



Chương 2

XU HƯỚNG PHÁT TRIỂN CỦA PHẦN MỀM MÃ NGUỒN MỞ

quangnm@live.com





Quy mô của phần mềm mã nguồn mở

Tại sao lập trình viên chọn mã nguồn mở?

Project mã nguồn mở đầu tiên

Bài tập về nhà

Question

Tổng quan



- ▶ Phần mềm mã nguồn mở đang được các doanh nghiệp áp dụng rộng rãi từ nhiều ngành công nghiệp cho đến các hãng có quy mô khác nhau;
- ▶ Nhìn về tương lai, các công nghệ mới như điện toán đám mây (39%), dữ liệu lớn (35%), hệ điều hành (33%) và IoT (31%) sẽ bị ảnh hưởng bởi phần mềm mở trong vòng 2 đến 3 năm nữa.



Một vài thống kê

- ▶ Tính đến tháng 03/2014, trên trang Web SourceForge.net¹ lưu trữ hơn 430,000 dự án phần mềm mã nguồn mở được phát triển bởi hơn 3.7 triệu nhà phát triển phần mềm;
- ▶ Tính đến tháng 04/2016, trên trang Web GitHub² (github.com) có khoảng 14 người lưu trữ hơn 35 triệu dự án. Đây là website lớn nhất thế giới lưu trữ dữ liệu code.



Nếu bạn học lập trình, hãy học cách sử dụng Github

¹Wikipedia: <https://en.wikipedia.org/wiki/SourceForge>

²Wikipedia: <https://en.wikipedia.org/wiki/GitHub>



Một vài thống kê (tiếp...)

- ▶ Tính đến tháng 06/2016, trình duyệt Web mã nguồn mở Google Chrome giữ trên 60%³ thị phần thị trường trình duyệt Web kể từ khi được ban hành vào tháng 9/2008 và hiện nay đang rất phát triển;
- ▶ Tính đến tháng 01/2016, trình duyệt Web FireFox, được phát triển bởi quỹ phi lợi nhuận Mozilla, chiếm khoảng 10% thị phần sử dụng trình duyệt Web trên toàn thế giới. Tại Cuba, Eritrea và Đức thị phần sử dụng Firefox tương ứng là 85.93%, 79.39%, và 38.36%, lớn nhất trong các loại trình duyệt tại các quốc gia này.

³Wikipedia: https://en.wikipedia.org/wiki/Google_Chrome



Một vài thống kê (tiếp.)

- ▶ Phần mềm mã nguồn mở là một giải pháp tốt cho các tổ chức, cơ quan và chính phủ điện tử.
- ▶ Ở một số nước phát triển đã chuyển sang dùng phần mềm mã nguồn mở trong các cơ quan nhà nước như Pháp, Hàn Quốc, Đức...
- ▶ Việt Nam: từ năm 2015 công ty Linagora là đối tác chính cung cấp các giải pháp xây dựng chính phủ điện tử ở Việt Nam.



Cộng đồng mã nguồn mở

- ▶ Ngày càng đông đảo.
- ▶ Tại sao các lập trình viên này đóng góp mã nguồn mà họ không cần một sự khích lệ về kinh tế nào?



© Can Stock Photo - csp4244980



Động lực bên trong (intrinsic motivation)

Thúc đẩy con người làm một việc vì họ xem đó là điều họ đang quan tâm, nó là điều thú vị, hoặc nó đang là một thách thức đối với mọi người và kể cả chính họ.

- ▶ Cần một giải pháp phần mềm cụ thể,
- ▶ Làm cho vui - thú vui của các lập trình viên,
- ▶ Mong muốn đóng góp cho cộng đồng.



Động lực bên ngoài (extrinsic motivation)

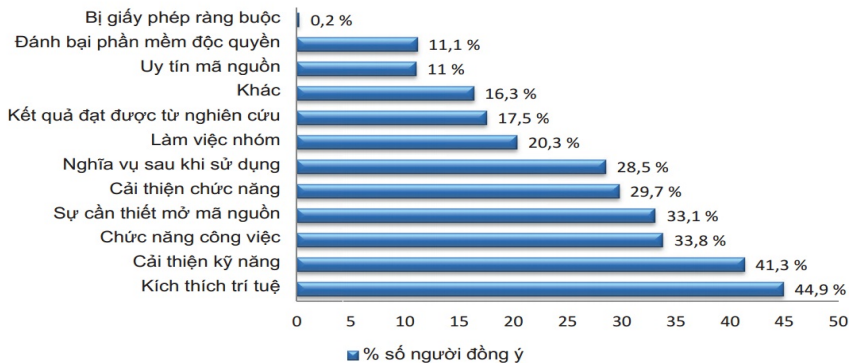
Thúc đẩy con người làm một việc nhằm hy vọng có được một kết quả riêng biệt phục vụ cho mục đích của họ.

- ▶ Tiếp cận mã nguồn mở cho phép các lập trình viên có được những kinh nghiệm và thấy được lợi ích của mã nguồn mở.
- ▶ Cho phép các lập trình viên gia nhập vào cộng đồng phát triển mã nguồn mở giống như việc đi học việc từ thực tế.
- ▶ Tham gia vào các dự án mã nguồn mở có thể là một kênh quảng cáo khả năng của lập trình viên - liên quan đến lợi nhuận trong tương lai của họ.
- ▶ ...



Một số động lực chính

Mục đích tham gia cộng đồng mã nguồn mở của lập trình viên:







Viết chương trình C++ đầu tiên

Thực hiện các bước sau:

1. Mở một thư mục mới.
2. Tạo ra 1 file, tên file là *main.cpp*. (có thể chọn tên khác).
3. Viết một chương trình C++ đơn giản trong file *main.cpp*.



Ví dụ

```
#include <iostream>
using namespace std;
int main(int[] argc, char* argv[])
{
    cout << "Xin chào " << argv[1] << endl;
    return 0;
}
```



Viết chương trình C++ đầu tiên (tiếp)

Thực hiện các bước sau:

1. Lưu lại nội dung file *main.cpp*: Ctrl - S.
2. Mở chương trình Terminal tại thư mục vừa tạo ra.
 - ▶ Thường vào thư mục đó, ấn chuột phải, và chọn Open in *Terminal*
 - ▶ Đang ở thư mục khác, muốn con trỏ của Terminal trở về thư mục hiện tại. Phải làm thế nào?



Viết chương trình C++ đầu tiên (tiếp)

Thực hiện các bước sau:

1. Lưu lại nội dung file *main.cpp*: Ctrl - S.
2. Mở chương trình Terminal tại thư mục vừa tạo ra.
 - ▶ Thường vào thư mục đó, ấn chuột phải, và chọn Open in *Terminal*
 - ▶ Đang ở thư mục khác, muốn con trỏ của Terminal trở về thư mục hiện tại. Phải làm thế nào?



Viết chương trình C++ đầu tiên (tiếp)

1. Biên dịch: nhập nội dung sau vào Terminal.

```
g++ -o main main.cpp
```

2. Thư mục hiện tại xuất hiện file mới: *main*: đây chính là file chạy của chương trình vừa biên dịch.
3. Chạy chương trình: nhập nội dung sau vào Terminal.

```
./main ten_cua_ban
```



Kết quả in ra màn hình là gì?



Ý nghĩa các tham số của hàm *main*

1. `int[] argc`: số lượng các tham số truyền vào.
2. `char* argv[]`: giá trị từng tham số
 - ▶ Nếu chỉ gọi `./main` thì `argc = 1` và `argv[0] = "./main"`;
 - ▶ Nếu chỉ gọi `./main ten_cua_ban` thì

```
argc = 2
argv[0] = "./main"
argv[1] = "ten_cua_ban"
```



Chia sẻ mã lệnh chương trình của bạn

Chia sẻ mã lệnh cho cộng đồng thông qua Github

- ▶ Đăng kí thành viên: <https://github.com/>



Hãy nhớ tên (username) và mật khẩu (password) và email đăng kí của bạn. Bạn sẽ phải dùng chúng ở các bước tiếp theo.



Tạo mới một kho chứa mã lệnh (repository) trên server của Github.

- ▶ Chọn *Repositories* → *New*
- ▶ Chọn tên cho kho chứa (repository) mới.



Bạn có 2 kho chứa dữ liệu:

- ▶ Kho chứa ở server: đã chuẩn bị xong ở bước trước.
- ▶ Kho chứa tại client? Slide tiếp theo...





Cài đặt **git** tại client

1. Cài đặt Github. Trong Ubuntu, bạn gõ dòng lệnh sau vào Terminal:
`apt-get install git`
2. Các thiết lập ban đầu: Bạn cần thiết lập tên và email của mình để mỗi khi đưa mã lệnh lên, server sẽ nhận biết được ai là người đưa lên (một repo có thể có nhiều người tham gia).

```
git config -global user.name richard_feyman  
git config -global user.email richard_feyman@gmail.com
```



Kho chứa tại client

1. Bật Terminal tại thư mục chứa mã lệnh.
2. Khởi tạo một repository tại client:

```
git init
```

3. Thêm 1 file vào repository tại client:

```
git add main.cpp
```

Nếu muốn thêm nhiều file cùng một lúc thì dùng câu lệnh:

```
git add file1.cpp file2.cpp ...
```



Kho chứa tại client

1. Bật Terminal tại thư mục chứa mã lệnh.
2. Khởi tạo một repository tại client:

```
git init
```

3. Thêm 1 file vào repository tại client:

```
git add main.cpp
```

Nếu muốn thêm nhiều file cùng một lúc thì dùng câu lệnh:

```
git add file1.cpp file2.cpp ...
```



Kho chứa tại server

Hiện tại, kho chứa tại client đã có mã lệnh của file main.cpp. Kho chứa tại server vẫn đang rỗng.

Cần đồng bộ hóa 2 kho chứa này.

Nhớ địa chỉ HTTPS được cung cấp tại server.

The screenshot shows the GitHub interface for the repository 'minhquangnguyen / test2'. At the top, there are buttons for 'Unwatch' (1), 'Star' (0), and 'Fork' (0). Below these are tabs for 'Code', 'Issues' (0), 'Pull requests' (0), 'Wiki', 'Pulse', 'Graphs', and 'Settings'. The 'Code' tab is selected, showing a 'Quick setup' section. This section includes a text input field with the URL 'https://github.com/minhquangnguyen/test2.git' and a dropdown menu with 'HTTPS' and 'SSH' options. Below the input field, it says 'We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).'



Kho chứa tại server

Hiện tại, kho chứa tại client đã có mã lệnh của file main.cpp. Kho chứa tại server vẫn đang rỗng.

Cần đồng bộ hóa 2 kho chứa này.

Nhớ địa chỉ HTTPS được cung cấp tại server.

The screenshot shows the GitHub interface for the repository 'minhquangnguyen / test2'. At the top, there are buttons for 'Unwatch' (1), 'Star' (0), and 'Fork' (0). Below these are tabs for 'Code', 'Issues' (0), 'Pull requests' (0), 'Wiki', 'Pulse', 'Graphs', and 'Settings'. The 'Code' tab is selected, showing a 'Quick setup' section. This section has a heading 'Quick setup – if you've done this kind of thing before' and a form with 'or' followed by 'HTTPS' and 'SSH' tabs. The 'HTTPS' tab is active, showing the URL 'https://github.com/minhquangnguyen/test2.git'. Below the form, it says 'We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).'

Kho chứa tại server

Đồng bộ 2 kho chứa qua 1 câu lệnh sau:

1. `git commit -m "first commit"`
2. `git remote add origin`
`https://github.com/minhquangnguyen/test2.git`
3. `git push -u origin master`





Bài tập về nhà



- ▶ Tìm hiểu cách biên dịch gỡ lỗi, sử dụng Terminal.
- ▶ Tạo 1 project C++ gồm:
 1. File *predefine.h*: Khai báo các thông tin về 1 lớp.
 2. File *class.cpp*: Chứa mã lệnh cho lớp đó.
 3. File *main.cpp*: Mã lệnh của hàm main. Sử dụng lớp vừa tạo ra.
 4. Sử dụng Terminal, biên dịch và chạy chương trình.
- ▶ Đưa lên kho chứa Github của riêng bạn.

