**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**LỚP SS004.K23**



**BÀI TẬP 1**

**ĐỀ TÀI: TÌM HIỂU VỀ GIT VÀ GITHUB**

**NHÓM THỰC HIỆN: 19**

**GIẢNG VIÊN HƯỚNG DẪN**

**THẦY: THÁI HUY TÂN**

**TP. HỒ CHÍ MINH, 2020**

**MỤC LỤC**

[Chương 1. GIỚI THIỆU VỀ GIT VÀ GITHUB 3](#_Toc35269278)

[1.1. Git 3](#_Toc35269279)

[1.2. Github 5](#_Toc35269280)

[1.3. Giao diện sử dụng GitHub 5](#_Toc35269281)

[Chương 2. CÁC LỆNH CƠ BẢN CỦA GIT 7](#_Toc35269282)

[2.1. Làm việc ở local repository 7](#_Toc35269283)

[2.2. Liên kết với remote repository 8](#_Toc35269284)

[Chương 3. Làm việc với Git, GitHub 9](#_Toc35269285)

[3.1. Ví dụ thực tế: 9](#_Toc35269286)

[TÀI LIỆU THAM KHẢO 13](#_Toc35269287)

# GIỚI THIỆU VỀ GIT VÀ GITHUB

## Git

Git là tên gọi của một Hệ thống quản lý phiên bản phân tán( *Distributed Version ControlSystem- DVCS*) được phát triển bởi Linux Torvalds vào năm 2005. Hiện nay Git đã trở thành một trong những phần mềm quản lý mã nguồn phổ biến nhất.

DVCS là hệ thống giúp mỗi máy tính có thể lưu trữ nhiều phiên bản khác nhau của một mã nguồn được nhân bản( **clone**) từ kho chứa mã nguồn (**repository**), mỗi thay đổi vào mã nguồn trên máy tính sẽ được **commit** rồi đưa lên máy chủ. Từ một máy tính đã được cấp quyền truy cập khác cung có thể clone lại mã nguồn từ kho chứa hoặc clone lại một tập hợp các thay đổi trên máy tính kia. Trong Git thư mục làm việc trên máy tính gọi là **Working Tree**. Sau đây chúng ta sẽ tìm hiểu một số khái niệm liên quan:

**Repository** được hiểu là kho chứa toàn bộ project bao gồm source code và lịch sử thay đổi cũng như nội dung thay đổi của từng file và từng cá nhân đóng góp vào project đó. Repository của Git gồm 2 loại:

+ Remote repository: Được chia sẽ cho nhiều người và đặt trên server chuyên dụng.

+ Local repository: Được đặt trên máy của bạn và chỉ dành cho một người

dùng.

Ảnh có chứa ảnh chụp màn hình

Mô tả được tạo tự động

**Branch** là nhánh của repository: các nhánh sẽ độc lập với nhau và phát triển một tính năng, không gây ảnh hưởng tới các nhánh khác. Khi các nhánh hợp lại với nhau thì gọi là **merge**, nhánh mặc định là master. Branch trên local repository gọi là local branch, branch trên remote repository gọi là remote branch. Một branch trên local có thể liên kết với một hoặc nhiều branch trên remote hoặc không branch nào cả.

Sau khi viết code để thực hiện chức năng mới ở branch nhỏ xong, bạn merge code vào branch dev, kiểm tra lại, nếu ok thì tiếp tục merge code từ dev vào master.

Git có 3 kho để lưu trữ dữ liệu:

- Working tree: thư mục bạn đang làm việc trên máy tính

- Repository: kho chứa mã nguồn

- Staging area: khu vực trung gian giữa working tree và repository. Đây là khu vực để lưu trữ những thay đổi của bạn trên tập tin để nó có thể được commit.

## Github

Github là một dịch vụ lưu trữ trên web dành cho các dự án có sự dụng hệ thống kiểm soát Git version. Có rất nhiều nhằm lẫn xung quanh việc Git là Github, tuy nhiên không phải như vậy. Github chỉ là một hệ thống giúp bạn xây dựng remote repository, các chức năng của nó được phát triển dựa vào các thư viện của Git.

Ngoài ra Github còn là một trang mạng xã hội dành cho developer, ở đây các lập trình viên có thể theo dõi, học hỏi những người khác, them bình luận cho các dự án phần mềm khác.

## 83Giao diện sử dụng GitHub

- **Edit**: cập nhật mô tả cho project, thay đổi giao diện website

- **Code**: có thể truy cập vào tất cả các tập tin được lưu trữ trong kho. Bạn có thể xem nội dung của từng tập tin, các lần lưu trạng thái (commit), so sánh mã nguồn giữa các lần chuyển giao với nhau, xem và chuyển đổi giữa các nhánh (branch), thêm tập tin vào kho, tạo một yêu cầu gộp (pull request) mới và sao chép (clone) về máy sử dụng GitHub Desktop,…

- **Issues**: issues được dùng để quan sát các ý tưởng, sự cải tiến, lỗi khi làm việc,… bạn cũng có thể ghi nhận những phản hồi của mọi người, báo cáo các lỗi phần mềm hay có thể sắp xếp các vấn đề mà bạn muốn trong kho lưu trữ.

- **Pull** requests: Pull requests giúp bạn có thể hợp tác với người dùng khác trong việc viết code, cho phép bạn trao đổi và đánh giá những thay đổi của mình và người cộng tác trước khi những ý kiến được thống nhất.

**- Actions**: actions cho phép bạn tạo quy trình phát triển phần mềm tùy chỉnh trực tiếp trong kho GitHub của bạn.

- **Projects**: projects la nơi giúp bạn tổ chức và quản lý công việc của mình. Với bẳng dự án, bạn có thể linh hoạt tạo ra các quy trình làm việc phù hợp với nhu cầu của bạn

- **Wiki**: bạn có thể lưu trữ tài liệu của bạn trên wiki để người khác cũng có thể sử dụng và đóng góp vào.

- **Insights**: cung cấp cho bạn các số liệu và báo cáo phân tích

- **Settings**: cho phép bạn thiết lập các tuỳ chọn cho dự án

- **Commit**: lưu các thông tin về sự thay đổi của tệp chỉ khi bạn thực hiện các thao tác như thêm, xóa, sửa tập tin

- **Branch**: giúp bạn làm việc trên các phiên bản khác nhau của kho lưu trữ cùng một lúc, cho phép bạn di chuyển qua lại giữa các trạng thái / phiên bản khác nhau của dự án.

# CÁC LỆNH CƠ BẢN CỦA GIT

## Làm việc ở local repository

- Thiết lập thông tin cá nhân: Trước khi bắt tay vào làm một project thì ta cần điều chỉnh username và email của mình dễ phân biệt với những người khác.

$ git config --global user.name "User Name"

$ git config --global user.email "username@gmail.com"

- Khởi tạo một kho Git để chứa dự án:

$ git init

- Kiểm tra những thay đổi trong thư mục làm việc:

$ git status

- Chuyển những thay đổi sang khu vực trung gian (Staging area)

$ git <tên file>

- Lưu lại lịch sử thay đổi và kèm theo mô tả: từ lịch sử commit trong git, đây bạn có thể khôi phục lại trạng thái cũ của một tập tin bất kì

$ git commit -m <chú thích>

- Tạo một nhánh làm việc mới: Để tránh những tác động không mong muốn lên sản phẩm chính, ta có thể tạo một môi trường làm việc riêng khác và thử nghiệm những thay đổi trên đó

$ git branch <tên nhánh>

-Liệt kê tất cả các nhánh trong git:

$ git branch

-Chuyển đổi giữa các nhánh làm việc:

$ git checkout <tên nhánh>

- Gộp hai nhánh lại với nhau: Lưu ý ta phải chuyển sang nhánh cần được gộp dùng lệnh $ git checkout <tên nhánh> trước khi gộp một nhánh nào đó:

$ git merge <tên nhánh>

- Kiểm tra lịch sử các thay đổi khi sử dụng git:

$ git log

## Liên kết với remote repository

-Kết nối kho git trong máy với Github:

$ git remote add <tên gọi máy chủ> <địa chỉ>

- Sao chép remote repository đã tồn tại về máy tính:

$ git clone <địa chỉ repository>

-Gửi những thay đổi lên remote repository:

$ git push <tên gọi máy chủ> <tên nhánh>

Ví dụ: Gửi nhánh master cục bộ lên máy chủ github mà ta đặt tên là ‘origin’

$ git push origin master

-Cập nhật local repository với những thay đổi mới nhất của kho trên Github:

$ git pull

# Làm việc với Git, GitHub

## Ví dụ thực tế:

Sau khi đã tìm hiểu những lệnh cơ bản của Git, chúng ta sẽ làm quen với môi trường Git qua một ví dụ minh họa đơn giản sau:

A screenshot of a cell phone

Description automatically generated- Tạo thư mục Demo chứa một file **Hello.txt** với nội dung “Hello”.

A screenshot of a cell phone

Description automatically generated- Khởi tạo một Repository trong thư mục **Demo**. Từ đây những thay đổi trong **Demo** sẽ được Git lưu lại và quản lí.

A screenshot of a cell phone

Description automatically generated- Thêm **Hello.txt** vào Repository vừa tạo.

A screenshot of a cell phone

Description automatically generated- Lưu lại sự thay đổi đầu tiên (trong thư mục) và kèm theo chú thích.

- Tạo project **Demo** trống trên GitHub, đây sẽ là nơi để upload repository **Demo** trên máy.A screenshot of a cell phone

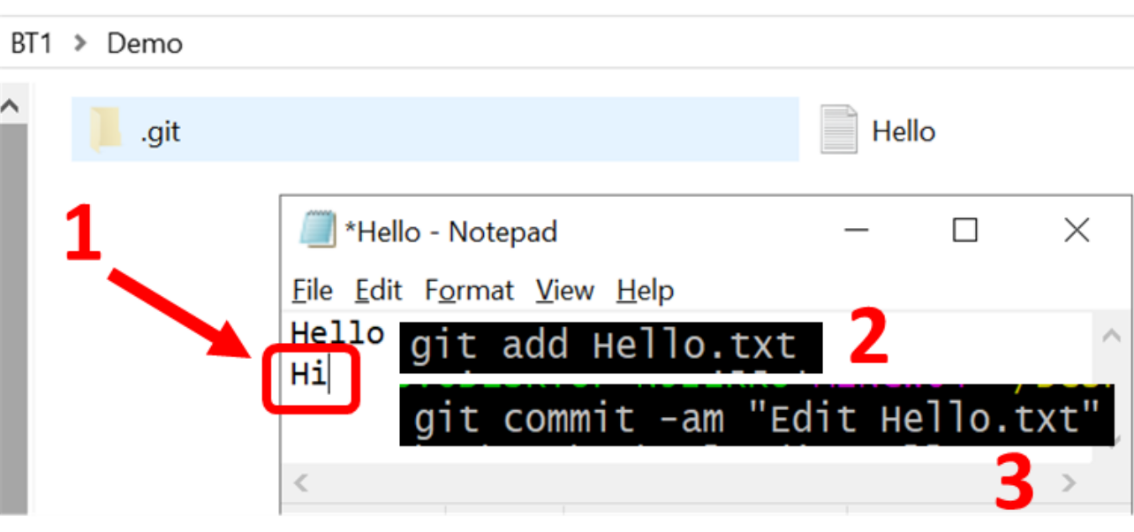
Description automatically generated

- Liên kết local repo với repo trên GitHub và đẩy repo lênA screenshot of a social media post

Description automatically generated

A screenshot of a cell phone

Description automatically generated- Tạo một nhánh làm việc mới (**nhanh1**), điều này sẽ giúp tránh tác động trực tiếp lên sản phẩm chính (nhánh **master**). Ở đây chúng ta có thể tự do làm việc với những tập tin trong **Demo** mà không sợ bị ảnh hưởng tới nhánh chính.

- Thêm một dòng mới (ví dụ ‘Hi’) trong **Hello.txt** và lưu lại thay đổi. Sau đó merge những thay đổi ở **nhanh1** vào nhánh chính (**master**). Quá trình merge sẽ diễn ra thành công

A screenshot of a cell phone

Description automatically generated

- Tuy nhiên, khi ta xóa dòng ‘Hello’ (nội dung ban đầu của file) và đổi thành ‘Goodbye’ thì quá trình merge sẽ xảy ra xung đột. Khi đó, những chỉnh sửa thủ công sẽ được yêu cầu.

- Một trường hợp khác dẫn tới merge conflicts là khi hai người (contributors) cùng chỉnh sửa **Hello.txt** nhưng với nội dung khác nhau. Ví dụ: trong **nhanh1** người A thêm ‘a’ phía dưới dòng ‘Hello’, còn người B thì thêm ‘b’, xung đột sẽ xảy ra khi ta thực hiện liên tiếp merge **nhanh1** **->** **master** và merge **nhanh2** **->master**.

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | #Rất-đơn-giản, “Giải thích công nghệ BLOCKCHAIN rất dễ hiểu,” 14 8 2018. [Trực tuyến]. Available: https://www.youtube.com/watch?v=auMA7jIHYvQ. [Đã truy cập 14 9 2019]. |
| [2] | B. Academy, “Lịch sử Blockchain (Chuỗi khối),” 7 10 2019. [Trực tuyến]. Available: https://www.binance.vision/vi/blockchain/history-of-blockchain. [Đã truy cập 14 9 2019]. |
| [3] | B. V. News, “https://www.binance.vision/vi/blockchain/history-of-blockchain,” 1 10 2019. [Trực tuyến]. Available: https://bitcoinvietnamnews.com/blockchain-la-gi. [Đã truy cập 14 9 2019]. |
| [4] | W. t. Việt, “Blockchain,” 7 10 2019. [Trực tuyến]. Available: https://vi.wikipedia.org/wiki/Blockchain. [Đã truy cập 16 9 2019]. |
| [5] | A. Brownworth, “How Blockchain works,” [Trực tuyến]. Available: http://blockchain.mit.edu/how-blockchain-works. [Đã truy cập 9 22 2019]. |
| [6] | D. Nam, “The Importance and Impact of Blockchain Technology and Decentralization,” 27 8 2017. [Trực tuyến]. Available: https://medium.com/@damonnam/the-importance-and-impact-of-blockchain-technology-and-decentralization-27c0de972233. [Đã truy cập 22 9 2019]. |
| [7] | I. f. t. F. (IFTF), “Understand the Blockchain in Two Minutes,” 18 4 2016. [Trực tuyến]. Available: https://www.youtube.com/watch?v=r43LhSUUGTQ. [Đã truy cập 16 9 2019]. |
| [8] | L. Fortney, “Blockchain Explained,” 25 6 2019. [Trực tuyến]. Available: https://www.investopedia.com/terms/b/blockchain.asp#how-blockchain-works. [Đã truy cập 22 9 2019]. |
| [9] | L. Fortney, “Bitcoin Mining, Explained,” 25 6 2019. [Trực tuyến]. Available: https://www.investopedia.com/terms/b/bitcoin-mining.asp. [Đã truy cập 22 9 2019]. |
| [10] | M. Murray, “Blockchain explained,” 15 6 2018. [Trực tuyến]. Available: http://graphics.reuters.com/TECHNOLOGY-BLOCKCHAIN/010070P11GN/index.html. [Đã truy cập 23 9 2019]. |
| [11] | M. Noor, “A Very Brief History Of Blockchain Technology | Blockchain History 2019,” 21 1 2019. [Trực tuyến]. Available: https://medium.com/@muhammadnoor/a-very-brief-history-of-blockchain-technology-blockchain-history-2019-3c9f9857e085. [Đã truy cập 22 9 2019]. |
| [12] | P. Dash, “Crypto Thoughts — Does it matter who came first Blockchain or Bitcoin?,” 20 2 2018. [Trực tuyến]. Available: https://medium.com/crypt-bytes-tech/crypto-thoughts-does-it-matter-who-came-first-blockchain-or-bitcoin-fcdc0268f8f9. [Đã truy cập 22 9 2019]. |
| [13] | Victor, “WHY IS BLOCKCHAIN TECHNOLOGY IMPORTANT?,” 8 31 2018. [Trực tuyến]. Available: https://crushcrypto.com/why-is-blockchain-important/. [Đã truy cập 22 9 2019]. |