



Bài 2 - Git

Khóa học: NỀN TẢNG LẬP TRÌNH

Mục tiêu



- Hiểu được cơ chế quản lý mã nguồn của Git
- Sử dụng được các lệnh Git cơ bản



Thảo luận

SCMs – Source Code Management System

Quản lý mã nguồn



- Lưu trữ mã nguồn tập trung
- Chia sẻ mã nguồn giữa các bên
- Cộng tác giữa các thành viên trong nhóm phát triển
- Khôi phục mã nguồn về các phiên bản khác nhau
- Dễ dàng chỉnh sửa mã nguồn
- Tránh trùng lặp, xung đột mã nguồn

SCMs – Source Code Management System



- Source Code Management System– Hệ thống quản lý mã nguồn là một phần mềm hỗ trợ:
 - Phối hợp giữa các thành viên trong một nhóm phát triển phần mềm
 - Quản lý tập tin và kiểm soát phiên bản
 - Các nhà phát triển khả năng làm việc đồng thời trên các tập tin, hợp nhất với các thay đổi khác của nhà phát triển khác
 - Theo dõi và kiểm tra các thay đổi được yêu cầu và thực thi
 - Theo dõi tình trạng sửa lỗi và thực thi

VCS – Version Control System



- Version Control System (VCS) – Hệ thống quản lý phiên bản mã nguồn là một phần mềm hỗ trợ:
 - Khôi phục lại phiên bản cũ của các file
 - Khôi phục lại phiên bản cũ của toàn bộ dự án
 - Xem lại các thay đổi đã được thực hiện theo thời gian
 - Xem ai là người thực hiện thay đổi cuối cùng có thể gây ra sự cố
 - Khôi phục lại các file vô tình xóa mất

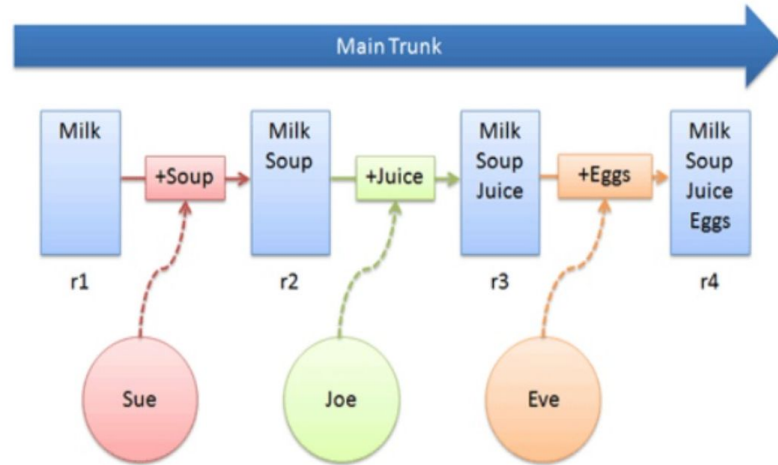
Các công cụ quản lý mã nguồn thông dụng



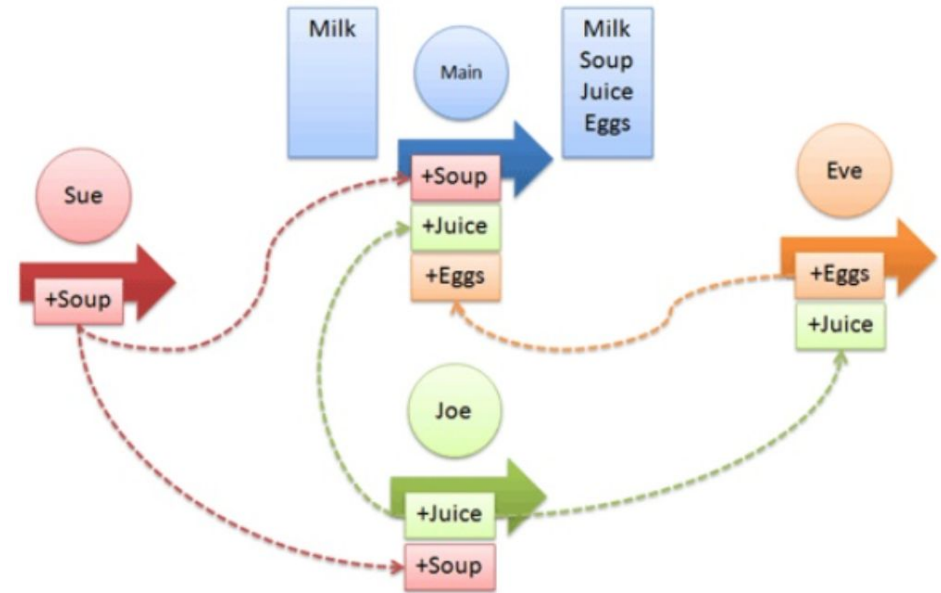
- Subversion
- Git
- Mercurial
- Bazaar
- CVS



Các kiểu của hệ thống quản lý phiên bản (VCS)



Centralized – VCS



Distributed- VCS

Git và GitHub



- Git là một hệ thống điều khiển phiên bản (version control system) theo hình thức phân tán
- Git được sử dụng để quản lý mã nguồn (source code) và ghi nhận các thay đổi
- GitHub là một dịch vụ Git được cung cấp miễn phí
- GitHub có phiên bản trả phí dành cho các doanh nghiệp





Sử dụng Git

Snapshot



- Snapshot là toàn bộ mã nguồn tại một thời điểm
- Các snapshot như là các lát cắt được tạo ra trong quá trình phát triển
- Lập trình viên quyết định lúc nào thì tạo một snapshot
- Có thể quay lại một snapshot bất kỳ

Commit



- Commit là cách để tạo các snapshot
- Các commit thường được tạo ra khi có một thay đổi đáng kể đối với mã nguồn:
 - Tạo một tính năng mới
 - Sửa được một lỗi
 - Cải tiến mã nguồn
 - ...
- Một commit bao gồm các thông tin:
 - Thay đổi ở các file so với trước
 - Một tham chiếu đến commit trước nó (gọi là commit cha)
 - Một mã băm đại diện, thường có dạng như
87878747939740429190ca307289c494311e27fe

Repository

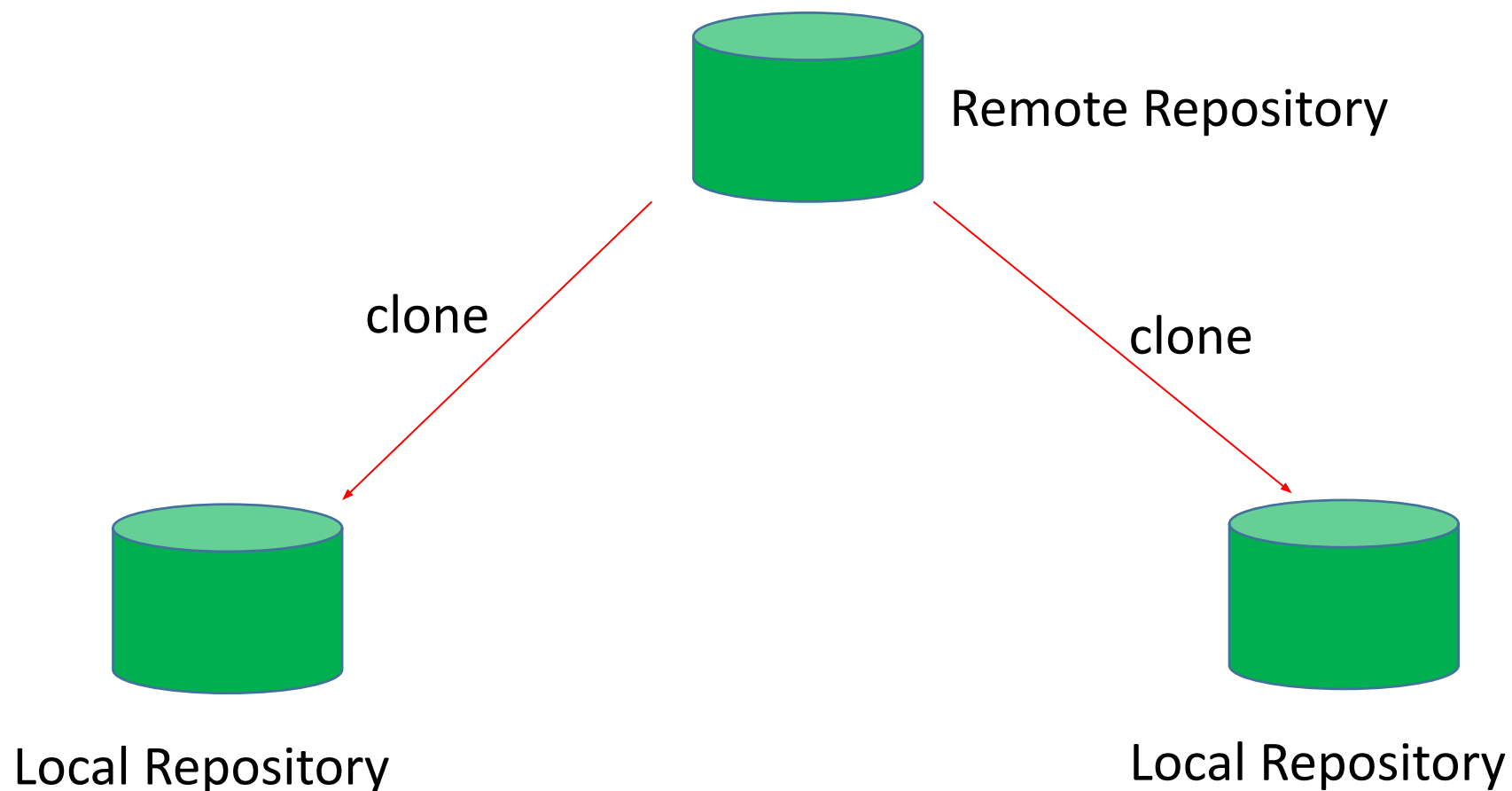


- Thường được gọi ngắn gọn là repo
- Repository là nơi chứa toàn bộ mã nguồn
- Repository bao gồm toàn bộ các file và lịch sử của các file đó
- Repository chứa tất cả các commit
- Có 2 loại repository:
 - Local Repository: Ở trên máy của lập trình viên
 - Remote Repository: Ở trên một máy chủ chia sẻ (chẳng hạn như GitHub)

Clone



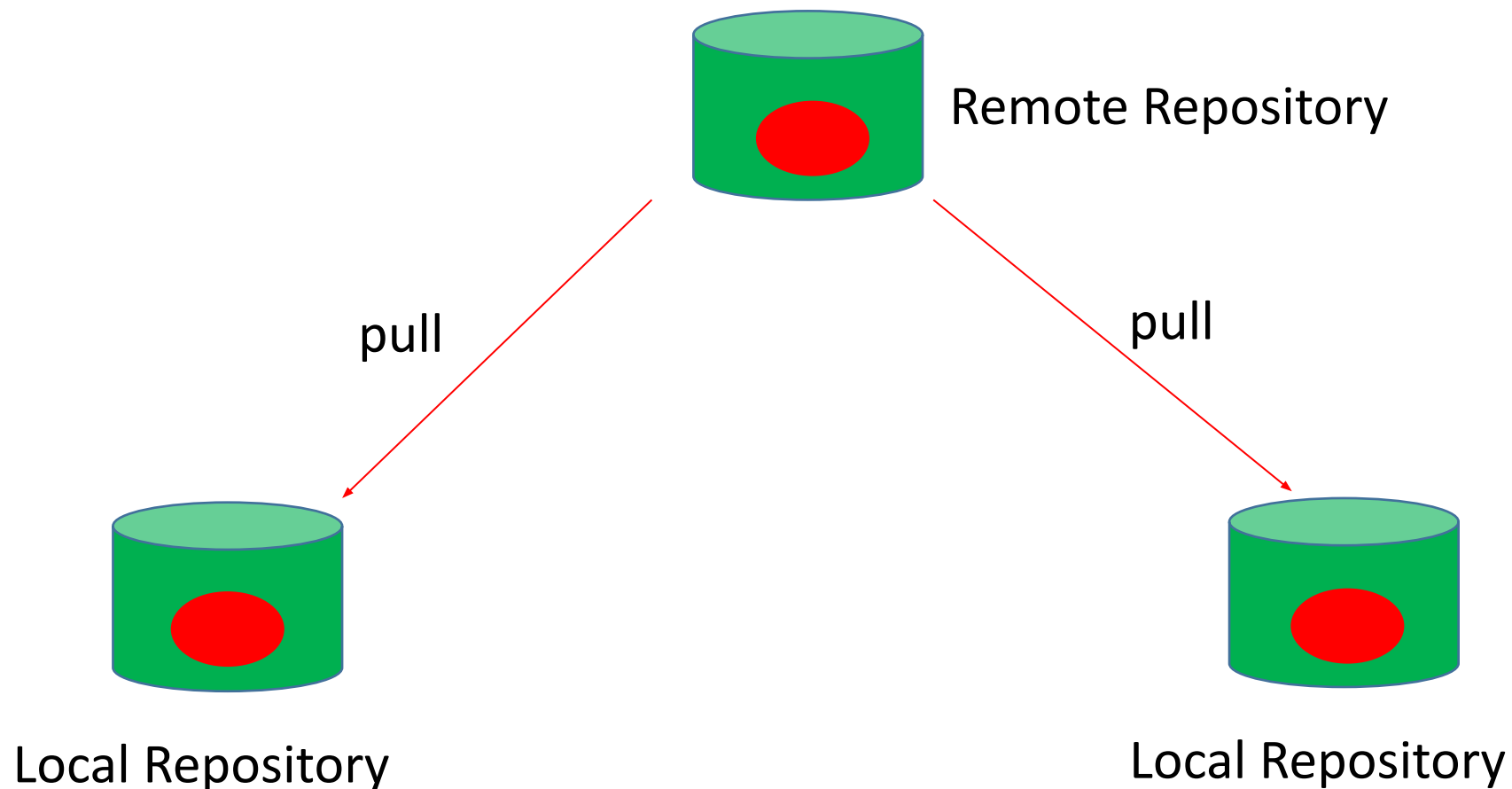
- Sao chép một Remote Repository về máy của lập trình viên



Pull



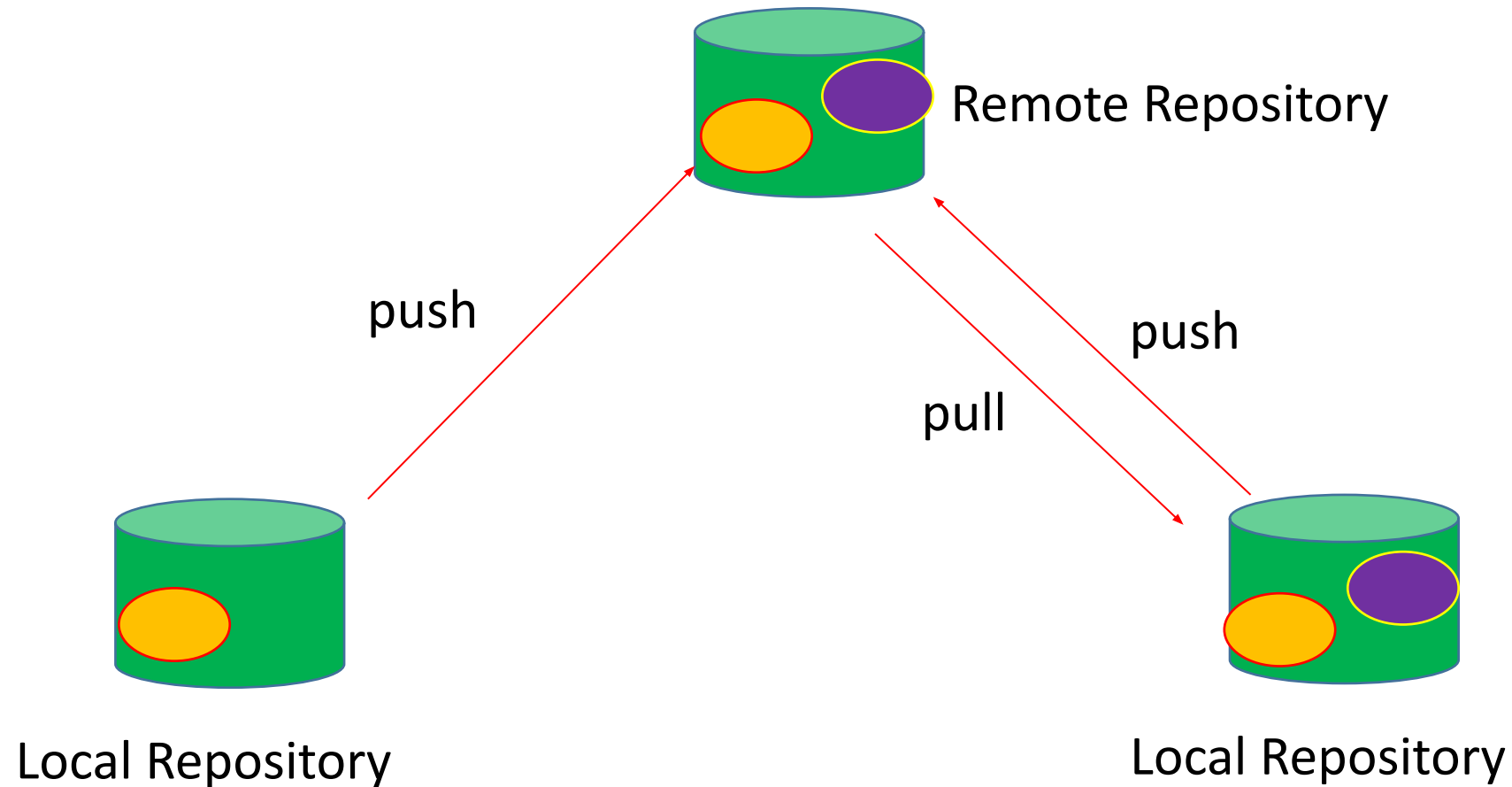
- Cập nhật mã nguồn từ một Remote Repository về Local Repository



Push



- Đẩy mã nguồn từ Local Repository lên Remote Repository





Demo

- Sử dụng lệnh Git cơ bản



Các câu lệnh cơ bản của Git

- `git clone`: Sao chép một repository
- `git init`: Khởi tạo một repository
- `git add`: Đưa các file vào trong vùng staged
- `git commit`: Ghi nhận các thay đổi
- `git push`: Đưa các thay đổi từ local repository lên remote repository



Tóm tắt bài học



- Git là cơ chế quản lý mã nguồn phổ biến nhất hiện nay, GitHub là một dịch vụ miễn phí
- Các câu lệnh thông dụng của Git bao gồm: clone, init, add, commit, push

Hướng dẫn

- Hướng dẫn làm bài thực hành và bài tập
- Chuẩn bị bài tiếp: *HTML Form & Table*