

SOFTWARE ENGINEERING (503074)

(WEEK 07)

MAI VAN MANH

VERSION CONTROL SYSTEM

Bài 1. Sinh viên cài đặt hai công cụ **VisualSVN Server** và **TortoiseSVN** trên máy tính để phục vụ cho bài học. Trong đó **VisualSVN Server** là công cụ đại diện cho phía server nơi quản lý tài khoản người dùng và các repository (kho lưu trữ mã nguồn) còn **TortoiseSVN** là công cụ phía client dùng để lưu trữ mã nguồn trong quá trình làm việc.

1. Sau khi cài đặt xong, tạo hai tài khoản **user1** và **user2** trong phần quản lý tài khoản của **VisualSVN Server**.
2. Tiếp tục tạo một repository đặt tên **cnpm-lab07**.
3. Sử dụng tài khoản **user1** để **checkout** (download) repository về desktop.
4. Làm tương tự với **user2**, lưu project vào một thư mục khác trên desktop.
5. Thực hiện các chức năng cơ bản của svn thông qua các tình huống sau:
 - **user1** tạo project java cơ bản sau đó commit để đưa lên máy chủ, **user2** update để lấy mã nguồn về rồi chạy thử.
 - Cả hai **user1** và **user2** đều tạo một tập tin mới trong mã nguồn của mình rồi cùng commit lên máy chủ (hai tập tin đặt tên khác nhau).
 - **user1** xóa một tập tin còn **user2** thì chỉnh sửa một tập tin khác rồi cả hai cùng commit lên máy chủ.
 - Cả hai user cùng chỉnh sửa một tập tin nhưng ở hai dòng khác nhau sau đó cả hai cùng commit lên máy chủ.
 - Cả hai user cùng chỉnh sửa một tập tin và cùng chỉnh sửa trên một dòng với nội dung khác nhau (ví dụ thay đổi giá trị hằng số) rồi cùng commit

lên máy chủ. Lúc này sẽ có conflict xảy ra đối với user commit chậm hơn, hãy khắc phục conflict này.

6. Thực hiện tiếp các tình huống sau:

- Tạo nhiều vài tập tin, vài thư mục mới; chỉnh sửa nội dung của vài tập tin đang có, xóa đi một vài tập tin, thư mục... giả sử lúc này project đã bị lỗi không chạy được, hãy khôi phục project trở lại hiện trạng ban đầu ở lần update/commit gần nhất.
- Cho biết hiện tại đang có bao nhiêu revision, đưa project quay lại revision đầu tiên (lúc project vừa được tạo và commit lần đầu) rồi chạy thử chương trình.
- **user1** thực hiện một tính năng mới (ví dụ viết phương thức kiểm tra xem một số có phải số chẵn hay không) sau đó tạo tập tin **patch** rồi chuyển sang cho **user2**. **user2** dùng tập tin patch này để nạp tính năng vừa được **user1** tạo để chạy và kiểm tra thử.

Bài 2. Sinh viên tạo tài khoản **github** và thực hiện các thao tác tương tự như đã làm ở bài tập 1.

1. Tạo một repository trên trang github.com, đặt tên **my-project**.
2. Clone **my-project** từ github về máy local, tạo một Java Application để chạy thử sau đó commit và push lên github.
3. Làm lại bước 2, tuy nhiên lần này yêu cầu tạo một project ở máy local trước với vài tập tin (khác với các tập tin đang có ở **my-project** trên github) sau đó push project ở local lên repository **my-project** ở github. Thao tác push lúc này sẽ thất bại vì mã nguồn ở remote và local lúc này không giống nhau, hãy khắc phục sự cố này.
4. Tạo một branch mới, bổ sung thêm mã nguồn để thử nghiệm tính năng mới.
5. Chuyển qua lại giữa hai branch và chạy thử chương trình
6. Merge branch vừa tạo về branch chính.