

## git

山下 俊樹, 外村 篤紀

Toshiki YAMASHITA, Atsuki TONOMURA

## 1 はじめに

近年, コンピュータプログラムは大規模化し, その更新頻度は増加の一途を辿っている. そこで, コンピュータ上で作成, 及び編集されるファイルの変更履歴を管理するためのバージョン管理システムが注目を集めている. 現在では, かつて主流であった SVN に代わり, より高機能な git への移行が進んでいる. 本報告では, git の概要と関連サービスについて述べる.

## 2 git とは

git (ギット) とは, 分散型バージョン管理システムの 1 つであり, 従来用いられてきた集中型バージョン管理システムに変わり登場した. その概要を以下の Fig. 1 に示す.

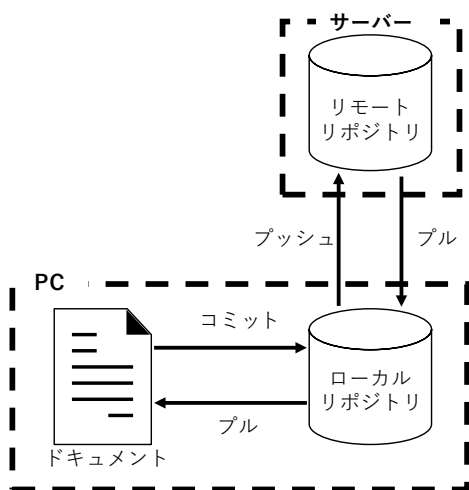


Fig.1 git の概要

git で管理されるファイルやディレクトリは, リポジトリと呼ばれる一種のデータベースにその変更が蓄積される. 分散型バージョン管理システムにおけるリポジトリには, サーバー上に配置され, 複数ユーザーで利用するリモートリポジトリと, 個人の PC 内に配置され, その個人が利用するローカルリポジトリの二種類がある. ユーザーがファイル等の変更をローカルリポジトリに記録する作業をコミッ

トと呼び, これを実行すると前回のコミットからの差分が記録される. 以前のコミットの時点の状態に戻す操作及び後述するブランチを切り替える操作をチェックアウトと呼ぶ. また, ローカルリポジトリの変更をリモートリポジトリにアップロードする作業をプッシュ, リモートリポジトリからローカルリポジトリに変更をダウンロードする作業をプルと呼ぶ.

次に, git の機能の一つであるブランチについて述べる. その一例を以下の Fig. 2 に示す.

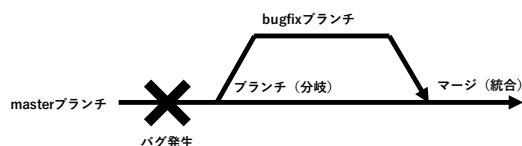


Fig.2 ブランチの一例

ブランチとは, 変更履歴の流れを分岐する作業と分岐後の流れを指す. また, ブランチ同士の結合をマージと呼ぶ. ブランチ同士は独立しており, 他のブランチの影響を受けない. また, Fig. 2 のように, 既に運用されているソフトウェアの開発を行っているブランチを master ブランチとし, そのソフトウェアのバグを修正するために用意したブランチを bugfix ブランチとする. この場合二つのブランチは独立しているため, ソフトウェアの運用を止めることなくバグの修正を行うことができる.

また, 他のバージョン管理ソフトウェアに対する git の利点は次の通りである.

- ローカルリポジトリを持つ
- ブランチ機能が強力
- 高速である

ローカルリポジトリによって, ネットワークに接続されていない環境でもコミットを行うことができる. あるいは, 個人のローカルリポジトリにバグの修正のためにコミットを適量蓄積してから, リモートリポジトリにプッシュし他者に公開する等の使用方法が可能である. ブランチについては, マージを行う際に自動化されている比率が高いこと, マー

ジが高速である等が挙げられる。更に、プログラム全体の動作が他サービスに比べて高速である。これは、git が Linux カーネル開発に使用するために開発されたため、OS カーネルのような巨大なソースコードの集合を高速に処理する必要があった経緯がある。

一方、欠点として、git で扱えるファイルの種類はテキストベースのファイルに限られることが挙げられる。このため、書類作成のデファクト・スタンダードである、Microsoft Office で作成したファイルを改変なしでは管理できない等の不具合を生じる。

### 3 GitHub とは

GitHub とは、git を利用した SNS である。ユーザーはリモートリポジトリを無料で持つことができ、他のユーザーと協力してソースコードを管理できるサービスとして広まった。他のユーザと協力するための機能として、コードレビュー、及びコメント機能等がある。コードレビューとは、ソースコードに含まれる誤りを検出、修正することを目的として行われるソースコードの査読を指す。これらの機能を用いることで、ユーザーのソースコードが他者の目に触れ、それに対してのコメントが付与される。その結果、ユーザーによる修正が行われ、ソースコードのバグが減少する、及び可読性が向上する等の利点を生む。

次に、フォーク機能について述べる。GitHub におけるフォークはブランチの分岐操作を指す。この機能によって、ユーザーが他のユーザのリポジトリからフォークしたブランチを用いて開発を行うことができる。利点としては、あるソフトウェアのデバッグを開発者以外のユーザーが行う事ができる、初心者がある程度完成されたソフトウェアをフォークすれば、フォーク元の本体には影響を与えず、様々な改変を行いながら学習ができる等が挙げられる。

更に、チケット機能アジャイル

これらが GitHub の利点である。一方、GitHub で作成したリモートリポジトリは全て公開されるため、ソースコードを公開しない場合の多い Web デザイン等のソースコード管理には不具合が発生する。また、GitHub は外部サービスであるため、サーバー障害の様な事態が発生した場合に関連サービスが使用不可能になる危険性がある。実際に、2016 年 1 月 28 日にサービス障害が発生し、多くのユーザーや企業が障害を被った。<sup>1)</sup>

GitHub に競合するサービスとして、BitBucket が挙げられる。Github に対して、非公開のリモートリポジトリを作成できる、及び git 以外の分散型バージョン管理システムである Mercurial を使用できる等の利点がある。ソーシャルコーディングの立役者 Github は最も人気のある git ホスティングサイトである。<sup>2)</sup>

### 4 今後の展望

この先、更なるソフトウェアの開発速度の向上が予想される。

### 参考文献

- 1) 佐藤由紀子, GitHub、1 月末のダウンの原因はデータセンターでの停電と説明, (<http://www.itmedia.co.jp/news/articles/1602/01/news070.html>).
- 2) Ken Nishimura, 970 万人の開発者を擁する GitHub が「ギットハブ・ジャパン」を始動、法人向けでマクニカが販売, (<http://jp.techcrunch.com/2015/06/04/github-launches-the-first-branch-in-japan/>).