Facultatea Calculatoare, Informatica si Microelectronica

Universitatea Tehnica a Moldovei

Medii Interactive de Dezvoltare a Produselor Soft

Lucrarea de laborator#1

Version Control Systems si modul de setare a unui server

Autor:
Artiom Nichifor

lector asistent:
Irina Cojanu

Lucrarea de laborator nr. 1

1 Scopul lucrarii de laborator

Formarea deprinderilor de lucru in VSC, acomodarea cu propritatile acestora si studiul functiilor, comenzilor si elementelor de baza.

2 Obiective

Version Control Systems (git —— bitbucket —— mercurial —— svn)

3 Desfasurarea lucrarii de laborator

3.1 Conditii si cerinte

Basic Level (nota 5 —— 6) :

- initializeaza un nou repositoriu
- configureaza-ti VCS
- crearea branch-urilor (creeaza cel putin 2 branches)
- commit pe ambele branch-uri (cel putin 1 commit per branch)

Normal Level (nota 7 —— 8):

- seteaza un branch to track a remote origin pe care vei putea sa faci push (ex. Github, Bitbucket or custom server)
- reseteaza un branch la commit-ul anterior
- salvarea temporara a schimbarilor care nu se vor face commit imediat.
- folosirea fisierului .gitignore

Advanced Level (nota 9 —— 10):

- merge 2 branches
- rezolvarea conflictelor a 2 branches
- comezile git care trebuie cunoscute

3.2 Analiza lucrarii de laborator

Linkul repozitoriului `https://github.com/artiomnichifor/midps`

Esenta lucrarii date de laborator a constat in acomodarea cu mediul de lucru al VCS si studiul lor. Creind un cont pe GitHub am avut posibilitatea de a lua cunostinata cu elementele si proprietatile unui VCS. La inceput am initializat si am clonat un repozitoriu, indeplinind si cerintele de modificae acestuia. Am creat a doua ramura si am efectuat mai multe commituri pentru fiecare, studiind si proprietatile de schimbare a ramurei prelucrate. Am evidentiat comenzile de eliminare a erorilor precum git reset si git revert pentru a aduce ramura la starea in care a fost imediat dupa ultimul commit si anularea ultimului commit respectiv. Am salvat temporar schimbarile fara a face commit prin intermediul comenzii git stash, pentru a fi utilizate ulterior. Am luat cunostinnta cu prpritatile fisierului .gitignore, care permite pastrarea unor fisiere doar in depozitoriul local, identificate dupa extensia sa sau dupa insasi numele sau. In sfirsit am efectuat concatinarea a doua ramuri si am cercetat erorile ce se pot petrce in urma acestora, care sunt cauzate de asemanrile dintre schimbarile in ramuri. Aceste erori pot fi usor identificate cu ajutorul functiei git status, identificindu- se nu doar fisierul in care s- a depistat conflictul ci si locul acesteia.

3.3 Imagini

```
nichi@DESKTOP-RUR9QT9 MINGW64 /d/midps (master)
$ git branch newvassal

nichi@DESKTOP-RUR9QT9 MINGW64 /d/midps (master)
$ git branch
* master
  newvassal

nichi@DESKTOP-RUR9QT9 MINGW64 /d/midps (master)
$ git checkout newvassal
Switched to branch 'newvassal'
```

Un exemplu de creare a unui nou branch si definirea acestuia ca ramura de prelucrat.

```
nichi@DESKTOP-RUR9QT9 MINGW64 /d/midps/lab 1 (master)
$ git log
commit dc44428f03c97a711f42a8ea36387c9f3c9ee0c1
Author: Artiom Nichifor <Artiom Nichifor>
Date:   Mon Feb 27 00:43:29 2017 +0200

    Revert "nodata"

    This reverts commit 91fd82244265aeced9815fe252e101d4292be45b.

commit 91fd82244265aeced9815fe252e101d4292be45b
Author: Artiom Nichifor <Artiom Nichifor>
Date:   Mon Feb 27 00:28:54 2017 +0200

    nodata
```

Demonstarea comenzii git log destinate afisarii listingului commiturilor si hashurilor sale utilizate pentru eliminarea commiturilor din git revert.

```
nichi@DESKTOP-RUR9QT9 MINGW64 /d/midps (master)
$ git stash list
stash@{0}: WIP on master: 03eabba Revert "4" x This reverts commit 04a476759a309
3655f00858257ebe21eb3743ed6.
```

Exemplu de utilizare a functiiei git stash list pentru pastrarea schimbarilor in buffer si utilizarea

ultioarea prin git stash aplly.

Concluzie

In cocluzie e trivial sa mentionam importantanta si necesitatea VCS precum git, in primul rind desigur ca posibilitatea de a inregistra orice modificare efectuata asupa fisierelor din repozitoriu si de a se reintoarce la orice moment al evolutiei proiectului. O alta proprietate find cea a ramurii, adica posibilitatea de ai oferi acetui proiect diferite cai sau branchuri de dezvoltare, sub influenta diferitor utilizatori. Ambele beneficii evidentiaza suparioritatea acestui sistem de gestionare, creind un mediu ideal pentru colaborarea utilizatorilor preocupati de efectuarea aceluiasi proiect. Utilitatea si eficacitatea adeveresc popularitatea metodei. In urma efectuarii lucrarii de laborator am constat ca gitul e un sistem logic cu greutate medie ba chiar simpla de sistematizare si intelegere, comenzile fiind usor de perceput, desi erorile in urma acestora fiind destul de dese la inceputul utilizarii.

References

1 GIT Tutorial: Commands, https://www.siteground.com/tutorials/git/commands.htm

2 Scott Chacon and Ben Straub , *Pro Git*, 2014