

FACULTATEA CALCULATOARE, INFORMATICĂ ȘI MICROELECTRONICĂ
UNIVERSITATEA TEHNICĂ A MOLDOVEI

MEDII INTERACTIVE DE DEZVOLTARE A PRODUSELOR SOFT
LUCRAREA DE LABORATOR#1

Version Control Systems și modul de setare a unui server

Autor:

Marina JECHIU

lector asistent:

Irina COJANU

Laboratory work #1

1 Scopul lucrării de laborator

Însusirea noțiunii de Version Control Systems și a modului de setare a unui server.

2 Obiective

Version Control Systems (git || *bitbucket* || *mercurial* || *svn*)

3 Condiția sarcinii

- inițializează un nou repository;
- configurează-ți VCS;
- crearea branch-urilor (cel puțin 2);
- commit pe ambele branch-uri;
- setează un branch to track a remote origin pe care vei putea să faci push ;
- salvarea temporară a schimbărilor care nu se vor face commit imediat;
- resetează un branch la commit-ul anterior;
- folosirea fișierului .gitignore;
- merge 2 branches;
- rezolvarea conflictelor a 2 branches.

4 Laboratory work implementation

4.1 Analiza lucrării de laborator

4.1.0.1 a <https://github.com/MarinaJechiuTI154/MIDPS>

Principalele noțiuni cu care voi opera:

- **repository** componenta server ce conține informații privind ierarhia de fișiere și reviziile.
- **branch** este o ramură secundară de dezvoltare a unui proiect.
- **checkout** preluarea în mediul local a unei anumite revizii publicate pe server.
- **commit** cerere de publicare pe server a unor modificări.
- **pull** acțiunea de actualizare (update) a informațiilor locale cu cele de pe server.
- **conflict** apare atunci când mai mulți utilizatori vor să publice modificări aplicate acelorași fișiere din proiect, însă sistemul de aplicare a versiunilor diferite nu poate îmbina modificările.
- **revert** revenirea la o versiune anterioară pe un anumit fir de dezvoltare (branch).
- **tag** branch “read-only” ce nu mai permite modificări ulterioare (folosit uneori pentru versiunile stabile și derivă dintr-un branch)

Am creat un cont public pe github cu denumirea mdps. Pentru a putea gestiona repositoryul am instalat GitBash. Pentru a activa contul avem nevoie să introducem în setări cheia, care se obține prin tastarea în linia de comandă: `ssh-keygen`. Pentru a deschide cheia obținută folosim următoarele instrucțiuni: `cat ~/.ssh/id - rsa.pub`. Pentru a face legătura între repositoryul pe github și cel local este necesar să clonăm repositoryul online cu instrucțiune: *git clone*.

La crearea repositoryului se creează un branch implicit numit master. Însă, dacă trebuie să creăm un nou branch folosim comanda: *git checkout 'denumire-branch'*. Astfel, nu doar se creează un nou branch, dar și se trece automat pe acest branch. În cazul în care dorim să trecem pe un branch deja existend, la tastarea denumirii acestuia sistemul îl recunoaște și face automat salt către acesta.

Deoarece repositoryile git sunt repository de tip distrib, pentru a încărca modificările efectuate local pe server este necesar pentru a efectua următoarii pași:

- *git add* . adăugarea în index a modificărilor realizate în directoriul meu, fișiere ce se intenționează a fi publicate.
git commit efectuarea commit-urilor în baza informației din index. Acestea pot conține denumiri pentru a gestiona mai ușor modificările.
- *git push* publicarea modificărilor pe repository.

4.2 Imagini

Adauga cite cel puțin o imagine (sau mai multe) pentru fiecare funcționalitate adăugată.

Concluzie

Aici trebuie sa fie concluzia ta.

article
test.