

UNIVERSITATEA TEHNICĂ A MOLDOVEI

FACULTATEA „CALCULATOARE, INFORMATICĂ ȘI
MICROELECTRONICĂ”

SPECIALITATEA : TEHNOLOGII INFORMATIONALE

DISCIPLINA : MEDII INTERACTIVE DE DEZVOLTARE A
PRODUSELOR SOFT

Lucrarea de laborator nr.1

Version control systems

A efectuat:

st. gr. TI-141

LEVCENCO ELENA

A verificat:

COJANU IRINA

January 28, 2017

1 Scopul lucrării:

Studierea unui Version Control System și lucru cu versiunea CLI a acestuia. Examinarea instrucțiunilor de bază și lucru cu un repository remote.

2 Obiective:

- Înțelegerea și folosirea CLI (basic level)
- Administrarea remote a mașinilor linux machine folosind SSH (remote code editing)
- Version Control Systems (git — bitbucket — mercurial — svn)

3 Laboratory Requirements: folosind drept VSC github

3.1 Basic Level (nota 5 .. 6) :

- initializeaza un nou repository
- configurarea VCS
- crearea branch-urilor (cel puțin 2 branches)
- commit pe ambele branch-uri (cel puțin 1 commit per branch)

3.2 Normal Level (nota 7 .. 8):

- setarea unui branch to track a remote origin pe care se face push (Github)
- resetarea unui branch la commit-ul anterior
- folosirea fișierului .gitignore

3.3 Advanced Level (nota 9 .. 10):

- merge 2 branches
- rezolvarea conflictelor a 2 branches

4 Concluzia:

Primul pas este crearea unui repository. Pentru aceasta dăm click pe + New Repository. Următorul pas este alegerea unui nume pentru repository. Bifăm „Initialize this repository with a README”, alegem în partea de jos a paginii, adăugarea unui fișier .gitignore (Fișierele sau directoarele ce se doresc a fi ignorate de sistemul de versionare). Pasul trei identificarea utilizatorului curent Comenzi linie de comanda:

- git config user.name “Nume Utilizator”
- git config user.email “email@gmail.com”

Sincronizarea cu repositoryul. Comenzi linie de comanda:

- git init (initializare workspace pentru GIT)
- git remote (asigura management-ul conexiunilor realizate cu repository-urile remote; comenzi aditionale: add, -v [listare], rm, rename)
- ssh-keygen -t rsa -b 4096 -C "email@gmail.com"
ssh-agent bash -c 'ssh-add /key; git clone git@github.com:ElenaLevcenco/MIDPS.git'
(cloneaza un repositoryul)

Commit-erea

- git status (afiseaza o lista cu fisierele modificate fata de ultima versiune sincronizata)
- git add text.txt undo.txt ;...; (adaugarea de fisiere in cadrul versiunii ce se intentioneaza a fi publicata)
- git commit -m “primul commit” (demarcarea cu ajutorul unui mesaj a noi versiuni ce urmeaza a fi publicata; commit-ul va avea atasat o versiune a reviziei [un hash], precum si detaliile utilizatorului [identitatea lui])

Actualizarea versiunii curente a unui branch

Comenzi linie de comanda:

- `git pull origin sensei`

Migrarea intre branch-uri

- `git branch` (va afisa branch-urile locale, precum si branch-ul curent pe care ne aflam)
- `git checkout guru` (va schimba branch-ul curent in 'sensei', migrandu-se astfel pe un nou fir al versionarii)
- `git checkout -b 'guru'` sau `git branch 'guru'` (va crea un branch nou pe baza celui curent)

Unirea a doua branch-uri

Comenzi linie de comanda:

- `git merge 'master'` (va aplica modificarile din branch-ul specificat branch-ului curent, daca acestea exista; branch-ul de unit poate fi atat local, cat si remote, deci nesincronizat local)

Rezolvarea conflictelor:

```
Aliona@DESKTOP-MEHBWPB MINGW64 /d/University/MIDPS/MIDPS (guru)
$ git pull origin sensei
From https://github.com/ElenaLevcenco/MIDPS
 * branch          sensei      -> FETCH_HEAD
Auto-merging Lab1/test.txt
CONFLICT (content): Merge conflict in Lab1/test.txt
Automatic merge failed; fix conflicts and then commit the result.
```

<pre> <<<<<< HEAD Some text here.Conflict situation ===== Some text here.Let's try to create co >>>>>> sensei </pre>	<pre> Some text here.Let's try to create con </pre>
<pre> ~ ~ ~ ~ ~ ~ ~ ~ </pre>	<pre> ~ ~ ~ ~ ~ ~ ~ ~ </pre>

<84.txt [dos] (21:46 14/11/2016)1,1 All <84.txt [unix] (21:46 14/11/2016)1,1 All

```

<<<<<< HEAD
<<<<<< HEAD
Some text here.Conflict situation
=====
Some text here.Let's try to create conflict...
>>>>>> sensei
=====
Some text here.Let's try to create conflict...first time fatality
>>>>>> 5397a62195602422da067a7c5376a5d205000a6b

```

```

~
~
~
~
~
~
~
~

```

```
Normal merge conflict for 'Lab1/test.txt':
  {local}: modified file
  {remote}: modified file
Hit return to start merge resolution tool (vimdiff):
4 files to edit
Stopped

Aliona@DESKTOP-MEHBRPB MINGW64 /d/University/MIDPS/MIDPS (guru|MERGING)
$ git checkout --ours Lab1/test.txt

Aliona@DESKTOP-MEHBRPB MINGW64 /d/University/MIDPS/MIDPS (guru|MERGING)
$ git checkout --theirs Lab1/test.txt

Aliona@DESKTOP-MEHBRPB MINGW64 /d/University/MIDPS/MIDPS (guru|MERGING)
$ git add Lab1/test.txt

Aliona@DESKTOP-MEHBRPB MINGW64 /d/University/MIDPS/MIDPS (guru|MERGING)
$ git commit -m "Mission completed"
[guru f390c20] Mission completed

Aliona@DESKTOP-MEHBRPB MINGW64 /d/University/MIDPS/MIDPS (guru)
$ git pull origin guru
From https://github.com/ElenaLevcenco/MIDPS
 * branch          guru          -> FETCH_HEAD
Already up-to-date.
```

Concluzia

Laboratorul dat isi propune insusirea comenzilor de baza a sistemului de versionare GIT de la linia de comanda. Sistemele de versionare permit gestionarea versiunilor multiple a unor fisiere, precum si asigurarea lucrului colaborativ asupra acestor fisiere. Aceste sisteme au fost concepute pentru a permite membrilor mai multor echipe sa opereze modificari pe acelasi proiect, aceste modificari urmand a fi reunite intr-o noua versiune a proiectului.