

Medii Interactive de Dezvoltare A Produselor Soft

Lucrare de laborator #1

Version Control System și modul de setare a unui server

autor: Popa Igor

TI-152

lector asistent: Irina Cojanu

lector superior: Radu Melnic

1. Sarcini și obiective:

2. *Basic Level* (nota 5 || 6) :

- a. inițializează un nou repository
- b. configurează-ți VCS
- c. crearea branch-urilor (creează cel puțin 2 branches)
- d. commit pe ambele branch-uri (cel puțin 1 commit per branch)

3. *Normal Level* (nota 7 || 8):

- a. setează un branch to track a remote origin pe care vei putea să faci push (ex. Github, Bitbucket or custom server)
- b. resetează un branch la commit-ul anterior
- c. salvareă temporară a schimbărilor care nu se vor face commit imediat.
- d. folosirea fișierului .gitignore

2. Analiza lucrării de laborator:

2.1 Basic level

-Inițializarea unui repository:

Pentru inițializarea unui nou repository apăsăm butonul “+” după care facem click pe “New repository”

-Configurarea VCS:

Pentru configurarea VCS vom folosi următoarele comenzi în linia de comandă gitbash:

git config --global user.name “user.name”

git config --global user.email email@domain.com

-Crearea branch-urilor (minim 2)

Pentru crearea unei ramuri în linia de comandă vom folosi următoarele comenzi:

git checkout –b branchname

iar pentru a vizualiza branch-urile existente:

git branch

-commit pe ambele branch-uri (cel puțin un commit per branch)

Pentru a face commit vom utiliza comanda:

git commit -m "fileName"

2.2 Basic level

-Setează un branch to track a remote origin pe care vei putea sa faci push (ex. Github, Bitbucket or custom server)

În cazul dat vomo folosi comanda:

git remote add origin <http://github.com//username/repository.git>

-Resetarea unui branch la commitul anterior

Pentru a reseta un branch la commitul anterior e nevoie să avem careva acțiuni efectuate după care dorim să revenim pe un branch anterior. Această revenire o putem face prin utilizarea comenzii hard reset sau soft reset. În cazul meu voi folosi hard reset.

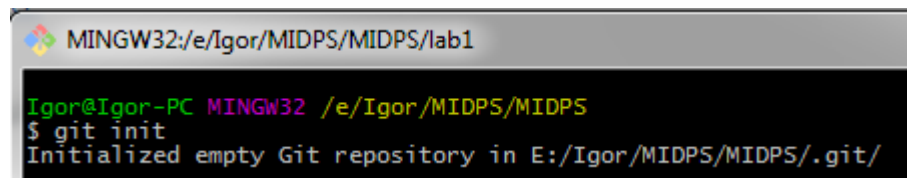
git reset --hard HEAD^ (în cazul dat ne vom întoarce înapoi cu un pas, pentru a o face cu mai mulți putem folosi **HEAD~x**Pasi)

-Salvarea temporară a schimbărilor la care nu se va face commit imediat:

Pentru a salva schimbările efectuate la un moment dat fără a face commit la moment vom folosi comanda:

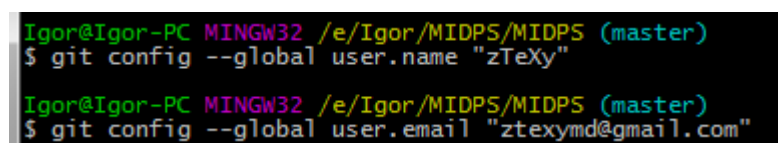
git stash

2.3 Screenshot-uri



```
MINGW32:/e/Igor/MIDPS/MIDPS/lab1
Igor@Igor-PC MINGW32 /e/Igor/MIDPS/MIDPS
$ git init
Initialized empty Git repository in E:/Igor/MIDPS/MIDPS/.git/
```

img.1 inițializarea unui nou repository



```
Igor@Igor-PC MINGW32 /e/Igor/MIDPS/MIDPS (master)
$ git config --global user.name "zTeXy"
Igor@Igor-PC MINGW32 /e/Igor/MIDPS/MIDPS (master)
$ git config --global user.email "ztexymd@gmail.com"
```

img.2 configurare VCS

```
Igor@Igor-PC MINGW32 /e/Igor/MIDPS/MIDPS (master)
$ git pull
remote: Counting objects: 11, done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 11 (delta 1), reused 2 (delta 0), pack-r
Unpacking objects: 100% (11/11), done.
From https://github.com/zTeXy/MIDPS
 * [new branch]      branch-1 -> origin/branch-1
 * [new branch]      master -> origin/master
```

img3. copierea de pe origin pe local

```
MINGW32:/e/Igor/MIDPS/MIDPS/lab1

Igor@Igor-PC MINGW32 /e/Igor/MIDPS/MIDPS (master)
$ git checkout -b fisrtBranch
Switched to a new branch 'fisrtBranch'

Igor@Igor-PC MINGW32 /e/Igor/MIDPS/MIDPS (fisrtBranch)
$ git branch
* fisrtBranch
  master

Igor@Igor-PC MINGW32 /e/Igor/MIDPS/MIDPS (fisrtBranch)
$ git checkout -
Switched to branch 'master'

Igor@Igor-PC MINGW32 /e/Igor/MIDPS/MIDPS (master)
$ git checkout -b secondBranch
Switched to a new branch 'secondBranch'

Igor@Igor-PC MINGW32 /e/Igor/MIDPS/MIDPS (secondBranch)
$ git checkout fisrtBranch
Switched to branch 'fisrtBranch'

Igor@Igor-PC MINGW32 /e/Igor/MIDPS/MIDPS (fisrtBranch)
```

img.4 crearea branch-urilor

```
Igor@Igor-PC MINGW32 /e/Igor/MIDPS/MIDPS/lab1 (fisrtBranch)
$ git add file1.txt

Igor@Igor-PC MINGW32 /e/Igor/MIDPS/MIDPS/lab1 (fisrtBranch)
$ git commit
[fisrtBranch 351f0b5] Add file1
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Lab1/file1.txt
```

img5. commit primul branch

```
Igor@Igor-PC MINGW32 /e/Igor/MIDPS/MIDPS/lab1 (secondBranch)
$ git add file2.txt

Igor@Igor-PC MINGW32 /e/Igor/MIDPS/MIDPS/lab1 (secondBranch)
$ git commit -m "Add file2"
[secondBranch a77b8c7] Add file2
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Lab1/file2.txt
```

img6. commit al doilea branch

```
Igor@Igor-PC MINGW32 /e/Igor/MIDPS/MIDPS (master)
$ git remote add origin https://github.com/zTeXy/MIDPS.git
```

img.7 remote

```
Igor@Igor-PC MINGW32 /e/Igor/MIDPS/MIDPS/lab1 (master)
$ git checkout fisrtBranch
Switched to branch 'fisrtBranch'

Igor@Igor-PC MINGW32 /e/Igor/MIDPS/MIDPS/lab1 (fisrtBranch)
$ git add file3.txt

Igor@Igor-PC MINGW32 /e/Igor/MIDPS/MIDPS/lab1 (fisrtBranch)
$ git commit -m "Add file3"
[fisrtBranch 2b9c3ba] Add file3
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Lab1/file3.txt

Igor@Igor-PC MINGW32 /e/Igor/MIDPS/MIDPS/lab1 (fisrtBranch)
$ git reset --hard HEAD^
HEAD is now at 351f0b5 Add file1

Igor@Igor-PC MINGW32 /e/Igor/MIDPS/MIDPS/lab1 (fisrtBranch)
$ git status
On branch fisrtBranch
nothing to commit, working tree clean
```

img.8 resetarea unui branch la un commit anterior

```
Igor@Igor-PC MINGW32 /e/Igor/MIDPS/MIDPS/lab1 (secondBranch)
$ git status
On branch secondBranch
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        file4.txt

nothing added to commit but untracked files present (use "git add" to track)

Igor@Igor-PC MINGW32 /e/Igor/MIDPS/MIDPS/lab1 (secondBranch)
$ git add file4.txt

Igor@Igor-PC MINGW32 /e/Igor/MIDPS/MIDPS/lab1 (secondBranch)
$ git stash
Saved working directory and index state WIP on secondBranch: a77b8c7 Add file2
HEAD is now at a77b8c7 Add file2
```

img9. salvarea temporară fără a face commit

3. Concluzii:

În urma efectuării aceste lucrări de laborator m-am am însușit bazele platformei github. Am aflat avantajele și dezavantajele. Am utilizat un șir de comenzi de bază care pe viitor îmi vor permite să mă împărtășesc cu proiectele efectuate, să fac modificări în proiectele străine și nu numai. Datorită acestui laborator am ajuns la concluzia că git-ul este o unealtă nu atât de simplă pe cât pare dar care merită însușită destul de adânc deoarece ne oferă o gamă largă de posibilități.

4. Bibliografie:

<http://stackoverflow.com>
<https://try.github.io/levels/1/challenges/1>
<https://git-scm.com/>

