệu biến đổi

Đường dẫn (path)

- Đường dẫn là một chuỗi các tên thư mục ngăn cách nhau bởi ký tự '/', kết thúc đường dẫn có thể là tên một tập tin.
- Đường dẫn tuyệt đối: là đường dẫn bắt đầu bằng thư mục gốc '/'; Ví dụ: /home/user1/Desktop
- Thư mục hiện hành: là một vị trí trên cây thư mục Ví dụ: /home/user1
- Đường dẫn tương đối: là đường dẫn được tính bắt đầu từ thư mục hiện hành

Một số thư mục đặc biệt

- Tên thư mục và tập tin có phân biệt chữ hoa và chữ thường
- Các thư mục đặc biệt
  - Thư mục gốc ký hiệu /
  - Thư mục hiện hành ký hiêu là . (một chấm)
  - Thư mục cha ký hiệu .. (hai chấm)
  - Thư mục cá nhân (home directory) ký hiệu ~: Mỗi người dùng có một thư mục cá nhân nơi mà người dùng có toàn quyền (thêm, sửa, xóa tập tin thư mục).

Cài đặt phần mềm trên Linux

- Phần mềm cho Linux thường có ở dưới những dạng sau:
  - Trong bộ đĩa cài đặt (thường với những bản phân phối lớn như Redhat, openSuse, Mandriva…)
  - Trên trang web của nhà sản xuất.
  - Trên các repository (gọi tắt: repo) là các nơi chứa phần mềm tập trung trên mạng dành riêng cho một hệ thống nào đó
- Các cách cài đặt: Có nhiều cách cài đặt
  - Sử dụng Add/Remove
  - Sử dụng apt-get với cửa sổ dòng lệnh
  - Synaptic: vỏ giao diện cho apt-get
  - Cài đặt trực tiếp từ file .rpm và .deb
  - Cài đặt từ GUI

Các lệnh cơ bản với thư mục

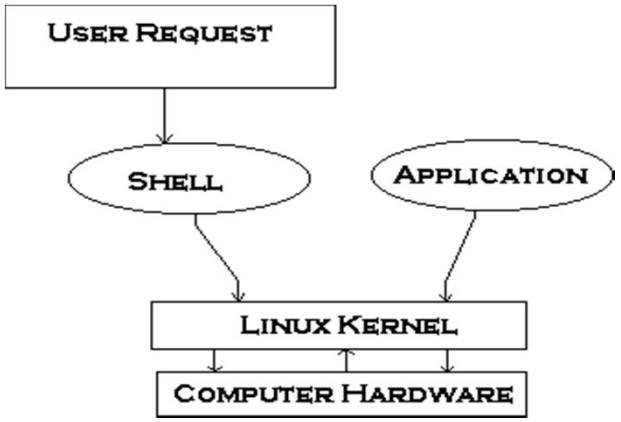
- Thao tác trên cửa sổ dòng lệnh sẽ là cách nhanh và thuận lợi trong quản lý tập tin, thư mục, phần mềm.
- sudo: viết tắt của “SuperUser Do”, cho phép thực hiện các tác vụ yêu cầu quyền quản trị hoặc quyền root.
- pwd: Xem thư mục hiện hành:
- ls: Xem nội dung thư mục ls [dir]
- cd: Chuyển thư mục: cd newdir
- mkdir: Tạo thư mục: mkdir newdir
- cp –r: Sao chép thư mục
- rm : Xóa thư mục
- …

Các lệnh cơ bản với tập tin

- cp old-file new-file: Sao chép tập tin
- mv old-name new-name: Đổi tên tập tin
- mv file-name dir-name: Di chuyển tập tin
- ln -s file-name link-name: Tạo liên kết
- touch file-name: Tạo/Cập nhật tập tin
- rm [-f] file-name: Xóa tập tin
- cat file-name: Hiển thị nội dung
- find: Tìm vị trí files trong thư mục nhất định
- grep: tìm kiếm tất cả text thông qua tập tin nhất định.
- head: xem nhanh các dòng đầu tiên của bất kỳ file văn bản nào
- tail: xem nhanh các dòng cuối cùng của bất kỳ file văn bản nào

Bộ thông dịch lệnh - Shell

- Shell là chương trình người dùng đặc biệt, cung cấp giao diện cho người dùng sử dụng các dịch vụ hệ điều hành. Shell nhận các lệnh có thể đọc được từ người dùng và chuyển đổi chúng thành thứ mà kernel có thể hiểu được.
- Nó là một trình thông dịch ngôn ngữ lệnh thực thi các lệnh được đọc từ các thiết bị đầu cuối vào như keyboard hoặc từ file.
- Shell được bắt đầu khi người dùng đăng nhập hoặc khởi động terminal.
- Shell được chia làm 2 loại: Command Line Shell, Graphical shell



Command Line Shell

- Shell có thể được truy cập bởi người dùng bằng cách sử dụng command line interface. Một chương trình đặc biệt có tên Terminal trong linux/ macOS hoặc Command Prompt trong Windows OS, được cung cấp để nhập vào các lệnh có thể đọc được của người dùng sau đó nó được thực thi. Kết quả được hiển thị trên Terminal.
- Làm việc với command line shell cho phép người dùng lưu trữ các lệnh trong một file và thực thi chúng cùng nhau. Với tính năng này, công việc lặp đi lặp lại nào có thể xử lý tự động. Các tệp này thường được gọi là batch file trong Windows và Shell Script trong Linux / macOS

Command Line Shell

```
vivek@dellm6700:~$ wc -c /etc/passwd
2010 /etc/passwd
vivek@dellm6700:~$
vivek@dellm6700:~$ wc -c /etc/passwd | awk '{print $1}'
2010
vivek@dellm6700:~$
vivek@dellm6700:~$ ls -l /etc/passwd
-rw-r--r-- 1 root root 2010 Jan  9 18:16 /etc/passwd
vivek@dellm6700:~$
vivek@dellm6700:~$ myFileSizeCheck=$(stat -c %s "/etc/resolv.conf")
vivek@dellm6700:~$ printf "My file size = %d\n" $myFileSizeCheck
My file size = 29
vivek@dellm6700:~$
vivek@dellm6700:~$ mfs=$(du --apparent-size --block-size=1  "$fileName" | awk '{ print $1}')
vivek@dellm6700:~$ echo "$fileName size = ${mfs}"
/etc/hosts size = 214
vivek@dellm6700:~$
vivek@dellm6700:~$ fileName="/etc/hosts"
vivek@dellm6700:~$
vivek@dellm6700:~$ mysize=$(find "$fileName" -printf "%s")
vivek@dellm6700:~$ printf "File %s size = %d\n" $fileName $mysize
File /etc/hosts size = 214
vivek@dellm6700:~$
vivek@dellm6700:~$ ls -l $fileName
-rw-r--r-- 1 root root 214 Jul  3  2016 /etc/hosts
vivek@dellm6700:~$
```

Graphical Shells

- Graphical Shells cung cấp phương tiện để thao tác với các chương trình dựa trên graphical user interface (GUI), người dùng không cần nhập lệnh cho mọi hành động.
- Một số shell có sẵn trong các hệ thống Linux: BASH (Bourne Again SHell), CSH (C Shell), KSH (Korn SHell).
- Mỗi shell thực hiện cùng một công việc nhưng hiểu các lệnh khác nhau và cung cấp các hàm dựng sẵn khác nhau.

Shell Script

- Tập các lệnh trong một file và có thể thực thi chúng trong shell, tránh các công việc lặp đi lặp lại. Các file này được gọi là Shell Script hoặc Shell Programs.
- Các Shell script tương tự như batch file trong MS-DOS. Mỗi shell script được lưu với phần mở rộng tệp .sh.
- Một shell script có cú pháp giống ngôn ngữ lập trình.
- Shell script bao gồm các thành phần sau:
  - Shell Keywords – if, else, break etc.
  - Shell commands – cd, ls, echo, pwd, touch etc.
  - Functions
  - Control flow – if..then..else, case and shell loops etc

Shell Script

- Tại sao cần shell script?
  - Tránh các công việc lặp đi lặp lại và tự động hóa.
  - System admins sử dụng shell script để sao lưu thường xuyên
  - Giám sát hệ thống (System monitoring)
- Ưu điểm của shell script
  - Lệnh và cú pháp hoàn toàn giống với lệnh được được nhập trực tiếp trong dòng lệnh. Vì vậy lập trình viên không cần phải chuyển sang cú pháp hoàn toàn khác.
  - Viết và thực thi Shell script sẽ nhanh hơn nhiều.
- Nhược điểm của shell script
  - Dễ xảy ra lỗi tốn kém, một lỗi duy nhất có thể thay đổi lệnh có thể gây hại.
  - Không phù hợp cho các task lớn và phức tạp.

Chương 5:

CÔNG CỤ VÀ KỸ THUẬT HỖ TRỢ PHÁT TRIỂN PMNM

Nội dung

- Tổng quan về vai trò của công cụ và kỹ thuật hỗ trợ
- Công cụ phát triển
- Công nghệ hỗ trợ
- Công cụ hỗ trợ quản lý dự án
- Kỹ thuật hỗ trợ

Vai trò của công cụ và kỹ thuật hỗ trợ

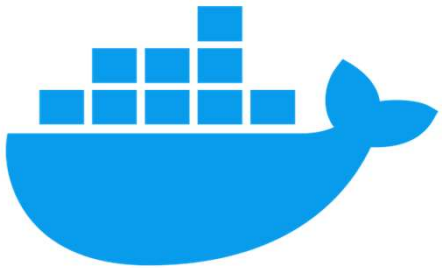
- Tổng quan về vai trò của công cụ và kỹ thuật hỗ trợ

Công cụ phát triển

- Docker
- Eclipse
- Visual code,..
- Sublime text

Docker

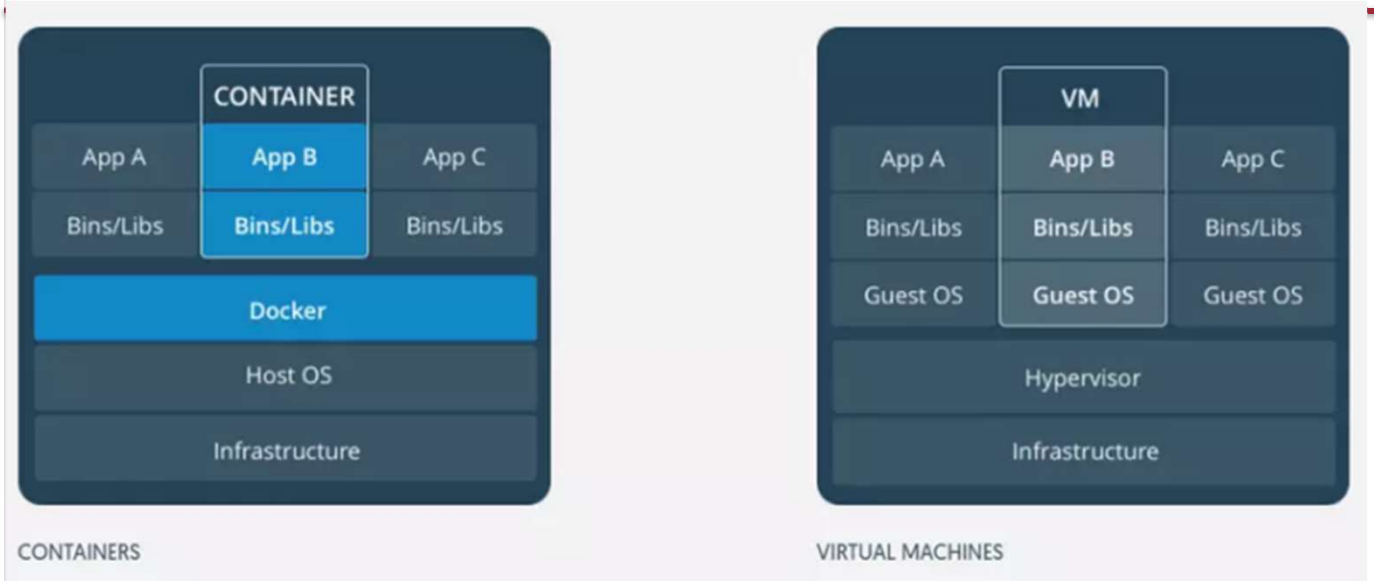
- Một nền tảng được sử dụng để cung cấp cho bạn cách building, deploying và running ứng dụng một cách dễ dàng hơn. Nó có thể thay thế cho các Virtual Machine.
- Hoạt động thông qua cách sử dụng những containers ở trên nền tảng ảo hóa.
- Cho phép tạo các môi trường độc lập và tách biệt để khởi chạy và phát triển ứng dụng. Môi trường này được gọi là container.
- Khi cần deploy lên bất kỳ server nào chỉ cần run container của Docker thì application sẽ được thực thi ngay.
- Lúc đầu, docker được viết bằng Python và hiện tại sử dụng Golang.



Các ứng dụng đặc trưng của Docker

- Thay tế máy ảo.
- Packaking software: Các image của Docker không phụ thuộc vào phiên bản Linux phát hành.
- Enabling a microservices architecture: Docker tạo các container chạy độc lập và giao tiếp với nhau tạo ra hệ thống microservice.
- Modeling networks: Cho phép làm việc với rất nhiều Docker container trên một máy tính local và có thể mô hình hóa nó như một mạng lưới thực tế.
- Reducing debugging overhead: Giảm việc cài đặt các thư viện khi code. Sử dụng các phiên bản khác nhau có thể dẫn tới lỗi giữa các teams khi tích hợp.
- Enabing continuous delivery (CD): Cho phép việc rebuild hệ thống khi sửa đổi bất kỳ dòng code nào đó được chuyển giao ngay tới sản phẩm hoặc server đang hoạt động.

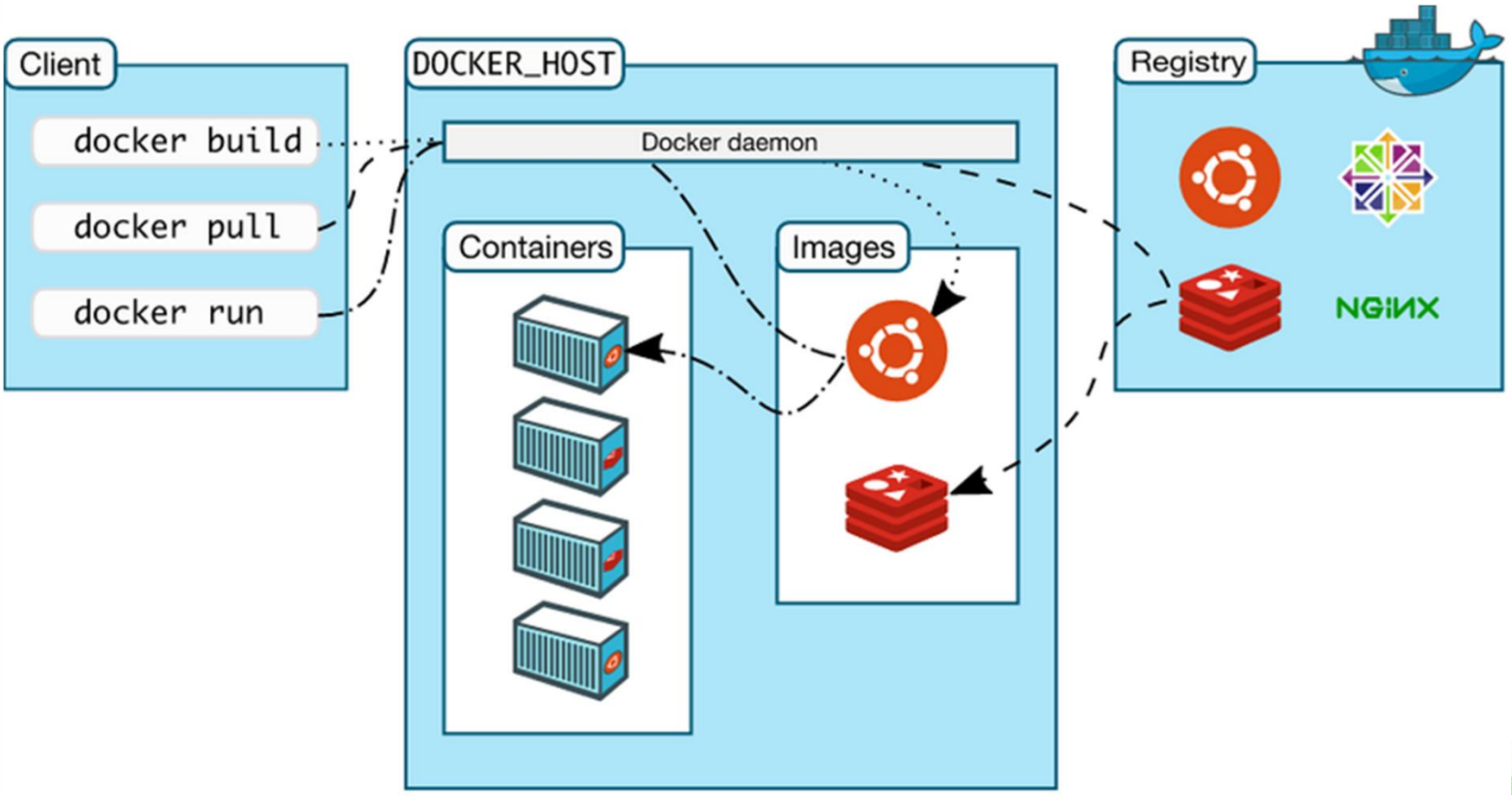
Docker và Virtual Machine (VM)



- VM: Khi build ứng dụng trên VM cần phải cung cấp tài nguyên cho trước cho VM.
  - Nếu không sử dụng hết tài nguyên sẽ gây lãng phí,
  - Nếu tài nguyên không đủ sẽ gây ảnh hưởng đến ứng dụng.
  - Các VM hoạt động độc lập không chia sẻ tài nguyên cho nhau

- Docker, các ứng dụng sẽ sử dụng chung tài nguyên và chỉ chiếm tài nguyên tùy thuộc vào mục đích sử dụng.

Một số khái niệm trong doker



Một số khái niệm trong doker

- Client: là cách chúng ta thao tác với docker qua các câu lệnh command.
- Registry: là nơi lưu trữ của docker image
- Docker Hub: Dịch vụ đăng ký đám mây công cộng do Docker, Inc. điều hành. Nó cho người dùng tìm kiếm và chia sẻ hình ảnh container. Người dùng có thể dễ dàng tìm kiếm, pull, push image lên Docker Hub.
- Docker daemon: Sẽ lắng nghe các API request từ phía docker client và quản lý các đối tượng của docker như: image, container, network, volumes. Chúng có thể giao tiếp với các daemon khác để quản lý các dịch vụ của docker.

Một số khái niệm trong doker

- Images: là khuôn mẫu để tạo ra container. Nó chứa những thứ cần như thư viện, các file cấu hình, biến môi trường để chạy mội ứng dụng nào đó.
- Containers: là instance của image. Khi một phiên bản của image chạy, phiên bản chạy đó gọi là container - (muốn có container phải có image)
- Mối quan hệ giữa image và container tương tự như mối quan hệ giữa class và đối tượng.
- Docker Desktop: là ứng dụng dễ cài đặt cho môi trường window và Mac. Docker desktop bao gồm cả docker client, docker daemon, docker compose, Docker Content Trust, Kubernetes và Credential Helper.

Một số khái niệm trong doker

- Docker Compose là một công cụ hỗ trợ xác định và chạy các ứng dụng multi-container .
  - Docker Compose hoạt động bằng cách áp dụng các quy tắc được xác định trong tệp docker-compose.yaml.
  - Khai báo app's environment trong Dockerfile.
  - Khai báo các services cần thiết để chạy application trong file docker-compose.yml.
  - Run docker-compose up để start và run app
- Dockerfile là một file có tên là dockerfile (không có phần mở rộng), chứa các chỉ thị, mà khi docker gọi tệp tin đó, nó có thể tự động tạo thành các image.

Các lệnh cơ bản trong docker

- Kiểm tra thông tin version
  Cú pháp: docker --version
  Hoặc: docker info
- Download (pull) image from hub (https://hub.docker.com/search?q=&type=image)
  Cú pháp: Docker pull package
  docker pull python
  Docker pull nginx:latest
- Kiểm tra các images đang có
  Cú pháp: docker images –a

```
C:\Users\hungtv>docker images -a
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
nginx        latest    5ef79149e0ec   2 weeks ago   188MB
python       latest    0218518c77be   3 weeks ago   1.02GB
```

Repository tên image trên kho chứa
TAG là phiên bản image, với giá trị latest có nghĩa là bản cuối.
IMAGE ID một chuỗi định danh duy nhất (tên) của image trên hệ thống của mình
.

Các lệnh cơ bản trong docker

- Chạy container: docker run
  Cú pháp: docker container run [OPTIONS] IMAGE [COMMAND] [ARG...]
  Ex: docker run -it --rm -d -p 8080:80 --name web nginx
- Kiểm tra có các container nào đang chạy:
  docker ps

```
C:\Users\hungtv>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED              STATUS              PORTS                   NAMES
cef13f44f697   nginx     "/docker-entrypoint.…"   About a minute ago   Up About a minute   0.0.0.0:8080->80/tcp   web
```

- Liệt kê tất cả các container:
  Cú pháp: docker container ls --all.
- xóa bỏ một container:
  Cú pháp: docker container rm containerid
  Ex: docker container rm cef13f44f697
- Stop container: cú pháp: docker container stop containerid
- Stop all the containers: cú pháp: docker stop $(docker ps -a -q)
- Remove all the containers: cú pháp: docker rm $(docker ps -a -q)

Các lệnh cơ bản trong docker

- Tạo mới một container
  Cú pháp: docker run -it imageid
- Thoát termial vẫn giữ container đang chạy
  CTRL +P, CTRL + Q
- Chạy container đang dừng
  Cú pháp: docker container start -i containerid
- Chạy một lệnh trên container đang chạy
  Cú pháp: docker exec -it containerid command

Cài đặt docker destop

- Trên Mac: https://docs.docker.com/desktop/install/mac-install/
- Trên Linux Kernel OS: https://download.docker.com/linux/

Ví dụ Ubuntu: Sử dụng các lệnh

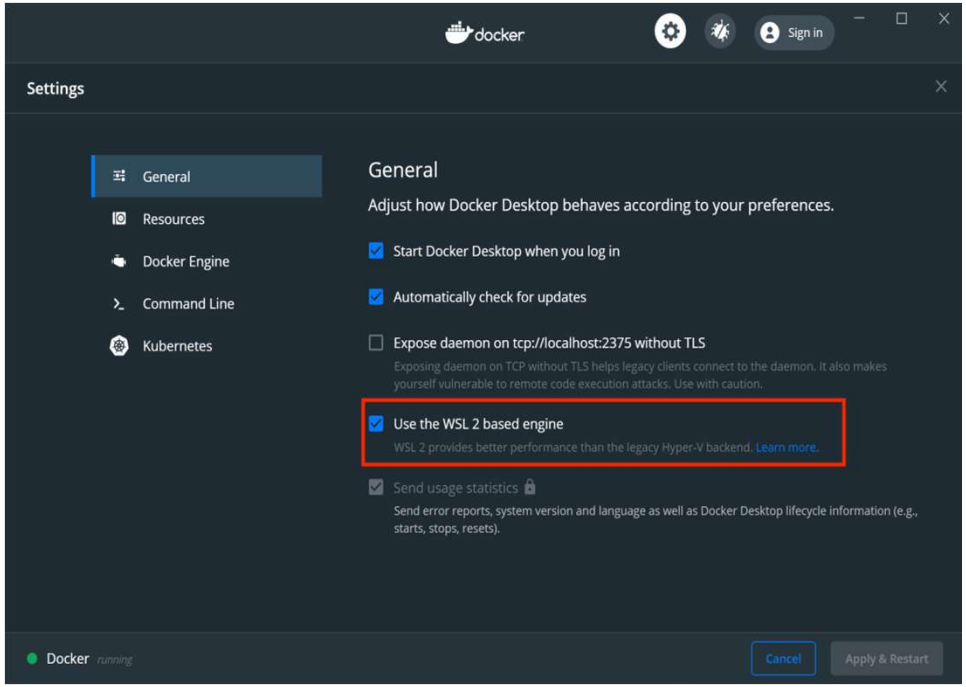
- sudo apt update
- sudo apt install apt-transport-https ca-certificates curl software-properties-common
- curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
- sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"
- sudo apt update
- apt-cache policy docker-ce
- sudo apt install docker-ce
- sudo systemctl status docker

Cài đặt Docker destop trên windows

Link: https://docs.docker.com/desktop/install/windows-install/

Có các lựa chọn khi cài

- Docker chạy trong máy ảo quản lý bởi Hyper-V,
- Docker chạy trong máy ảo VirtualBox.
- Nếu windows đã cài đặt WSL 2, có thể chạy các Container Linux mà không cần những máy ảo trên (native). Cách này ổn định và nhanh hơn sử dụng Hyper-V (hình).



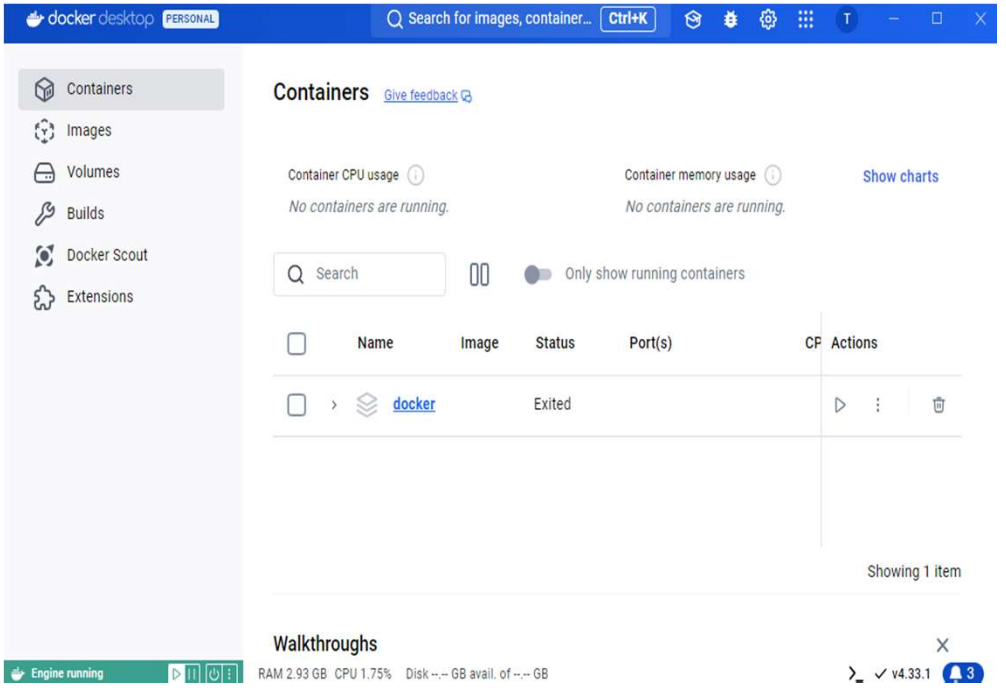

```
C:\Users\hungtv>wsl --list --verbose
  NAME                   STATE           VERSION
* Ubuntu                 Running         2
  docker-desktop         Running         2
```

Cài đặt Docker destop trên windows

- Giao diện Docker Desktop
- Kiểm tra bản distro (OS được phân phối dựa trên Linux Kernel) khi cài trên WSL2:

  wsl --list --verbose

  Hoặc: wsl -l -v




```
C:\Users\hungtv>wsl --list --verbose
  NAME                   STATE           VERSION
* Ubuntu                 Running         2
  docker-desktop         Running         2
```

Ví dụ sử dụng docker

Ví dụ 1: Xây dựng môi trường dev để phát triển ứng dụng web php-mysql dựa theo phần mềm trên máy chủ web php.

- Các cách lựa chọn khi phát triển phần mềm trên máy dev:
  - Cài đặt: Apache (nginx, IIS,…), php (using Extension PDO,…), mysql
  - Cài đặt một lựa chọn: Wamp, Xampp, Mamp, Laragon, EasyPhp, WinginX, Appserv, …
  - Cài đặt máy ảo và cài các phần mềm phù hợp trên VM
  - Sử dụng docker

Sử dụng Docker

- PHP / MySQL / NGINX (Apache) được cung cấp dưới dạng một image và máy chủ có thể chạy nhiều image khác nhau.
- Máy chủ web và máy dev đều chạy cùng một image.
- Deploy image của dev lên máy chủ web, chạy toàn bộ ứng dụng ở đó và trang web hoạt động mà không cần bất kỳ cấu hình thêm nào
- Mỗi image sẽ hoàn toàn tách biệt với image khác trên máy chủ.
- Mỗi website sẽ có cấu hình riêng (nginx, apache, php, mysql). Mỗi website có thể chạy các phiên bản PHP hoàn toàn khác nhau.

Ví dụ sử dụng docker

- Lựa chọn môi trường cần phát triển
  - Webserver: (Apache, NginX,..)?
  - Php (Version 5.x, 7.x, 8.x)? Extension: (PDO or Mysqli,…)
  - Database: (Mysql, MariaDb, ...)
- Thực hiện:
  - Cài đặt WSL2
  - Cài đặt Docker Desktop
  - Thực hiện: Tham khảo https://viblo.asia/p/thiet-lap-moi-truong-phat-trien-php-voi-docker-RQqKLqz0Z7z

Eclipse

- Eclipse là một môi trường phát triển tích hợp dùng cho lập trình máy tín.
- Nó chứa một không gian làm việc cơ sở và một hệ thống plug-in để mở rộng để tùy chỉnh môi trường.
- Phát triển bởi: Eclipse Foundation
- Phát hành lần đầu: 1.0 / 7 tháng 11 năm 2001;
- Nền tảng: Java Platform, Standard Edition, Standard Widget Toolkit, X86-64
- Viết bằng: Java, C

Visual studio code

- Phần mềm IDE miễn phí của Microsoft
- Chạy trên Windows, Linux and macOS.
- Có rất nhiều chức năng hỗ trợ lập trình viên soạn thảo code nhanh chóng.
- Dễ dàng cài đặt, sử dụng và quản lý các extension

Hệ thống quản lý phiên bản

- Hệ thống quản lý phiên bản – (VCS version control system):
  - Một hệ thống ghi nhận và lưu lại sự thay đổi của các file theo thời gian, từ hệ thống đó một file có thể phục hồi quay về trạng thái (phiên bản) ở một thời điểm trước đó.
  - Theo dõi sự thay đổi của một file theo thời gian, ai đã thay đổi, thay đổi vào lúc nào ....
  - Có nhiều hệ thống VCS mà bạn có thể chọn sử dụng như: Concurrent Versions System, Subversion, Git, Mercurial

Hệ thống quản lý phiên bản

- Các khái niệm, thuật ngữ quan trọng
  - Nơi chứa (repository) là một nơi chung chứa mọi bản chính (master copy) của các tập tin khi chúng được thêm vào trong cơ sở dữ liệu của một VCS.
  - Thư mục làm việc (working folder) là một nơi mà chúng ta lấy phiên bản của tập tin và hiệu chỉnh. Thư mục làm việc thường được đặt trên máy client cho từng thành viên trong khi làm việc nhóm. Khi các phiên bản được lấy ra, thư mục này là nơi mặc định mà một VCS sẽ dùng để chứa các bản sao phiên bản đó.

Hệ thống quản lý phiên bản

- Các khái niệm, thuật ngữ quan trọng
  - Dự án, mô đun và tập tin
  - Dự án là một thư mục chứa toàn bộ thông tin của các tập tin của nhóm. Một nhóm có thể thực hiện nhiều dự án khác nhau. Các dự án này có thể chia sẻ các tập tin (shared file). Mỗi VCS thường có một dự án gốc cho mọi dự án.
  - Mô đun là một dự án con (sub-project) nằm trong một dự án mẹ.
  - Tập tin trong dự án có thể thuộc về một hoặc nhiều dự án (shared file).

Hệ thống quản lý phiên bản

- Các khái niệm, thuật ngữ quan trọng
  - Phân nhánh: Khi làm việc nhóm, mọi thành viên đều có một phần mã chung, gọi là nhánh chính (mainline). Và các thành viên đều làm việc với một phần của nhánh chính. Khi có yêu cầu tách một phần mã chung đó, ví dụ như để tạo một phiên bản để thử nghiệm, các thành viên của nhóm phân phối thử nghiệm đó cần mã độc lập với nhánh chính. Giải pháp là dùng phân nhánh.
  - Chia nhánh (branching) là một cơ chế của hệ thống quản lý phiên bản cho phép tách một phần của dự án ra riêng. Phần này sẽ hoạt động độc lập với dự án chung. Khi đó, việc đánh phiên bản cũng thay đổi.

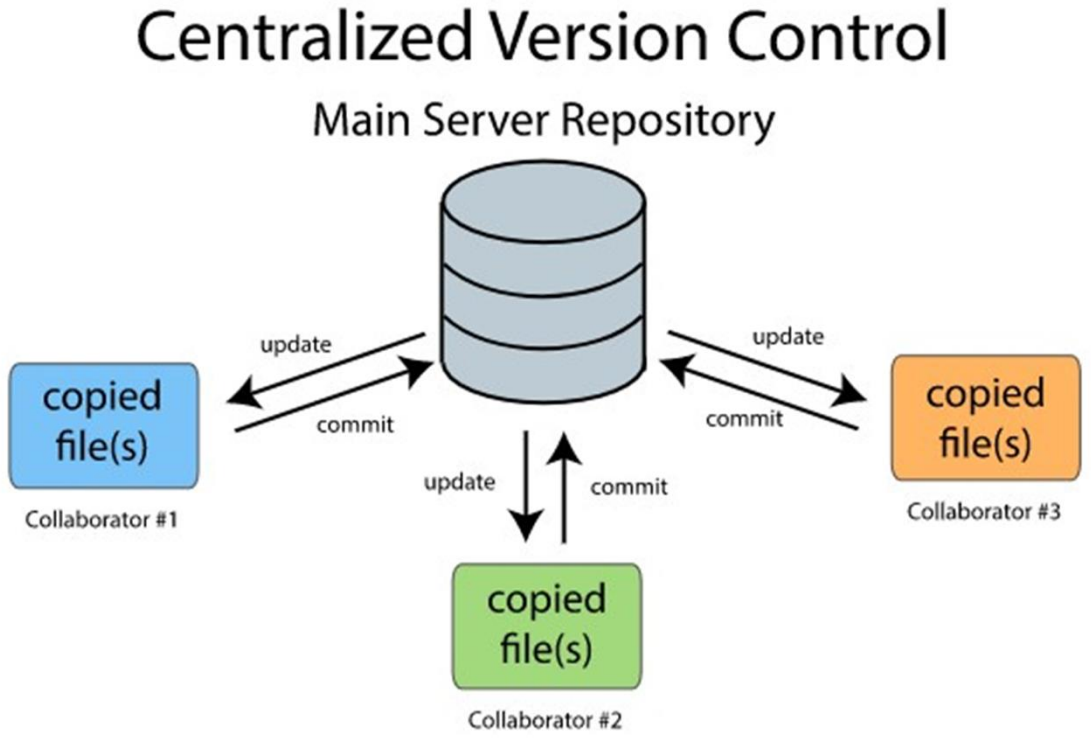
Hệ thống quản lý phiên bản

- Các khái niệm, thuật ngữ quan trọng
  - Ghép tập tin:Có nhiều nhánh (branch) có thể được tách ra làm việc độc lập. Nếu họ trong khi làm thấy rằng có một số lỗi (bug) phát sinh và có ảnh hưởng tới nhánh chính. Họ sẽ làm sao để cập nhật nhanh nhất mã trong nhánh chính và dùng cách ghép tập tin (merge).
  - Ghép tập tin sẽ cho phép loại bỏ được việc phải cắt dán nhiều lần trên các phiên bản khác nhau của hệ thống.

Hệ thống quản lý phiên bản

- Các khái niệm, thuật ngữ quan trọng
  - Phiên bản: Mỗi tập tin có thể có nhiều phiên bản (version). Các phiên bản được đánh số khác nhau. Mỗi lần chúng ta hiệu chỉnh nội dung tập tin và cập nhật vào nơi chứa của VCS, phiên bản sẽ được cập nhật.
  - Một số VCS sử dụng hệ thống phiên bản 1, 2, 3, ... trong khi có một số khác sử dụng hệ thống 1.0, 1.1, 1.2, ... Các VCS thường lưu phiên bản cuối cùng và các thay đổi của các phiên bản trước so với phiên bản cuối đó.
  - Các số phiên bản (như 1.0, 1.1, hay 1.2,...) còn được gọi là revision.

Công cụ quản lý phiên bản SVN



A Centralized Version Control System. Users could check out files they wanted to work on, then commit them once they made their changes.

Công cụ quản lý phiên bản SVN

- Subversion là gì? Subversion (viết tắt SVN) là một hệ thống VCS được giới thiệu vào năm 2000 bởi công ty CollabNet (subversion.tigris.org).
- SVN là Hệ thống kiểm soát phiên bản tập trung (Centralized Version Control System - CVCS)
- Phần mềm:
  - Cho client: TortoiseSVN, Download:http://tortoisesvn.net/
  - Cho server: VisualSVN – Server Download: http://tortoisesvn.net/downloads.htmls

Công cụ quản lý phiên bản SVN

- SVN Subversion giải quyết được vấn đề gì?
  - Khi một nhóm làm việc trên cùng một project, việc nhiều người cùng chỉnh sửa nội dung của một file là điều không thể tránh khỏi. SVN Subversion cung cấp các chức năng để có thể thực hiện việc này một cách đơn giản và an toàn.
  - SVN Subversion được thiết kế với mục đích thay thế hệ thống quản lý phiên bản Concurrent Versioning System (CVS) đã cũ và có nhiều nhược điểm. Subversion có thể được sử dụng để quản lý bất cứ hệ thống phiên bản nào.
  - SVN Subversion là hệ thống quản lý phiên bản mạnh mẽ, hữu dụng, và linh hoạt.
  - SVN Subversion quản lý tập tin và thư mục theo thời gian.
  - SVN Subversion giống như một hệ thống file server mà các client có thể download và upload file một cách bình thường.

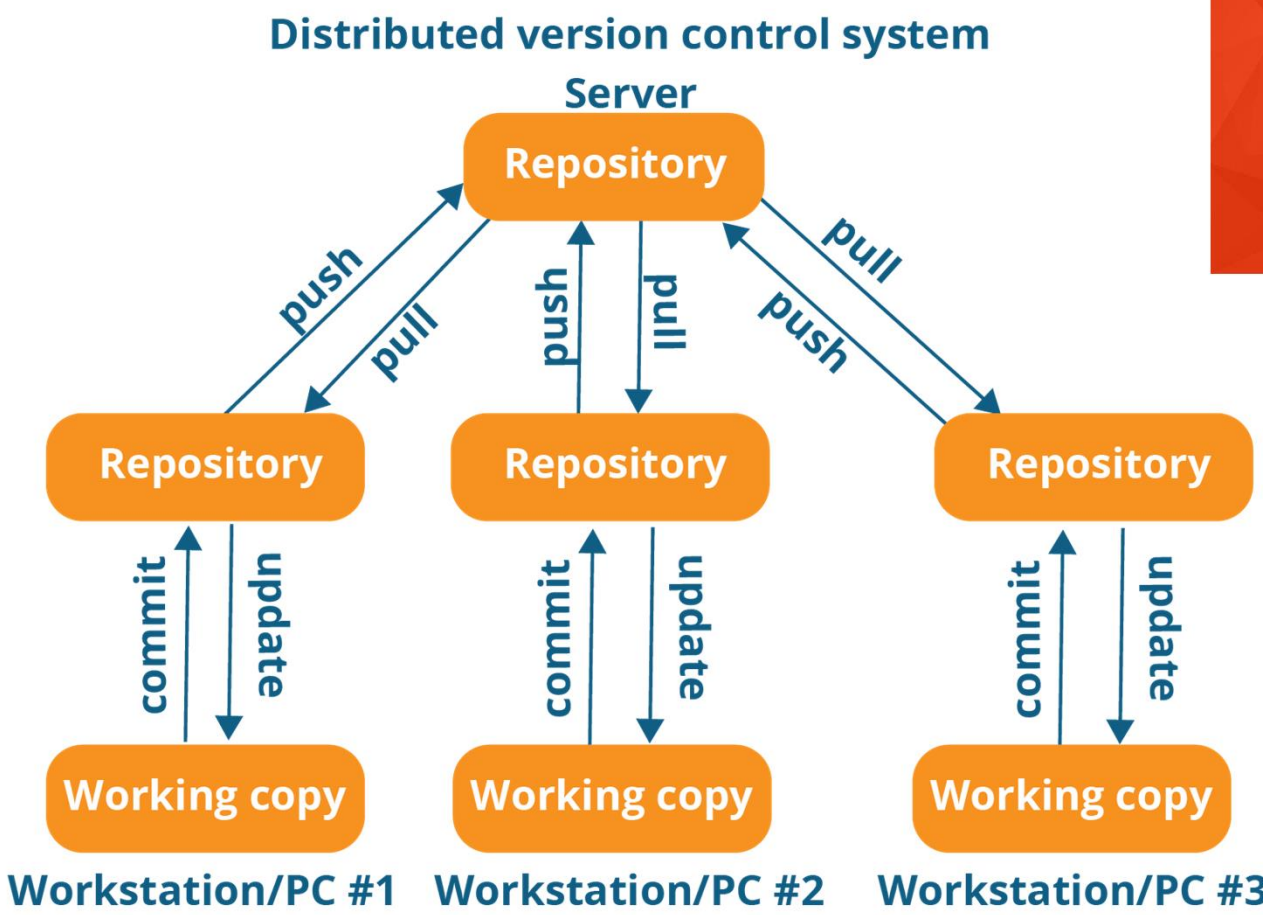
Công cụ quản lý phiên bản SVN

- SVN Subversion giải quyết được vấn đề gì?
  - Điểm đặt biệt của SVN Subversion là nó lưu lại tất cả những gì thay đổi trên hệ thống file: file nào đã bị thay đổi lúc nào, thay đổi như thế nào, và ai đã thay đổi nó.
  - SVN Subversion cũng cho phép recover lại những version cũ một cách chính xác. Các chức năng này giúp cho việc làm việc nhóm trở nên hiệu quả và an toàn hơn rất nhiều.
  - Thông thường, client và server kết nối thông qua mạng LAN hoặc Internet. Client và server có thể cùng chạy trên một máy nếu SVN Subversion có nhiệm vụ theo vết lịch sử của dự án do các nhà phát triển phần mềm phát triển trong nội bộ.

Công cụ quản lý phiên bản SVN

- Giao thức kết nối
  - SVN Subversion hỗ trợ khá nhiều giao thức để kết nối giữa client và server. Ví dụ có thể dùng các giao thức của ứng dụng web như http:// hoặc https://, hay các giao thức của svn như svn:// hoặc svn+ssh://, hoặc nếu phần mềm client và server cài chung trên 1 máy thì có thể dùng file://.
  - Việc cho phép server hỗ trợ giao thức nào phụ thuộc vào lúc cấu hình.
  - Cài đặt SVN Subversion (Client): tool dùng trên Client
  - Cài đặt VisualSVN (Server): tool dùng cho Server
  - Chi tiết cài đặt, sử dụng có thể xem trên website: https://tortoisesvn.net/

Công cụ quản lý phiên bản Git



Công cụ quản lý phiên bản Git

- Git là một hệ thống VCS (Version Control System) dùng để quản lý và kiểm tra các phiên bản mã nguồn khác nhau trong quá trình phát triển mã nguồn.
- Git là phần mềm quản lý mã nguồn phân tán (Distributed) được phát triển bởi Linus Torvalds vào 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. Git là phần mềm mã nguồn mở được phân phối theo giấy phép công cộng GPL2.
- Git có khả năng chạy trên nhiều hệ điều hành khác nhau như Linux, Windows, Mac OSX...
- Download, cài đặt và hướng dẫn sử dụng tại website:

  https://git-scm.com/

Công cụ quản lý phiên bản Git

- Cách quản lý source code của Git:
  - Đối lập với các hệ quản lý sourcecode tập trung: Quản lý phân tán của Git là một repositories không cần có chung một nơi để lưu trữ, mà mỗi thành viên sẽ có một repository ở local của họ.
  - Tất cả thao tác ta làm việc với Git đều ở trên máy local, local repository, khi quyết định đưa những thay đổi đó lên server, sử dụng thao tác "push" nó lên server.
  - Vẫn có thể share thay đổi của cá nhân cho thành viên khác, bằng cách commit hoặc update trực tiếp từ máy cá nhân mà không phải thông qua repositories gốc trên server. Mọi thao tác đều mang theo thông tin history với Git.

Github

- GitHub là một công ty cung cấp dịch vụ lưu trữ các dự án được kiểm soát phiên bản bởi Git
- Địa chỉ: https://github.com/.
- Tất cả các dự án được kiểm soát phiên bản bởi Git và lưu trữ trên GitHub.
- Các thành viên của một dự án đồng bộ code với nhau thông qua một remote repository được lưu trữ trên máy chủ riêng biệt.
- GitHub là một trong những dịch vụ phổ biến nhất để lưu trữ các dự án mã nguồn mở.

Bugzilla

- Bugzilla là hệ thống phần mềm theo dõi lỗi mã nguồn mở, cho phép cá nhân hoặc nhóm các nhà phát triển theo dõi các lỗi xác suất xảy ra trong dự án của họ một cách hiệu quả.
- Đội ngũ kiểm tra chất lượng phần mềm QC (Quality Control) có trách nhiệm quản lí hệ thống này.
- Thông tin download, cài đặt và sử dụng: https://www.bugzilla.org/

Bugzilla

- Chức năng
  - Bugzilla giúp quản lí quy trình sửa lỗi phần mềm miễn phí.
  - Cho phép quản lí quy trình hoạt động cũng như tiến độ test lỗi của từng dự án
  - Cho phép nhiều user làm việc cùng lúc, dễ tìm kiếm và phân bổ công việc cho từng thành viên
  - Cập nhập thông tin cho từng thành viên tham gia dự án thông qua chức năng gửi thư điện tử

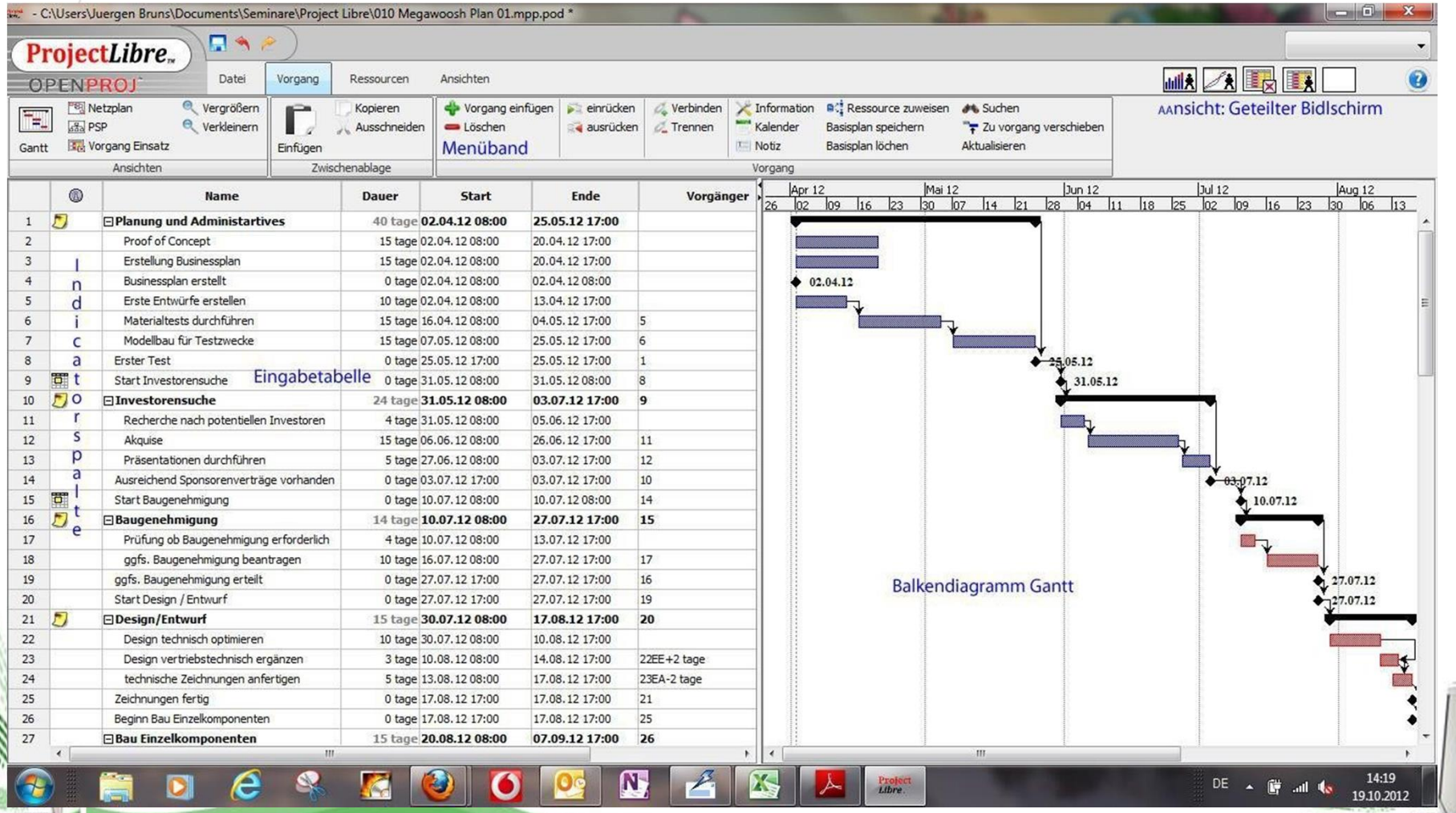
Bugzilla

- Các thành phần của Bugzilla
  - Administration: người quản lí của một Bug
  - Bugzilla-General: tạo, thay đổi và xem bugs
  - Những hoạt động được gửi bởi bugzilla liên quan đến email như post lỗi và sửa lỗi.
  - Query/Buglist: liên quan đến các hoạt động tìm kiếm lỗi và xem buglist.
  - Tài khoản người dùng: các hoạt động quản lí tài khoản người dùng , các truy vấn đã lưu, tạo tài khoản, thay đổi mật khẩu, đăng nhập...
  - Giao diện người sử dụng

Công cụ hỗ trợ quản lý dự án

- Có nhiều công cụ quản lý dự án mã nguồn mở dạng ứng dụng web hoặc ứng dụng destop.
- Một số công cụ quản lý dự án mã nguồn mở trên web:
  - MyCollab: https://www.mycollab.com
  - Odoo: https://www.odoo.com
  - OpenProject: https://www.openproject.org
- Công cụ quản lý dự án mã nguồn mở trên destop: PROJECT LIBRE,..

PROJECT LIBRE



PROJECT LIBRE

- ProjectLibre là một phần mềm quản lý dự án.
- Có hai phiên bản: ứng dụng destop và phiên bản Cloud (sắp phát hành). Phiên bản ứng dụng destop là phần mềm mã nguồn mở và được viết bằng java.
- ProjectLibre thay thế cho phần mềm Microsoft Project - một công cụ quản lý project mã nguồn đóng của Microsoft.
- Bản phát hành đầu tiên là vào tháng 8 năm 2012
- Được InfoWorld trao giải "Phần mềm nguồn mở tốt nhất" và Opensource.com đưa vào danh sách "10 dự án nguồn mở hàng đầu“ vào 2/2016.
- Chạy trên nhiều hệ điều hành: Windows, BSD, Mac, Solaris, Linux,
- Website: https://www.projectlibre.com/

PROJECT LIBRE

- Các tính năng chính của ProjectLibre:
  - Khả năng tương thích với Microsoft Project.
  - Biểu đồ Gantt
  - Giản đồ hệ thống
  - Biểu đồ WBS- Work Breakdown Structure / RBS - Work Breakdown Structure
  - Tính toán chi phí
  - Xây dựng các biểu đồ nguồn tài nguyên
  - ...
- Cài đặt và sử dụng
  - Tham khảo trên website: https://www.projectlibre.com/

Công nghệ hỗ trợ

- Hệ điều hành: Linux
- Dịch vụ webserver: Apache, JSP, Samba,..
- Ngôn ngữ lập trình: gcc (C/C++), java, Perl, Php,..
- Hệ quản trị cơ sở dữ liệu: MySQL, PostgreSQL,..

Dịch vụ web

- Có nhiều thành phần liên quan đến phát triển các dịch vụ ứng dụng web.
- Các thành phần này, có rất nhiều là mã nguồn mở và chiếm một vị trí khá lớn trong các ứng dụng web.
- Một số các thành phần mã nguồn mở tiêu biểu, theo đánh giá của website https://opensource.com/
  - Web server: Apache, Ngnix, Apache tom cat, Node.js, …
  - Ngôn ngữ lập trình: php, java, python,…
  - Các hệ quản trị CSDL: Mysql, MariaDb,..
  - …

Dịch vụ web server - Apache

- Theo đánh giá của https://opensource.com , top 5 webserver opensource: Apache HTTP Server, NGINX, Apache Tomcat, Node.js,...
- Máy chủ Apache HTTP: Apache cung cấp 52% tất cả các trang web trên toàn cầu và cho đến nay là máy chủ web phổ biến nhất.
- Apache httpd chạy trên Linux và cả trên OS X và Windows.
- Apache httpd sử dụng kiến trúc mô-đun, trong đó các mô-đun bổ sung có thể được tải để mở rộng các tính năng của nó.
- Hướng dẫn cài đặt, sử dụng và download tại https://httpd.apache.org/

Dịch vụ web server- NGINX

- Được phát triển NGINX vào năm 2002, với bản phát hành công khai lần đầu tiên vào năm 2004.
- NGINX đứng thứ hai trong danh sách các máy chủ web nguồn mở theo mức độ sử dụng, chỉ chạy hơn 30% tổng số các trang web.
- NGINX dựa trên kiến trúc hướng sự kiện không đồng bộ để giúp thúc đẩy mục tiêu xử lý các phiên đồng thời lớn
- sử dụng tài nguyên nhẹ và khả năng mở rộng quy mô dễ dàng.
- NGINX được phát hành theo giấy phép giống BSD và không chỉ có thể được triển khai dưới dạng máy chủ web mà còn như máy chủ proxy hoặc bộ cân bằng tải.
- Download: https://nginx.org/

Ngôn ngữ lập trình

- Ngôn ngữ lập trình: Có nhiều ngôn ngữ lập trình (trình biên dịch) được cài đặt sẵn trên các hệ điều hành nhân linux hoặc có thể download miễn phí và dễ dàng cài đặt tại các trang web được cung cấp.
- gcc (C/C++):
- Java
- Perl
- Php
- ,..

Hệ quản trị cơ sở dữ liệu

- Nhiều hệ quản trị CSDL mã nguồn mở hay miễn phí ra đời, hỗ trợ và phát triển mạnh, đóng góp nhiều cho việc phát triển các phần mềm nguồn mở sử dụng CSDL.
- Các CSDL có thể kể đến là: MySQL, MariaDb, PostgreSQL,..

Hệ quản trị cơ sở dữ liệu

- MySQL, MariaDb: hai trong số những hệ quản trị cơ sở dữ liệu đang được sử dụng rộng rãi nhất trên thế giới hiện nay.
- Sử dụng MariaDB hay Mysql đều có 2 phiên bản thương mại và cộng đồng, tuy nhiên với MySQL bản thương mại (Entrerprise) và bản cộng đồng (miễn phí) và MySQL được Oracle mua lại năm 2009 và được bảo trì và phát triển bằng đội ngũ của Oracle.
- MariaDB là mã nguồn mở được vận hành bởi cộng đồng được tách ra khi mysql bị mua lại, đứng đằng sau là công ty Monty Program.
- MariaDB hoàn toàn tương thích với Mysql. Do vậy, có thể sử dụng chuyển đổi giữa hai dòng sản phẩm này.

Hệ quản trị cơ sở dữ liệu

PostgreSQL:

- PostgreSQL là một hệ thống quản trị cơ sở dữ liệu quan hệ-đối tượng (object-relational database management system) mã nguồn mở được phát triển dựa trên POSTGRES 4.2 tại phòng khoa học máy tính Berkeley, Đại học California.
- PostgreSQL có thể chạy được trên nhiều nền tảng khác nhau như Mac OS X, Solaris và Windows.
- PostgreSQL là một phần mềm mã nguồn mở miễn phí. Mã nguồn của phần mềm khả dụng theo license của PostgreSQL, một license nguồn mở tự do.
- PostgreSQL không yêu cầu quá nhiều công tác bảo trì bởi có tính ổn định cao.

Các phần mềm web Opensource

- Các công nghệ hỗ trợ phong phú, cộng đồng phát triển mạnh, nhiều phần mêm web mã nguồn mở chất lượng được tạo ra trong rất nhiều lĩnh vực, ứng dụng.
- Các phần mềm dưới đây được đánh giá bởi website: https://opensource.com
- Framework: Ruby on Rails, Cake PHP, Spring Web MVC, ASP.NET, Django, Laravel…
- CMS (Content Management System)
  - Tin tức: Joomla, Wordpress, Drupal…
  - Thương mại điện tử: WooCommerce (WordPress), Magento Community Edition, PrestaShop, OpenCart, osCommerce, Zen Cart,…
  - Đào tạo trực tuyến: moodle, sakai,…
  - …

Tài liệu tham khảo

- [1] Bài giảng - Phát triển phần mềm mã nguồn mở, TS Ngô Bá Hùng, Đại học Cần Thơ
- [2] Succeeding with Open Source, Addison Wesley, Bernard Golden, 2004
- [3] Producing Open Source Software, Karl Fogel, O'Reilly, 2005
- [4] https://opensource.com