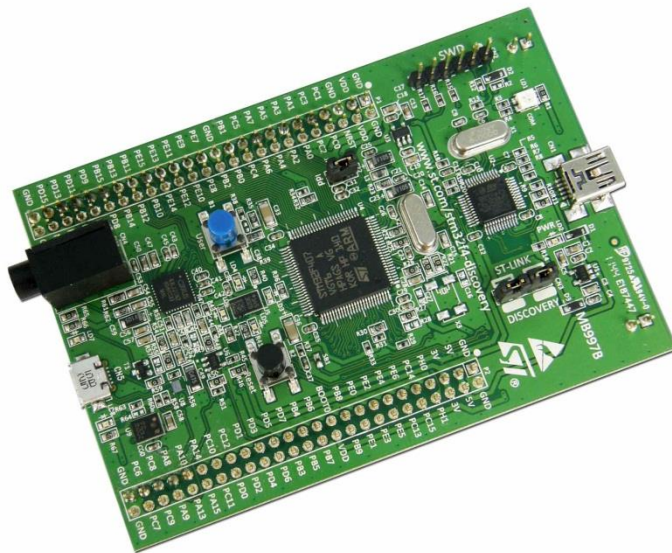


Politechnika Poznańska, Instytut Automatyki i Inżynierii Informatycznej

Podstawy technik mikroprocesorowych

Projekt – system kontroli wersji



Michał Fularz
2013-03-04

1. Wstęp

System kontroli wersji (*ang. version/revision control system*) jest to oprogramowanie służące do śledzenia zmian głównie w kodzie źródłowym oraz pomocy programistom w łączeniu zmian dokonanych przez wiele osób w różnych momentach (http://pl.wikipedia.org/wiki/System_kontroli_wersji, 2013).

Jednym z popularnych rozproszonych systemów kontroli wersji jest **Git** stworzony przez Linusa Torvaldsa na potrzeby rozwoju jądra Linuxa ([http://pl.wikipedia.org/wiki/Git_\(oprogramowanie\)](http://pl.wikipedia.org/wiki/Git_(oprogramowanie)), 2013).

2. Dodatki

Strona projektu git:

<http://git-scm.com/>

Darmowa książka:

<http://git-scm.com/book>

3. Przebieg zajęć

Pobrać najnowszą stabilną wersję git dla używanego systemu operacyjnego ze strony:

<http://git-scm.com/downloads>

Utworzyć konto w serwisie:

<https://github.com/>

Wykonać zadania (podstawowe komendy git):

<http://try.github.com/levels/1/challenges/1>

W celu rozpoczęcia pracy należy podać swoje dane, najlepiej, aby pokrywały się z danymi podanymi przy tworzeniu konta w serwisie github (w konsoli Git Bash):

```
git config --global user.name "Your Name Here"
```

```
git config --global user.email "your_email@example.com"
```

Następnie wykonać:

- utworzyć nowe repozytorium (za pomocą webowego interfejsu github),
- korzystając z programu Git Gui skopiować (clone) do lokalnego folderu utworzone w poprzednim punkcie repozytorium,
- dodać plik tekstowy (first.txt, zawartość: „Hello git!”),
- w Git Gui zatwierdzić i opisać zmiany (przyciski Stage Changed – oznacz zmienione, Commit – dodaj zmiany – ten krok wymaga dodania opisy, Push – wyślij na serwer. Jeśli dodany plik nie jest widoczny nacisnąć przycisk Rescan,

- dodać osobę ze stanowiska obok jako współpracownika (strona github -> Settings -> Collaborators),
- skopiować na swój komputer repozytorium do którego zostaliśmy dodani (analogicznie do poprzednich kroków),
- dodać do repozytorium kolegi / koleżanki kolejny plik tekstowy (second.txt, zawartość: „United we stand, divided we fall!), wysłać zmiany na serwer (Stage Changed -> Commit -> Push),
- pobrać zmiany dokonane przez kolegę/koleżankę w oryginalnie utworzonym repozytorium (menu Remote -> Fetch from – pobierz listę zmian, a następnie menu Merge -> Local Merge – połącz pobrane zmiany z lokalną wersją kodu.
Uwaga: w Git Gui nie ma z pewnych względów (<http://longair.net/blog/2009/04/16/git-fetch-and-merge/>, 2013) poznanego w trakcie tutoriala polecenia pull. W związku z tym korzystamy z Fetch -> Merge.
- na stronie github obejrzeć zakładkę Network oraz Statistics dla utworzonego repozytorium.

4. Zadania do samodzielnej realizacji

- utworzyć repozytorium o nazwie:
PP_PTM_2013_nazwa_projektu
- dodać plik readme.txt z opisem projektu,
- dowiedzieć się do czego służy plik .gitignore (przeanalizować strukturę projektu w ColIDE i wygenerować reguły usuwające zbędne pliki),
- różnica w logowaniu przez HTTPS oraz SSH, zrealizować oba sposoby,
- dla zainteresowanych – wielu użytkowników preferuje inne wersje Gui dla git.
Przetestować dostępne programy (wybrany):
<http://git-scm.com/downloads/guis>
<https://code.google.com/p/tortoisegit/>

Bibliografia

<http://longair.net/blog/2009/04/16/git-fetch-and-merge/>. (2013, 03 05). git: fetch and merge, don't pull.

[http://pl.wikipedia.org/wiki/Git_\(oprogramowanie\)](http://pl.wikipedia.org/wiki/Git_(oprogramowanie)). (2013, 03 04). Git (oprogramowanie).

http://pl.wikipedia.org/wiki/System_kontroli_wersji. (2013, 03 04). System kontroli wersji.