

Hệ thống quản lý phiên bản phân tán - GIT (Git command line)

+ Mục tiêu

- Sử dụng công cụ GIT để quản lý phiên bản của mã nguồn: tạo repo, sao chép, quản lý sự thay đổi mã nguồn, tạo nhánh, trộn các nhánh làm việc, giải quyết xung đột mã nguồn.

+ Hướng dẫn

- Tải công cụ git cho windows: <https://git-scm.com/download/win>
- Cài đặt công cụ
- Khởi động Git Bash
- Cấu hình người sử dụng:
git config - - global user.name "username"
git config -- global user.email "youremail@email.com"
- Tạo mới repo:
git init
- Sao chép mã nguồn từ một repo:
git clone /path/to/repository
- Add & commit:
git add <filename>
git add .
VD: git commit -m "Commit message"
- Kết nối tới một server repo:
git remote add origin <server>
- Đưa thay đổi mã nguồn lên repo server nhánh master:
git push origin master,
- Tạo một nhánh mới
git checkout -b feature_x hoặc git branch feature_x
- Chuyển nhánh:

git checkout master

- Xóa nhánh:

git branch -d feature_x

- update & merge branch:

git pull

git merge <branch>

- Xem sự khác nhau của hai nhánh:

git diff <source_branch> <target_branch>

- Xem lịch sử commit:

git log

git log --author=bob

- Thay thế thay đổi file dưới local thành phiên bản trên repo server:

git checkout -- <filename>

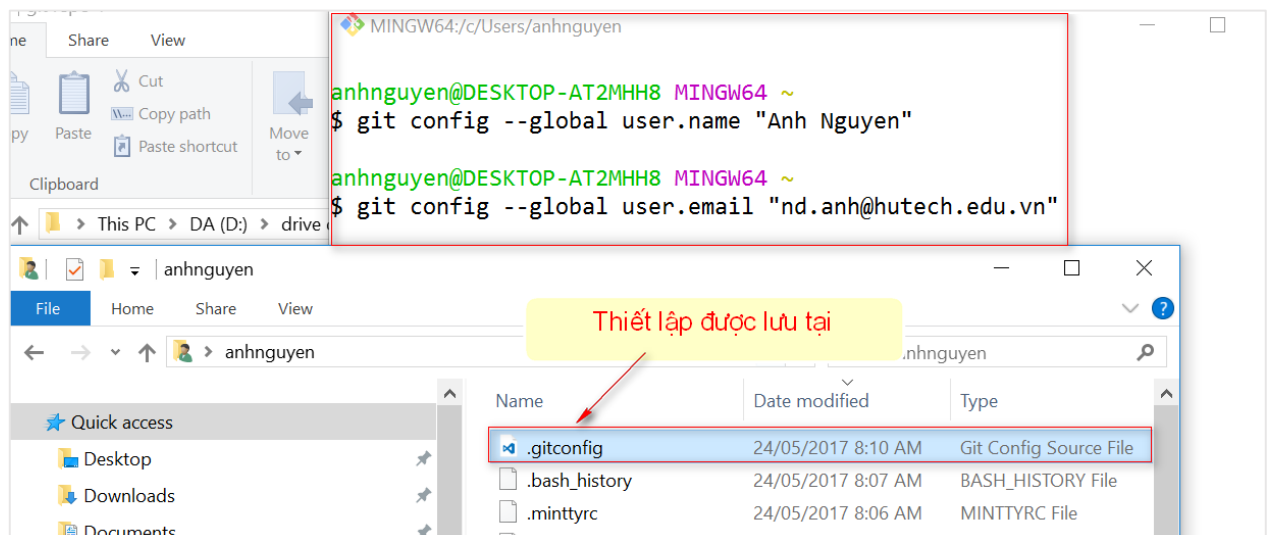
- Xóa tất cả những thay đổi tại local repo, lấy mã nguồn mới nhất từ remote repo server:

git fetch origin

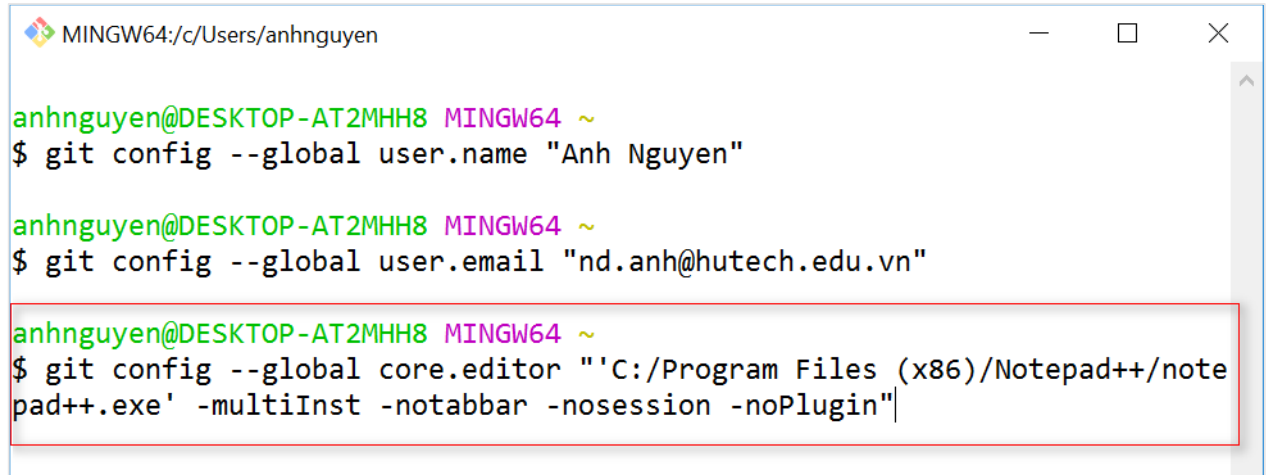
git reset --hard origin/master

Chuẩn bị

- Cài đặt git, bản cho Windows: <https://git-scm.com/download/win>
- Thiết lập username và email (mở git Bash từ cửa sổ windows)



- Thiết lập công cụ Editor mặc định là Notepad ++



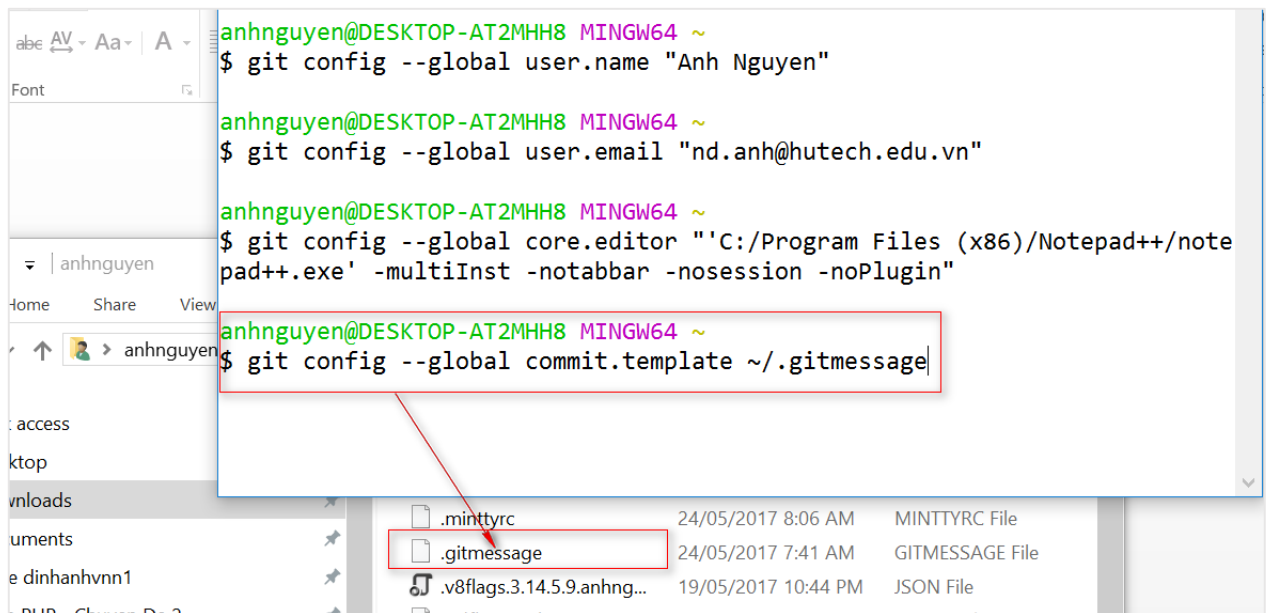
```
MINGW64: c:/Users/anhnguyen

anhnguyen@DESKTOP-AT2MHH8 MINGW64 ~
$ git config --global user.name "Anh Nguyen"

anhnguyen@DESKTOP-AT2MHH8 MINGW64 ~
$ git config --global user.email "nd.anh@hutech.edu.vn"

anhnguyen@DESKTOP-AT2MHH8 MINGW64 ~
$ git config --global core.editor "'C:/Program Files (x86)/Notepad++/note
pad++.exe' -multiInst -notabbar -nosession -noPlugin"
```

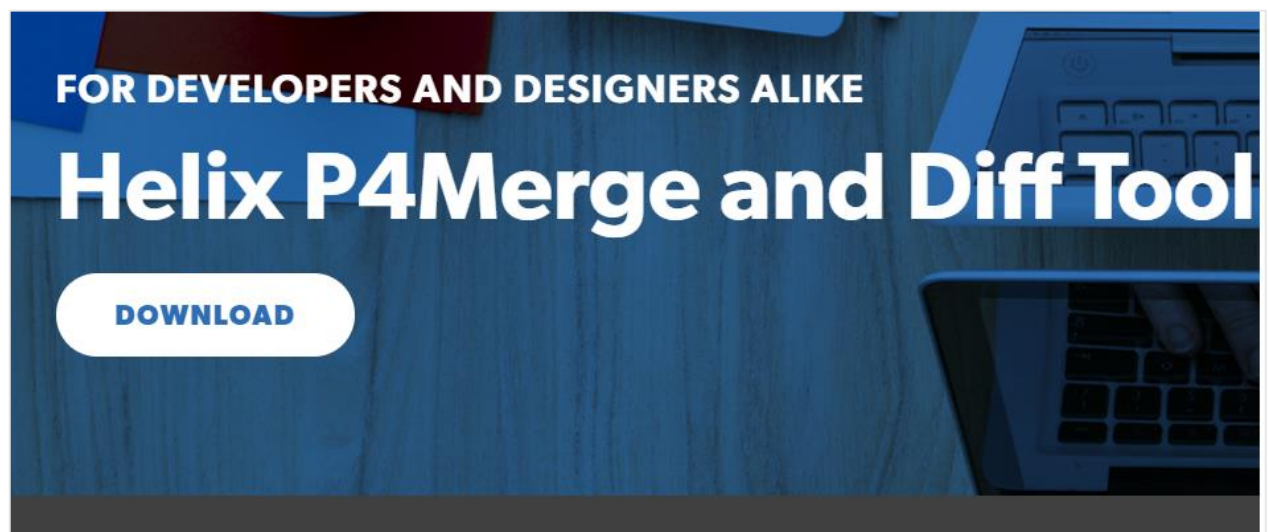
- Thiết lập mẫu template cho commit



- Thiết lập mẫu template cho commit

```
.gitmessage x
1 # <type>: (If applied, this commit will...) <subject> (Max 50 char)
2 # |<---- Using a Maximum Of 50 Characters ---->|
3
4
5 # Explain why this change is being made
6 # |<---- Try To Limit Each Line to a Maximum Of 72 Characters ---->|
7
8 # Provide links or keys to any relevant tickets, articles or other resources
9 # Example: Github issue #23
10
11 # --- COMMIT END ---
12 # Type can be
13 #   feat      (new feature)
14 #   fix       (bug fix)
15 #   refactor  (refactoring production code)
16 #   style     (formatting, missing semi colons, etc; no code change)
17 #   docs     (changes to documentation)
18 #   test     (adding or refactoring tests; no production code change)
19 #   chore    (updating grunt tasks etc; no production code change)
20 # -----
```

- Thiết lập công cụ xem lịch sử phiên bản, công cụ Merge (External Merge and Diff Tools)



- Thiết lập công cụ xem lịch sử phiên bản, công cụ Merge (External Merge and Diff Tools)

```
.gitconfig
1 [user]
2   name = Anh Nguyen
3   email = nd.anh@hutech.edu.vn
4 [diff]
5   tool = p4merge
6 [difftool "p4merge"]
7   path = C:\\Program Files\\Perforce\\p4merge.exe
8 [difftool]
9   prompt = false
10 [merge]
11   tool = p4merge
12 [mergetool "p4merge"]
13   path = C:\\Program Files\\Perforce\\p4merge.exe
14 [mergetool]
15   prompt = false
16 [core]
17   editor = 'C:/Program Files (x86)/Notepad++/notepad++.exe' -multiInst
18   -notabbar -nosession -noPlugin
```

- Tạo tài khoản github, tạo mới repo lưu trữ, thêm thành viên vào dự án

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: anhnghuyen888 / Repository name: WebsiteTMDT ✓

Great repository names are short and memorable. Need inspiration? How about didactic-octo-rotary-phone.

Description (optional): Wesbite thương mại điện tử

☒ Public Anyone can see this repository. You choose who can commit.

Thực hành: Sinh viên cài đặt phần mềm Git Tool, đăng ký tài khoản trên trang github.com để thực hiện các yêu cầu:

- Cấu hình, thiết lập cơ bản
- Lệnh tạo Repo mới
- Lệnh commit file
- Push lên remote repository
- Clone một remote repository

- Pull từ remote repository
- Lệnh phục hồi về một phiên bản trước
- Lệnh xem lịch sử các phiên bản
- Lệnh tạo nhánh, xóa nhánh
- Lệnh chuyển qua làm việc tại các nhánh khác nhau
- Lệnh merge từ các nhánh
- Lệnh merge trong trạng thái xung đột
- Mở rộng: thực hành trên GitLab.com