
Bài Lab Số 2: 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`

 **Thực hành:** Sinh viên đăng ký tài khoản trên trang github.com và thực hành xử lý các trường hợp sau:

- Tạo mới local repo, cấu hình định danh người sử dụng, kết nối tới remote repo github, đưa code lên github, thêm thành viên cùng dự án vào repo trên github, thay đổi nội dung mã nguồn dưới local repo, đưa sự thay đổi lên remote repo, thay đổi mã nguồn trong cùng một nội dung file dưới local repo và trên remote repo sau đó giải quyết vấn đề conflicts mã nguồn, khôi phục một file với phiên bản cũ hơn.