Chương 1: Tổng quan về phát triển nguồn mở

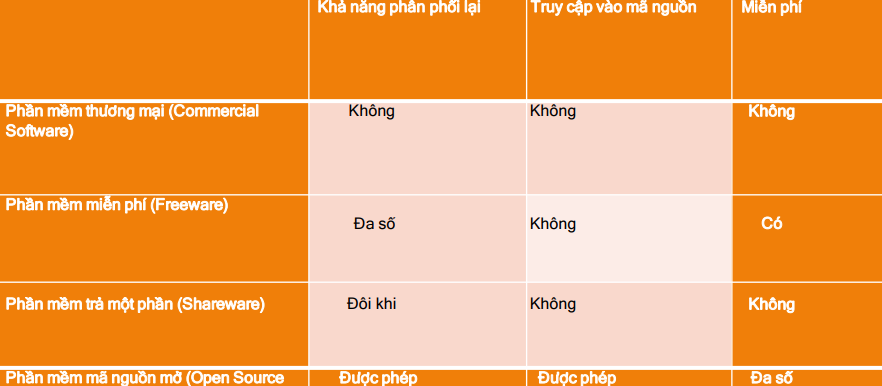
**Khái niệm nguồn mở?**

Phần mềm nguồn mở là phần mềm được phân phối cùng với mã nguồn, do đó, luôn sẵn sàng đối với việc sử dụng, sửa đổi và chia sẻ quyền truy cập. Mã nguồn là một phần của phần mềm mà hầu hết người dùng không bao giờ thấy. Đó là mã được các lập trình viên máy tính thiết lập để kiểm soát hoạt động của một chương trình hoặc ứng dụng. Các lập trình viên có quyền truy cập vào mã nguồn, tiến hành các thay đổi chương trình bằng cách thêm hoặc sửa chữa các phần trong đó. OSS thường bao gồm một giấy phép cho phép các lập trình viên sửa đổi phần mềm để phù hợp nhất với nhu cầu của họ và kiểm soát cách phần mềm có thể được phân phối.

**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. 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.

* **Phần mềm thương mại:** Cho phép người sử dụng khai thác theo những rang buộc đã ghi rõ trong giấy phép.Tuy nhiên bản quyền rất hạn chế,trường hợp có lỗi hay chức năng ko tốt thì ng sử dụng phải chờ cho chủ sở hữu sửa xong
* **Phần mềm miễn phí:** Là các phần mềm có chủ sở hữu. Được phân phối một cách tự do. không đòi hỏi tiền bản quyền.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.Cho phép truy cập vào mã nguồn 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. có thể phân phối lại mà không bị một ràng buộc nào khác. Khả năng phân phối lại. Khả năng truy cập vào mã nguồn. Phí sử dụng phần mềm(Free)



* **Phần mềm tự do:** 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.

**Các phiên bản nguồn mở**

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

**Tiêu chí 1 giấy phép chuẩn**

Tiêu chí (1): Tự do phân phối lại Bản quyền sẽ không hạn chế bất cứ ai bán hoặc cho phần mềm,ko đòi hỏi chi phí

Tiêu chí (2): Mã nguồn ( Source Code) phải được phân phối cùng với mã nguồn được công bố bằng những phương tiện công cộng người ta có thể lấy được mã nguồn với một chi phí sao chép hợp lý nhất

Tiêu chí (3): Sản phẩm kế thừa Giấy phép phải công nhận những sửa đổi và những sản phẩm kế thừa;

Tiêu chí (4) Tính toàn vẹn của mã nguồn của tác giả 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

Tiêu chí (5): Không phân biệt đối xử giữa các cá nhân và các nhóm

Tiêu chí (6): Không phân biệt đối xử với mục đích sử dụng

Tiêu chí (7): Phân phối giấy phép 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 phải thực thi thêm những giấy phép phụ.

Tiêu chí (8): Giấy phép không được dành riêng cho một sản phẩm

Tiêu chí (9): Giấy phép không được cản trở phần mềm khác 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 .Không có sự dự trù nào của giấy phép dành cho một công nghệ riêng hay một kiểu giao diện nào đó

**Các loại mô hình**

**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

**Phát triển**: Là mô hình xây dựng chợ: Không có một thiết kế ban đầu rõ ràng, không có một qui trình quản lý chính thức. 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 chuyến 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 nhau giữa 2 mô hình:**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

**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

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)

**Chương 2: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.

**Khác biệt giữa Linux và 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)

Có hai loại loại phiên bản Linux kernel: Phiên bản ổn định (Stable) và phiên bản phát triển (Development).

**Hệ điều hành Linux**

Là các hệ điều hành sử dụng hạt nhân Linux.Gồm các thành phần: 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...

Chuẩn phân cấp hệ thống tập tin Là một tài liệu mô tả cách sắp xếp các thư mục trên hệ thống 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

**Các cách cài đặt phần mềm trên Linux:**

– 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**

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

**Chương 3:Các ứng dụng**

**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. 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

**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

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

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

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.

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.

**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...

**Cách quản lý source code của Git:**

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**

là một công ty cung cấp dịch vụ lưu trữ các dự án được kiểm soát phiên bản bởi Git.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**

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.

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

Thành phần: 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

**Project Libre**

là một phần mềm quản lý dự án.

Tính năng: 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

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,..