Git によるバージョン管理入門

田中 健策（株式会社 RAKUDO）

第一回

バージョン管理とは何か？

- 「何の」修正を行ったのか調べることができる。
- 「だれが」その修正を行ったのか調べることができる。
- 「いつ」その修正が行われたのか調べることができる。
- 「なぜ」その修正が行われたのか調べることができる。
- ファイルを過去の状態に戻すことができる。

ファイル名によるバージョン管理をやめよう

ファイル名-20191004-田中-ver2.xlsx
のように、ファイル名に色々くっつけて、バージョン管理する職場がまだまだ多いようですが、機械のほうが上手なことをわざわざ人の手でやることはやめましょう。

バージョン管理システム小史（その１）

- 1972 年、ベル研究所の Marc J. Rochkind が、世界初のバージョン管理システム SCSS を開発。
- 1982 年、Walter F. Tichy が RCS を開発。

これはファイルを１つずつローカルで管理するためのツール。
ファイルを編集するためには**ロック**を取得する。ロックを解放するまで、そのファイルはその人しか編集できない。
新規に使われることはないが、保守業務で RCS が使われるという話は聞いたことがある。

バージョン管理システム小史（その２）

- 1990 年、CVS が RCS の上に開発される（後に RCS から独立する）。
- 2000 年、CVS の弱点を解消するために Subversion が開発される。

複数のファイルの同時管理や、枝分かれしたヴァージョンの管理ができるようになった。
また、ネットワーク上の使用を考慮するようになった。情報をリモートで一括管理するサーバー・クライアント方式により、より多人数での開発が便利になった。
ネットワーク上でロックの仕組みは使いづらいので、同時編集を許可して、**マージ**する、というワークフローが始まった。

バージョン管理システム小史（その３）

- 2005 年、Linus Torvalds によって Git が開発される。
- 同年、Matt Mckall によって Mercurial が開発される。
- Bazar,Darcs など、他にも色々あった。でも正直今は Git 一強。

分散バージョン管理を実現している。ネットワークが繋がってない状態でも、ローカルでバージョンを管理して、それをリモートに反映させることができる。
リモートが複数あってもいい。

ホスティングサービス

- 1999 年、SourceForge.net スタート
- 2008 年、GitHub がスタート
- 同年、Bitbucket もスタート
- 2011 年、GitLab がスタート

Git などのリモートリポジトリの管理をしてくれるサービス。
今では、リポジトリを結節点として、様々な便利な機能を提供するようになり、Git の使い方だけではなく、GitHub などの使い方を覚えることが、重要になり始めた。

Gitを使ってみよう

- git に関する入門記事は web にたくさんあります。自分で探して読んでみましょう。
- お好きな git クライアントをインストールして動かしてみよう。GUI なら Sourcetree などがあります。また VSCode などのエディターにも git クライアントが付属しています。
- CLI でもいいなら git for Windows（git bash）などもあります。mac や linux なら CLI の git は最初から入っています。
- 入門記事に従って、適当なファイルを git に add して、commit してみましょう。それを何回か繰り返して log をみて見ましょう。

これは全てローカルで行われているので、複数人でコラボレーションをするという git の醍醐味はまだ始まっていませんが、全てを記録に録るという git の大きな特徴は既に現れています。

今週の課題

- GitHub にアカウントを登録してください。
- 作成したアカウントの email アドレスから `kensaku_tanaka@rakudo.io` に「アカウント登録お願いします」という表題でメールを送信してください。名前と学籍番号、github でのユーザーネームを必ず記入すること。
- github のリポジトリのコラボレーターとして招待します。その招待メールに反応して、コラボレーターとして登録してください。

コラボレーターとして登録され、それが学籍番号と紐づけられることによって合格です。

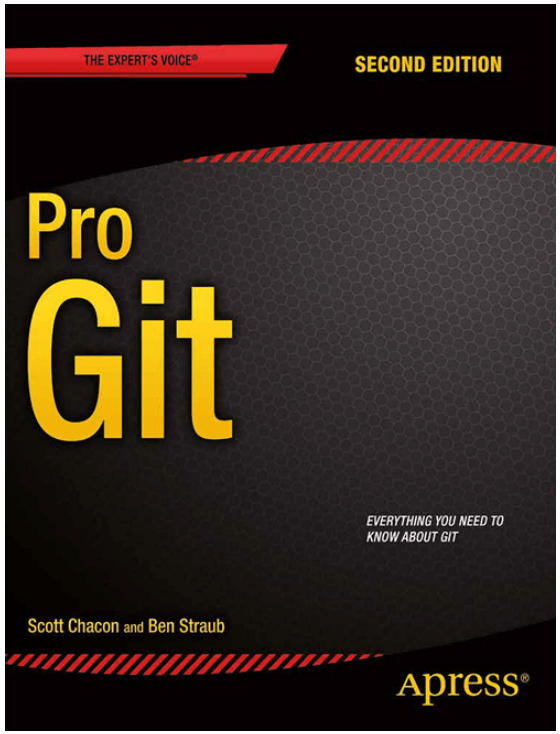
講義の最終目標

大学の数学の入試問題（例えば昨年の名大の入試問題）の解答を LaTeX で作成してもらいます。
例えば http://www.nagoya-u.ac.jp/admission/upload_images/06%20R2%20sugakurikakei-mondai.pdf などです。
リポジトリは
https://github.com/tannakaken/nugitlecture2020 です。
最終的に、解答の pdf が一つできるようになることが目標です。
実際に開発をしながら、ブランチ分けや CI/CD などのテクニックを学習します。
自分の環境に LaTeX の実行環境をインストールしておいてください。

最終的な成績評価

成績評価は、GitHub の log から貢献度を計測して行います（目で見るだけです）。
なので NUCT を使った提出物はありません。
log が残っていれば（github 上で活動の記録があれば）、5 点与えます。
最終成果物に意味のある貢献があれば、10 点与えます。
活発に活動している記録があれば、15 点与えます。

参考文献
Pro Git



https://git-scm.com/book/ja/v2 で無料公開中。
技術的細部も含めた本格的な内容です。