GitのcloneとSubversionのチェックアウト

ローカルやテストサーバーなどにファイルをダウンロードして、更新作業を行うには、Subversionでは、チェックアウトという操作を行い作業ディレクトリを作成する。

そして、Gitではこれとよく似た操作でcloneという操作を行う。

では、このcloneとsubversionのチェックアウトは何が違うのか？

両者の最も大きな違いは、cloneの結果得られたローカルの作業ディレクトリ内には、レポジトリの全ての修正履歴が含まれている点にある。

したがって、サーバーへのアクセスができないときでも、履歴が参照できたりする！これはエライ！

実際にcloneを行って作業ディレクトリを作成する

ターミナルでデスクトップ上等に任意のディレクトリを作成する。

mkdir ~/Desktop/WorkDir

cd ~/Desctop/WorkDir

この後、cloneを行うためのコマンドを入力するのだが、そのためのリポジトリのパスをあらかじめ調べておく。

リポジトリのパスは、前回の記事で作成したforkの画面上に記載されている。

forkをしたリポジトリは、”Read+Write Access”となっていることに注目。

では、実際にこのパスをコマンドにコピペして使用する。

git clone git@github.com:user/path/to/repos.git

以上を実行すると、カレントディレクトリ内に作業ディレクトリが作成される。

commitしてpush

作業ディレクトリ内に作成された、ファイルを修正したらリポジトリに修正内容を反映する。

それには、以下の操作を行う。

１）追加したファイルや修正したファイルをGitの索引に追加する。

git add file.txt

file.txtというのが修正したファイル。

これに該当するコマンドは、Subversionにはない。

２）コミットする

git commit -m “hogeをhugaした”

これは、コマンドのフォーマットは、Subversionとにているが、この時点ではサーバーのリポジ鳥に反映されていないことが、大きく違う。

なので、以下のコマンドを実行して、”なかった事にする”ことができる。これもエライ！

git reset --soft HEAD^

３）サーバーに反映する

コミットした結果をリポジトリに反映するには、以下のコマンドを実行する。

git push

以上で修正した結果が、forkに反映される。

もし、forkに反映されたものをマスターに反映してもらいたいなら、メール等でマスターのオーナーにお願いすると、反映してもらえるかもしれない。

というわけで、gitについてはここまで。

ちなみに、Gitコマンドについては以下のサイトがわかりやすくまとまっていた。