

Git によるバージョン管理入門

田中 健策（株式会社ぺあのしすてむ）

第三回

前回までと今後の方針

前回まで

- tortoiseGit の設定の Network > Proxy に、名大の Proxy サーバーの URL を設定してください。
- もしまだコラボレーターに登録できていない人がいたら教えてください。

今後の方針

- 分割した LaTeX ファイルをまとめて一つの pdf にする方法を考える
- コンパイルなどの作業をできるだけ自動化する。

LaTeX のソース分割 1

複数人で LaTeX を書くときは、ファイル分割をする必要がある。一番簡単なファイル分割は、`\input` や `\include` を使うものである。

しかしこれには、単独のファイルでコンパイルできないという欠点がある。

LaTeX のコンパイルはそれなりの時間がかかるので、これは避けたい（コンパイルして結果を見るためだけに数秒かかるのは、執筆時に多大なストレスになる）。

LaTeX のソース分割 2

そこで、自分でマクロを各方法や、用意されたパッケージを使う方法など、様々な方法が開発されている。

- 効率的な LaTeX ファイル分割術

<https://qiita.com/wtsnjp/items/6ba3b8e12514d8a3bd41>

- 分割した LaTeX ファイルを subfiles を使ってコンパイルする

[https:](https://qiita.com/sankichi92/items/1e113fcf6cc045eb64f7)

[//qiita.com/sankichi92/items/1e113fcf6cc045eb64f7](https://qiita.com/sankichi92/items/1e113fcf6cc045eb64f7)

今回は、subfiles パッケージを使ってみようと思う。

git hooks

git には操作をした時に、自動的に別のプログラムを走らせるフックという仕組みがある。

これを使えば、様々な定型作業（決まり切った作業）を自動化することができる。

ただし使うためにはシェルスクリプトなどの、コマンドライン・プログラミングの知識が必要になる。

ローカルリポジトリ用のフック

- pre-commit commit する前に走る commit を中止できる
- post-commit commit した後に走る
- pre-push push する前に走る push を中止に知ることができる
- poist-push push した後に走る
- checkout 前後や merge 前後や rebase 前後に走るフックもある

これを使うと、

- commit 前にコンパイルを走らせ、失敗したら commit 中止
- push 前に検査プログラムを走らせ、例えばルールに反していたら push 中止

などの自動化ができる。

リモートリポジトリ用のフック

- pre-receive push を受け取った直後に走るフック push をリモートの側から中止にできる
- update push を受け取った時に、ブランチごとに走るフック
- post-receive push 後、全ての処理が終わった後に走るフック

これを使うと

- テストに失敗すれば push を中止
- master ブランチに push したときだけ、ファイルをコンパイルして、その pdf ファイルを自動的に web ページに設定する
- 全ての処理終了後、変更部分等をメール、slack、Twitter で通知。また CI ツールなどに情報を渡すこともできる

などの自動化ができる。

Github の自動化ツール

git のフックでできることは限られているが、もっと高度な自動化機能が求められることもある。

ソフトウェア業界では様々な定型作業を自動化していく習慣を **CI**（継続的インテグレーション）や **CD**（継続的デプロイ）と呼ぶ。Github は今まで CI/CD ツールについては Gitlab に遅れをとり、外部の CI/CD ツールとの連携が必要であったが、マイクロソフトによる Github 買収後、マイクロソフトのマネーパワー流入によって、必要なツールは急ビッチで整えられている。

特に CI/CD ツールについて Github は **Github actions** というツールを昨年リリースし、今年に入ってそれが CI/CD のツールとしての機能を標準サポートするようになった。

今回はこれを使って、 \LaTeX の自動コンパイルと pdf の自動的な追加を実現してみよう。