

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）

第二回

前回の訂正・確認等

色々グダグダですみません。

- メールアドレスで `github` のコラボレーターに登録できるかと思っていましたが、メールアドレスが公開されていないとメールアドレスでは登録できないことが判明しました。すみませんが、ユーザー名でメールを送り直してください。
- この部屋はどうやらものすごくネットワークが悪いです。なので、`git clone` をする時に、全員一斉に行わずに、時間差でやる必要があるかもしれません。ご協力をお願いします。
- ゴミ箱にゴミが溜まっていて問題を起こすこともあると、指摘を受けました。ゴミ箱のゴミは消さないで溜まっています。確認して決しておいてください。

コンフリクトとは何か？

リポジトリに違う場所から矛盾する変更を加えると「コンフリクト（衝突）」が起こる。

github 上とローカルで矛盾する変更を加えて、git pull をすると、ファイルが次のようになる。

```
<<<<<<< HEAD
goodbye world
=====
hello world
>>>>>>> 943145c978f2c15e0ee82ec7baae9671dfdec54e
```

HEAD は現在のレポジトリの先頭

下は HEAD と衝突しているコミットの番号

コンフリクトを解消する

これを github に push することはできない。
どちらかいない方を消すか、またはより正しい記述に直すかして、コンフリクトを解消しないと push できない。
もし大人数で一つのリポジトリを同時に編集していくと、誰かが編集するたびにコンフリクトが発生する可能性がある。
これを俗に「コンフリクト地獄」と呼ぶ

ブランチを分ける

なので、編集する目的に分けてリポジトリのコピーを作り、別々に編集する。それを「ブランチ（枝）」と言う。
大規模なチームでは、細かくブランチを分ける（ブランチを切るという謎の方言を使う人も多い）。
このブランチの分け方の手順をフローといい、いくつか開発されている。

分けたブランチをマージする

ブランチを分けることで、一つ一つのブランチではそれほど衝突の危険を冒さずに編集できる。

そしてブランチの編集が終わったらそれを、main などの名前のついた中央ブランチにマージ（融合）させる。

その時にコンフリクトが発生すれば、解消する。

これによって、コンフリクト地獄に陥らずに、編集ができる。

また編集途中でも中央ブランチは汚れていないので、他の人がそこから別の編集を始めるときにもやりやすい。

コミットの粒度

一つのコミットはあまり大きくないことが推奨される。
何か困ったことがあったあった時に、どのコミットが原因かがわかりやすくしたい。
また diff で差分を調べたときにも、見るべき場所が少なくてわかりやすい方が良い。
一つのコミットは一つの変更、コミットのコメントが一言で言えるようにすると良い。
だが、今回はプログラミングではなく文書なので、正しいやり方は世間でもまだ見つかっていない。

その他 git は奥深い

git には過去を改変したり、過去のグラフを繋げ変えたり、消えてしまった過去を修復したり、変更を一時的に記録したり、と様々な奥深い機能がある。

今回はそこまで行くつもりはないが（もし必要になったら、この授業がかなり失敗してる証拠かもしれない）、もしかしたら将来必要になるかもしれない。

git は変更を木状に分岐し、それを融合させてグラフ状になるという構造をしている。

それ自体数学的になかなか面白い構造をしているので、興味があったら調べてみるといいかもしれない。