

バージョン管理システムについて

Sho Nagao

2020/12/30

ブランチを切って作業をコミット
リポジトリにプッシュしてマージする。
を理解しよう。

バージョン管理システムとは

バージョン管理システム（version control system）とは、コードなどの編集履歴を管理、保管するシステムである。



システム管理サーバー
（リポジトリ）

作業者は、基本的に以下の流れで作業を行う。

1. リポジトリをローカルに複製（clone）する。
2. ローカルで修正・追加・削除を行う。
3. 変更内容をリポジトリに反映させる（commit）。

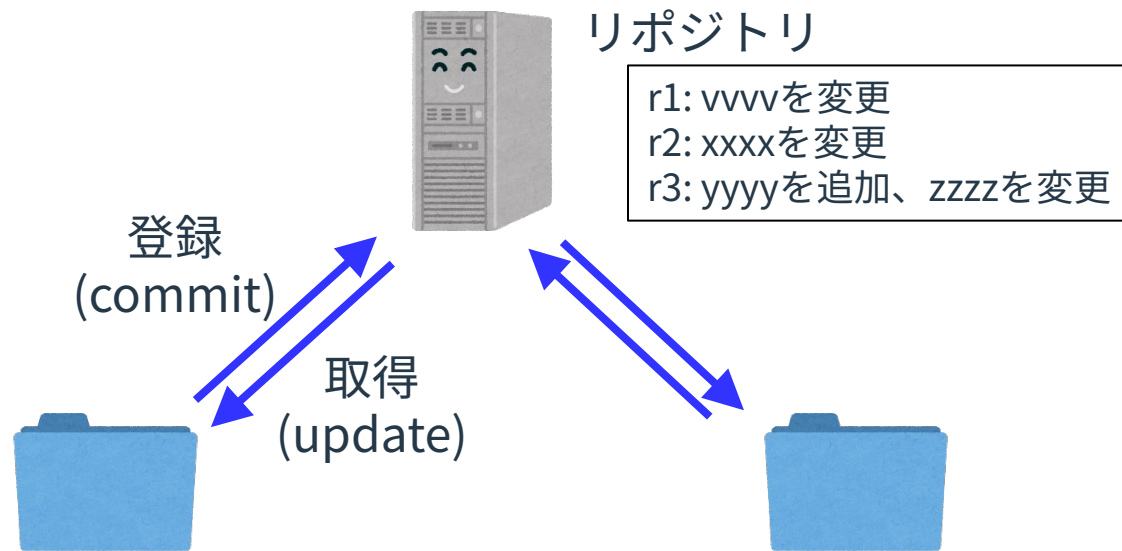
作業者
（ローカル）



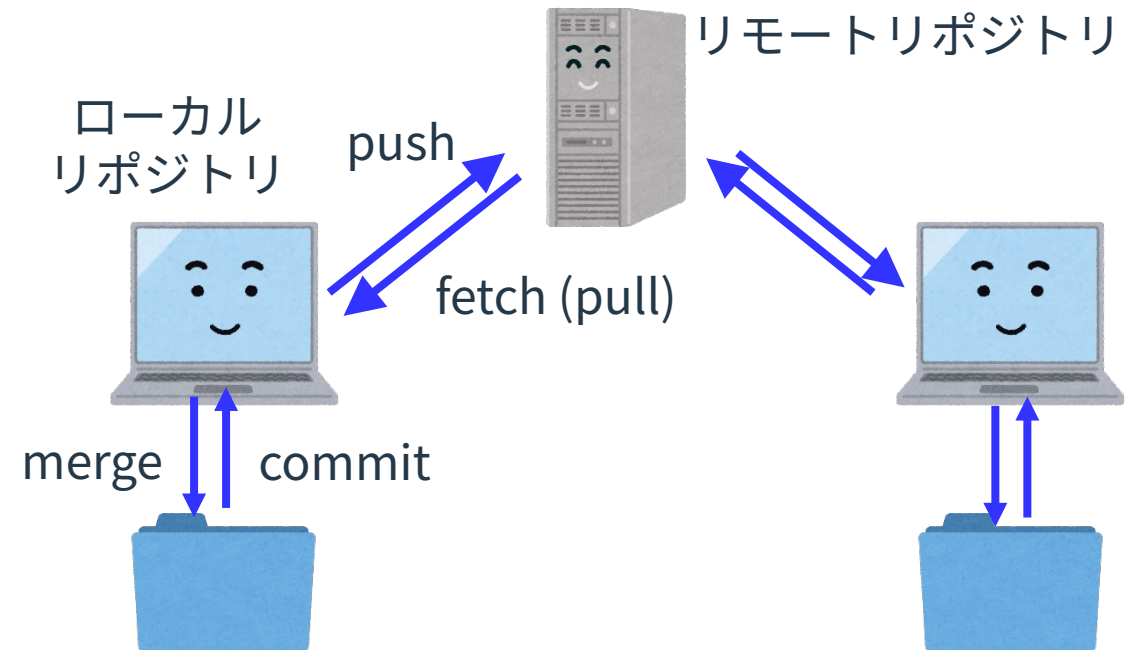
バージョン管理システムの種類

バージョン管理システムは「集中型」「分散型」に分けられる。

集中型：バージョン管理を一か所のリポジトリがリビジョン番号を割り振って行う。CVS、Subversion



分散型：各作業者がリポジトリを保有している（ローカルリポジトリ）。Git、Mercurial



集中型の方が分かりやすいが、作業者は他の作業者の内容がリポジトリに反映されるまで待つ必要がある。
近年は分散型の Git が頻繁に使われる。

作業者の流れ (Subversion)

リポジトリ名は vcs。

リポジトリに既にある test.cc を変更、新たに hoge.cc を追加して登録することを考える。

- | | |
|--|---------------|
| 1. svn checkout vcs | リポジトリの内容を複製 |
| 2. 内容を編集、追加 | |
| 3. svn update | 最新の状態にする |
| 4. svn add hoge.cc | ファイルを追加 |
| 5. svn commit test.cc hoge.cc -m “modify test.cc, add hoge.cc” | 作業内容をリポジトリに反映 |

※一度チェックアウトしてしまえば、次からは手順 1 は必要ない。

その他

変更ログを確認したい

svn log

リポジトリとの差を確認したい

svn diff test.cc

現在の状況を確認したい

svn status

変更内容をリポジトリの内容に戻したい

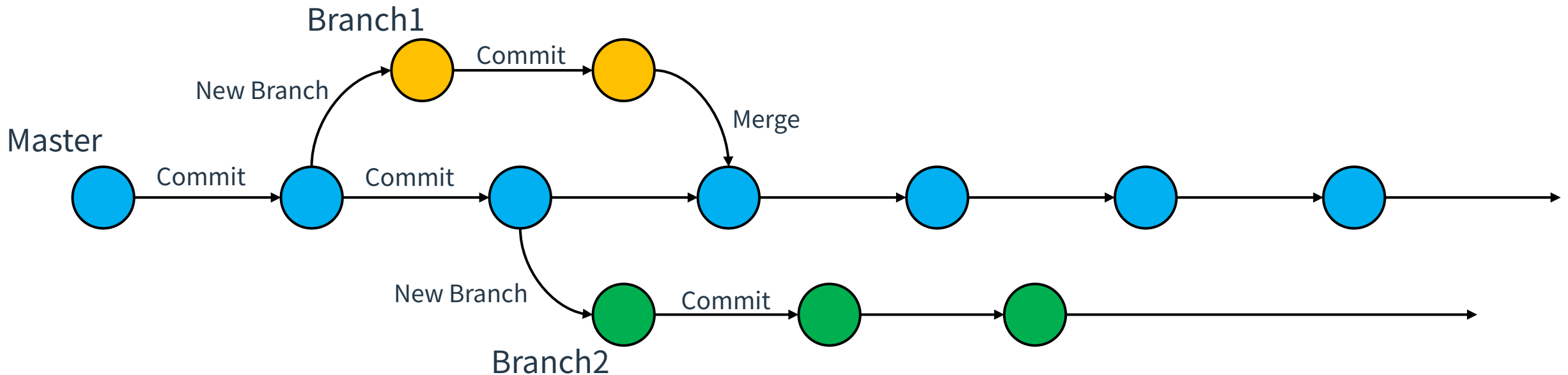
svn revert test.cc

ブランチという考え方 (Git)

Git はブランチという機能が充実している。他者の変更内容と競合を避けるため、作業者は、

1. Master の内容を Branch を切って編集作業を進め、
2. ローカルリポジトリに変更内容を逐次コミット。
3. ある程度形になったところで Branch のリモートリポジトリにプッシュし、
4. 最終的に Master に Branch の内容を Merge する。

といったように進めることが多い。Subversionでも似たような機能があるが充実していない。開発方針によっては、Master のみで進めることもある。



作業者の流れ (Git)

リポジトリ名は vcs。master ブランチから local ブランチを作成。リポジトリに既にある test.cc を変更、新たに hoge.cc を追加。最終的に master ブランチにマージすることを考える。

- | | |
|--|----------------------|
| 1. git clone vcs | リモートリポジトリの内容を複製 |
| 2. git pull | 最新の状態にする |
| 3. git branch local | local ブランチを作成 |
| 4. git checkout local | local ブランチに移動 |
| 5. 内容を編集、追加 | |
| 6. git add hoge.cc | ファイルを追加 |
| 7. git commit test.cc hoge.cc -m “modify test.cc, add hoge.cc” | 作業内容をローカルリポジトリに反映 |
| 8. git push origin local | ローカルをリモートに反映 |
| 9. git checkout master | master ブランチに移動 |
| 10. git pull origin master | master ブランチを最新の状態にする |
| 11. git merge local | local を master にマージ |
| 12. git push origin master | 内容をリモートに反映 |

※一度クローンしてしまえば、次からは手順 1 は必要ない。

※ブランチ機能を使わないのであれば灰色は不要。



GitHubとは、Gitを利用した、開発者を支援するWebサービスです。

自分でサーバーを立ててリポジトリを作成する必要なし。

リポジトリを public にすれば、他人とコードの共有も可能です。

執筆中の論文の管理にも使えます。

開発のレビューなども簡単にできるので非常に便利。是非アカウントを作っておきましょう。

Subversion と Git のコマンド

内容	Subversion	Git
リポジトリを作成	svn create	git init
リモートリポジトリの内容を複製	svn checkout	git clone
作業ディレクトリを更新	svn update	git pull
ローカルリポジトリのみ更新		git fetch
ローカルリポジトリの内容を作業ディレクトリに反映		git merge
ファイルの追加（移動、削除）	svn add (mv, rm)	git add (mv, rm)
変更をリポジトリに反映	svn commit	git commit
変更をリモートリポジトリに反映		git push
変更の取消	svn revert	git reset
状態確認	svn status	git status
差分確認	svn diff	git diff
ログの確認	svn log	git log
ブランチの作成、一覧		git branch
ブランチの切替		git checkout
ブランチのマージ	svn merge	git merge