

## 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.

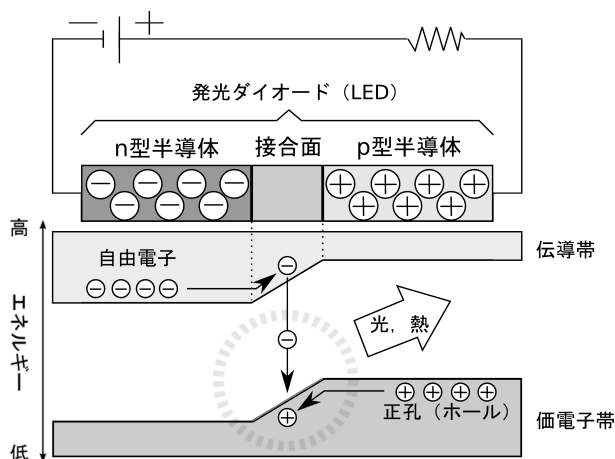


Fig.1 git の概要

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. 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.

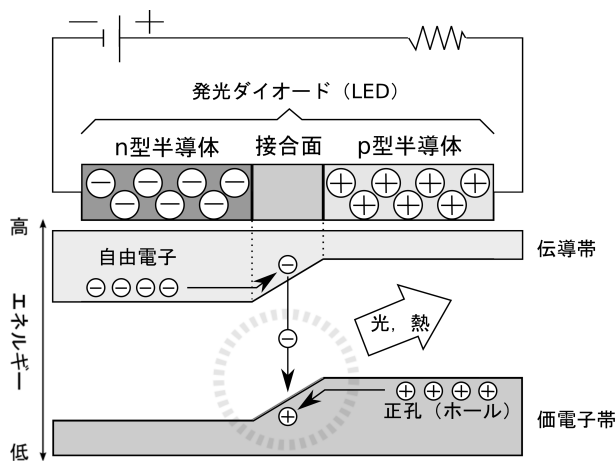


Fig.2 ブランチの一例

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

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。また、例えば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法が可能である。

一方、欠点として、git で扱えるの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。

### **3 GitHub とは**

GitHub とは、git を利用した SNS であり、プロジェクトホスティングサービスである。

### **4 今後の展望**

#### **参考文献**