**Thực hành công cụ và môi trường phát triển phần mềm**

**Sáng T6 ngày 02/04/2021**

**Git-cheat-sheet**

Nguyễn Bá Thịnh

Lớp 17TH\_N1\_05

**MAKE CHANGES**

**$ git status:** Để check trạng thái của những file bạn đã thay đổi trong thư mục làm việc. VD: Tất cả các thay đổi cuối cùng từ lần commit cuối cùng.

**$ git diff:** dùng để xem thay đổi(chưa được add) của những file hiện tại.

**$ git add [file]:** Lệnh git add có thể được dùng để thêm file vào index. Ví dụ, command sau có thể thêm một file có tên temp.txt vào thư mục local vào index:

git add temp.txt

**$ git diff –staged:** hiển thị các thay đổi đối với các tệp trong khu vực "staged"

**$ git reset [file]:** Git reset được dùng để quay về một điểm commit file nào đó, đồng thời xóa lịch sử của các commit trước nó.

**$ git commit -m "[descriptive message]":** Để ghi lại việc thêm/ thay đổi file hay thư mục vào repository thì sẽ thực hiện thao tác gọi là Commit. Khi thực hiện commit, trong repository sẽ tạo ra commit (hoặc revision) đã ghi lại sự khác biệt từ trạng thái đã commit lần trước đến trạng thái hiện tại. Commit này đang được chứa tại repository trong trạng thái đã nối tiếp với nhau theo thứ tự thời gian như trong hình dưới đây. Bằng việc lần theo commit này từ trạng thái mới nhất thì có thể hiểu được lịch sử thay đổi trong quá khứ hoặc nội dung thay đổi đó. Khi gọi điện git commitbắt buộc phải gửi kèm tin nhắn descriptive message. Thông báo phải là một mô tả ngắn gọn về những thay đổi được cam kết. Thông báo phải ở cuối lệnh và nó phải được bao bọc trong dấu ngoặc kép " ".

**GROUP CHANGES**

**$ git branch:**  xem toàn bộ các branch mà bạn đang có trong working tree.

**$ git checkout [branch-name]:** Checkout ở đây nghĩa là truy cập kiểm tra mã nguồn trong branch đó để làm việc đấy. Khi thực hiện lệnh thì lúc này đã đổi sang branch “branch-name” rồi.

**$ git merge [branch]:** Để merge một branch bất kì vào branch hiện tại: git merge <branch name>. Giả sử bạn đang ở branch master, bây giờ bạn muốn merge branch task1 vào branch master thì làm như sau:

$ git merge task1

**$ git branch -d [branch-name]:** dùng để xóa nhánh [branch-name].

**CONFIGURE TOOLING**

**$ git config --global user.name "[name]":** thiết lập tên người dùng.

**$ git config --global user.email "[email address]":** thiết lập địa chỉ email.

**$ git config --global color.ui auto:** Dùng để tô màu kết quả xuất ra

**CREATE REPOSITORIES**

**$ git init [project-name]:** Khởi tạo repository có tên là [project-name].

**$ git clone [url]:** Lệnh git clone để sao chép, copy một Git Repo (kho chứa dự án Git) về máy đang local. [url] là copy một Repo từ một Url (https).

Ví dụ: git clone https://github.com/TTXuanDao/LAB3CONGCU.git

**REFACTOR FILENAMES**

**$ git rm [file]:** thuật ngữ rm là viết tắt của loại bỏ,xoá bỏ. Nó được sử dụng để loại bỏ các file riêng lẻ hoặc một bộ sưu tập các file. Chức năng chính của git rm là xóa các file được theo dõi khỏi chỉ mục Git. Ngoài ra, nó có thể được sử dụng để xóa các file khỏi cả thư mục làm việc và chỉ mục dàn dựng.

**$ git rm --cached [file]:** Lệnh trên sẽ xóa một file khỏi hệ thống kiểm soát phiên bản. file đã xóa sẽ vẫn còn trong kho lưu trữ. Bằng cách nào đó, lệnh này sẽ hoạt động như lệnh rm.

Ví dụ: Giả sử chúng ta muốn xóa một file khỏi Git, lấy newfile1.txt để thực hiện thao tác xóa file này, sử dụng lệnh dưới đây:

git rm --cached newfile1.txt

Lệnh trên sẽ xóa file khỏi hệ thống kiểm soát phiên bản, nhưng vẫn có thể theo dõi file trong kho lưu trữ. Nó cũng có thể được thêm lại trên hệ thống kiểm soát phiên bản.

**$ git mv [file-original] [file-renamed]:** Tự động cập nhật chỉ mục cho cả đường dẫn mới và cũ

**REVIEW HISTORY**

**$ git log**: Lệnh “git log” giúp bạn xem lại thông tin lịch sự commit, nhằm giám sát sự thay đổi của dự án. Mặc đinh thi hành git log nó liệt kê các commit theo thứ tự từ mới nhất đến cũ nhất, mỗi commit có các thông tin gồm: mã hash của commit, dòng thông báo, người tạo commit và ngày tạo commit.

**$ git log --follow [file]:** Liệt kê lịch sử phiên bản cho một ﬁle, bao gồm cả tên

**$ git diff [first-branch]...[second-branch]:** kiểm tra sự thay đổi của hai nhánh.

**$ git show [commit]:** Để hiển thị thông tin về bất kỳ git object nào.

**SUPPRESS TRACKING**

**$ git ls-files --other --ignored --exclude-standard:** Hiển thị danh sách file đã bị ignore.

**REDO COMMITS**

**$ git reset [commit]:** Bạn đã đưa một tập tin nào đó vào Staging Area nhưng bây giờ bạn muốn loại bỏ nó ra khỏi đây để không phải bị commit theo

**$ git reset --hard [commit]:** Hủy tất cả các thay đổi đối với một commit

**SAVE FRAGMENTS**

**$ git stash:** Git stash được sử dụng khi muốn lưu lại các thay đổi chưa commit, thường rất hữu dụng khi bạn muốn đổi sang 1 branch khác mà lại đang làm dở ở branch hiện tại. Khi này branch đã trở nên "sạch sẽ" và git status sẽ cho thấy bạn có thể chuyển sang branch tuỳ thích.

**$ git stash pop:** Git cho phép người dùng áp dụng lại các commit trước đó bằng cách sử dụng lệnh git stash pop. Tùy chọn popping xoá các thay đổi khỏi kho lưu trữ và áp dụng chúng vào file làm việc của bạn.

**$ git stash list:** Lệnh “git stash list” được sử dụng để xóa một stash khỏi hàng đợi. Nói chung, nó xóa kho lưu trữ gần đây nhất. Cần thận trọng trước khi sử dụng lệnh này, vì rất khó để hoàn lại nếu đã áp dụng.

**$ git stash drop:** Xóa stash mới nhất khỏi stack.

**SYNCHRONIZE CHANGES**

**$ git fetch [bookmark]:** ]: Tìm nạp các bản sao và tải xuống tất cả các tệp branch vào máy tính của bạn. Sử dụng nó để lưu các thay đổi mới nhất vào kho lưu trữ của bạn. Nó có thể tìm nạp nhiều branch cùng một lúc.

**$ git merge [bookmark]/[branch]:** Kết hợp chi nhánh của dấu trang vào chi nhánh cục bộ hiện tại

**$ git push [alias] [branch]:** để cập nhật kho lưu trữ từ xa với những thay đổi bạn đã thực hiện cục bộ. Nó sẽ trông giống như [chi nhánh] của bạn và đẩy nó thành [chi nhánh] trên điều khiển từ xa, nếu có thể. Nếu ai đó đã đẩy kể từ lần cuối cùng bạn tìm nạp và hợp nhất, máy chủ Git sẽ từ chối việc đẩy của bạn cho đến khi bạn được cập nhật.

**$ git pull:** để lấy về thông tin từ remote và cập nhật vào các nhánh của local repo.