

Facultatea Calculatoare, Informatica si  
Microelectronică  
Universitatea Tehnică a Moldovei

Medii Interactive de Dezvoltare a Produselor Soft  
Lucrarea de laborator#1

---

## **Version Control Systems si modul de setare a unui server**

---

A efectuat: studentul gr. TI-151 Pastuh Vitalie

A verificat: Cojanu Irina

Chişinău 2017

## Obiective:

- Version Control Systems (git)

## Cerințele laboratorului:

- *Basic Level :*
  - initializeaza un nou repository
  - configureaza-ti VCS
  - crearea branch-urilor (creeaza cel puțin 2 branches)
  - commit pe ambele branch-uri (cel puțin 1 commit per branch)
- *Normal Level :*
  - 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 :*
  - merge 2 branches
  - rezolvarea conflictelor a 2 branches
  - comenzile git care trebuie cunoscute

## Analiza Lucrării de laborator:

Link-ul la repository <https://github.com/Pastuh2/MIDPS>

Am creat repositoryul prin metoda online. Am deschis pagina mea pe github.com, click pe Repositories și apoi pe butonul New. Atunci când am creat repositoryul MIDPS, l-am inițializat cu un fișier README.

Următorul pas constă în configurarea git-ului. Configurăm numele și email-ul prin comenzile **git config --global user.name "NUMELE"** **git config --global user.email "EMAIL"**.

```
MINGW64:/c/Users/vitalik/Desktop

vitalik@vitalik-MINGW64 ~/Desktop
$ git config --global user.name "Pastuh"

vitalik@vitalik-MINGW64 ~/Desktop
$ git config --global user.email "373688898073@yandex.ru"

vitalik@vitalik-MINGW64 ~/Desktop
$ |
```

Urmează generarea cheii SSH pe care o vom copia în setările de pe github.

```
MINGW64:/c/Users/vitalik/Desktop

$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/vitalik/.ssh/id_rsa): git_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in git_rsa.
Your public key has been saved in git_rsa.pub.
The key fingerprint is:
SHA256:bNoQKPJ1JkGuzMj7GKYZPJvDprFkZw1uHmp6qtcjw9Q vitalik@vitalik-
The key's randomart image is:
+----[RSA 2048]-----+
|      .o                |
|      .o                |
|    . = +              |
|   . = + + o           |
|  ..=o . S             |
| . + E =               |
|+@ B .. .              |
| =X^ +                  |
| #O. = .                |
+----[SHA256]-----+

vitalik@vitalik-MINGW64 ~/Desktop
$
```

Search GitHub

Pull requests

Issues

Gist

Personal settings

Profile

Account

Emails

Notifications

Billing

SSH a

Secur

Blocke

Repos

Organ

Saved


Autho

Install

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

 Vitalie

Fingerprint: 86:b6:49:a0:65:0a:34:ef:b5:6e:c6:90:2f:57:9b:6c

SSH

Added on 20 Feb 2017 — Never used

Delete

Check out our guide to generating SSH keys or troubleshoot common SSH Problems.

C:\Users\vitalik\.ssh\id\_rsa.pub - Notepad++

Файл Правка Поиск Вид Кодировки Синтаксисы Опции Tools Макросы Запуск Плагины Окна ?

1 ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQAC6WYUW2DU3HaJB0sbYcqzkH3uAm4wUeRA9aHq1QIGBbFnSOzHbSZ1cvq3EyTN9by2OPEya8B4cmXkiwHCx/+JmjbEEF

După ce am generat keygen-ul,clonăm repozitoriul pe mașina locală.

```
MINGW64:/c/Users/vitalik/Desktop/MIDPS
vitalik@vitalik: MINGW64 ~/Desktop
$ git clone https://github.com/Pastuh2/MIDPS.git
Cloning into 'MIDPS'...
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.

vitalik@vitalik: MINGW64 ~/Desktop
$ cd MIDPS

vitalik@vitalik: MINGW64 ~/Desktop/MIDPS (master)
$ ls -l
total 1
-rw-r--r-- 1 vitalik 197121 7 feb 20 20:39 README.md

vitalik@vitalik: MINGW64 ~/Desktop/MIDPS (master)
$ |
```

Pentru a adăuga fișiere pe repozitoriu, vom folosi următoarele comenzi: **git add \*** - comanda indexează toate fișierele. **git commit -m** - comanda face un snapshot la toate schimbările noastre.

**git push origin master** - comanda încarcă toate fișierele indexate pe git. Totodată vom folosi **git status** și **git show** pentru a ne asigura că fișierele au fost adăugate în repozitoriu.

```
vitalik@vitalik: MINGW64 ~/Desktop/MIDPS (master)
$ git add *

vitalik@vitalik: MINGW64 ~/Desktop/MIDPS (master)
$ git commit -m "Text"
[master 7ccde3c] Text
4 files changed, 12 insertions(+)
create mode 100644 lab2/README.md
create mode 100644 lab3/README.md
create mode 100644 lab4/README.md
create mode 100644 lab5/README.md

vitalik@vitalik: MINGW64 ~/Desktop/MIDPS (master)
$ git push origin master
Warning: Permanently added the RSA host key for IP address '192.30.253.112' to the list of known hosts.
Counting objects: 3, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 347 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:Pastuh2/MIDPS.git
   4a37695..7ccde3c  master -> master

vitalik@vitalik: MINGW64 ~/Desktop/MIDPS (master)
$ |
```

Revenirea la o versiune mai veche poate fi efectuată cu ajutorul comenzii **git reset -TYPE "codul comitului"**. Există diferența între **-soft** și **-hard**, când facem soft reset indexurile rămân neschimbate. Iar în cazul în care facem hard reset, pierdem indexurile.

Am creat un fișier nou text.txt în versiunea 1. După care l-am șters și am făcut commit la versiunea 2 în care am sters fișierul test.txt. Dorim să revenim la versiunea 1. La început vom lansa comanda **git log** care ne arată logul de commituri și codul pentru fiecare commit. Vom avea nevoie de primele 7 cifre la comitul anterior.

```

MINGW64:/c:/Users/vitalik/Desktop/MIDPS
vitalik@vitalik- MINGW64 ~/Desktop/MIDPS (master)
$ git add *
vitalik@vitalik- MINGW64 ~/Desktop/MIDPS (master)
$ git commit -m "versiunea 1"
[master 491fe76] versiunea 1
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Lab1/ignore.txt
create mode 100644 Lab1/test.txt
vitalik@vitalik- MINGW64 ~/Desktop/MIDPS (master)
$ git push origin master
Counting objects: 4, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 353 bytes | 0 bytes/s, done.
Total 4 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local objects.
To github.com:Pastuh2/MIDPS.git
8b43cf2..491fe76 master -> master
vitalik@vitalik- MINGW64 ~/Desktop/MIDPS (master)
$ ls -l
total 1
drwxr-xr-x 1 vitalik 197121 0 feb 20 21:43 Lab1/
drwxr-xr-x 1 vitalik 197121 0 feb 20 21:10 lab2/
drwxr-xr-x 1 vitalik 197121 0 feb 20 21:10 lab3/
drwxr-xr-x 1 vitalik 197121 0 feb 20 21:10 lab4/
drwxr-xr-x 1 vitalik 197121 0 feb 20 21:10 lab5/
-rw-r--r-- 1 vitalik 197121 25 feb 20 21:04 README.md
vitalik@vitalik- MINGW64 ~/Desktop/MIDPS (master)
$ ls
Lab1/ lab2/ lab3/ lab4/ lab5/ README.md
vitalik@vitalik- MINGW64 ~/Desktop/MIDPS (master)
$ git rm test.txt
fatal: pathspec 'test.txt' did not match any files
vitalik@vitalik- MINGW64 ~/Desktop/MIDPS (master)
$ rm 'Lab1/test.txt'
vitalik@vitalik- MINGW64 ~/Desktop/MIDPS (master)
$ ls
Lab1/ lab2/ lab3/ lab4/ lab5/ README.md
vitalik@vitalik- MINGW64 ~/Desktop/MIDPS (master)
$ git add *
vitalik@vitalik- MINGW64 ~/Desktop/MIDPS (master)
$ git commit -m "version 2"
[master cb13c01] version 2
1 file changed, 0 insertions(+), 0 deletions(-)
delete mode 100644 Lab1/test.txt
vitalik@vitalik- MINGW64 ~/Desktop/MIDPS (master)
$ git pull origin master
From github.com:Pastuh2/MIDPS
* branch master -> FETCH_HEAD
Already up-to-date.
vitalik@vitalik- MINGW64 ~/Desktop/MIDPS (master)
$ git push origin master
Counting objects: 3, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 284 bytes | 0 bytes/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:Pastuh2/MIDPS.git
491fe76..cb13c01 master -> master

```

Acum folosim comenzile **git reset --hard** și **git reset --soft**

```

vitalik@vitalik- MINGW64 ~/Desktop/MIDPS (master)
$ git reset --hard 7ccde3c
HEAD is now at 7ccde3c Text
vitalik@vitalik- MINGW64 ~/Desktop/MIDPS (master)
$ ls
Lab1/ lab2/ lab3/ lab4/ lab5/ README.md
vitalik@vitalik- MINGW64 ~/Desktop/MIDPS (master)
$ ls Lab1
README.md
vitalik@vitalik- MINGW64 ~/Desktop/MIDPS (master)
$ git reset --soft 7ccde3c
vitalik@vitalik- MINGW64 ~/Desktop/MIDPS (master)
$ ls Lab1
README.md
vitalik@vitalik- MINGW64 ~/Desktop/MIDPS (master)
$ git reset --soft 87e8346
vitalik@vitalik- MINGW64 ~/Desktop/MIDPS (master)
$ ls Lab1
README.md
vitalik@vitalik- MINGW64 ~/Desktop/MIDPS (master)
$

```

VCS ne permite să avem mai multe **branch-uri**. Branch-urile sunt comod de folosit când dorim să lucrăm paralel la un proiect și apoi dorim să unim toate modificările.

**git branch "name"** - creează un branch nou cu numele "name". **git branch** - vizualizarea branch-urilor (\* indică branch-ul curent). **git branch -d "nume"** - șterge branch-ul "nume". **git checkout -b "name"** - creează un branch nou cu numele "name" și face switch la el.

```
vitalik@vitalik- [1] MINGW64 ~/Desktop/MIDPS (nou)
$ git branch
  copie
  master
* nou

vitalik@vitalik- [1] MINGW64 ~/Desktop/MIDPS (nou)
$ git checkout copie
Switched to branch 'copie'

vitalik@vitalik- [1] MINGW64 ~/Desktop/MIDPS (copie)
$ git branch -d copie
error: Cannot delete branch 'copie' checked out at 'C:/Users/vitalik/Desktop/MIDPS'

vitalik@vitalik- [1] MINGW64 ~/Desktop/MIDPS (copie)
$ git checkout nou
Switched to branch 'nou'

vitalik@vitalik- [1] MINGW64 ~/Desktop/MIDPS (nou)
$ git branch -d copie
Deleted branch copie (was c56e72d).
```

```
vitalik@vitalik- [1] MINGW64 ~/Desktop/MIDPS (nou)
$ git checkout -b nou1
Switched to a new branch 'nou1'

vitalik@vitalik- [1] MINGW64 ~/Desktop/MIDPS (nou1)
$ ls
Lab1/  lab2/  lab3/  lab4/  lab5/  README.md

vitalik@vitalik- [1] MINGW64 ~/Desktop/MIDPS (nou1)
$ ls 'lab1'
ignore.txt  README.md  test.txt

vitalik@vitalik- [1] MINGW64 ~/Desktop/MIDPS (nou1)
$ git add *

vitalik@vitalik- [1] MINGW64 ~/Desktop/MIDPS (nou1)
$ git commit -m "branch nou"
On branch nou1
Untracked files:
  (use "git add" to track)
        .swp

nothing added to commit but untracked files present

vitalik@vitalik- [1] MINGW64 ~/Desktop/MIDPS (nou1)
$ git push origin nou1
Total 0 (delta 0), reused 0 (delta 0)
To github.com:Pastuh2/MIDPS.git
 * [new branch]      nou1 -> nou1
```

```

vitalik@vitalik- [MINGW64 ~/Desktop/MIDPS (nou1)]
$ git push origin nou1
Total 0 (delta 0), reused 0 (delta 0)
To github.com:Pastuh2/MIDPS.git
 * [new branch]      nou1 -> nou1

vitalik@vitalik- [MINGW64 ~/Desktop/MIDPS (nou1)]
$ git checkout master
Your branch is up-to-date with 'origin/master'.
Switched to branch 'master'

vitalik@vitalik- [MINGW64 ~/Desktop/MIDPS (master)]
$ git branch
* master
  nou
  nou1

vitalik@vitalik- [MINGW64 ~/Desktop/MIDPS (master)]
$ vim to_merge
Файлов для редактирования: 2

vitalik@vitalik- [MINGW64 ~/Desktop/MIDPS (master)]
$ cat to_merge
Vitalie Pastuh
cat: merge: No such file or directory

vitalik@vitalik- [MINGW64 ~/Desktop/MIDPS (master)]
$ cat to_merge
cat: to_merge: No such file or directory

vitalik@vitalik- [MINGW64 ~/Desktop/MIDPS (master)]
$ git checkout nou
Switched to branch 'nou'

vitalik@vitalik- [MINGW64 ~/Desktop/MIDPS (nou)]
$ vim to_merge

vitalik@vitalik- [MINGW64 ~/Desktop/MIDPS (nou)]
$ cat to_merge
Vitalie Pastuh

```

```

$ git config --global merge.tool kdiff3

vitalik@vitalik- [MINGW64 ~/Desktop/MIDPS (master)]
$ git merge nou
Already up-to-date.

vitalik@vitalik- [MINGW64 ~/Desktop/MIDPS (master)]
$ git mergetool mergeing: origin master
No files need merging

vitalik@vitalik- [MINGW64 ~/Desktop/MIDPS (master)]
$ ls
Lab1/  lab2/  lab3/  lab4/  lab5/  README.md  to_merge

vitalik@vitalik- [MINGW64 ~/Desktop/MIDPS (master)]
$ ls 'lab1'
ignore.txt  README.md  test.txt

vitalik@vitalik- [MINGW64 ~/Desktop/MIDPS (master)]
$ cat to_merge
Vitalie Pastuh

```

**Concluzie:** Am studiat VCS.Mi-am aprofundat cunoștințele în GitHub.Am învățat cum se creează mai multe branch-uri,cum se mută de la unul la altul,să fac operațiile de resetare la commit-ul anterior.Am aplicat comenzile fundamentale.Consider că fiecare programator trebuie să cunoască GitHub,să lucreze cu VCS. Chiar daca am avut problem cu conexiunea ssh am rezolvat problema cu ajutorului forumului Github.com(Redactind config din mapa .git inlocuiid Http cu SSH problema cu Git push origin master s-a rezolvat ce mi-a permis de a incarca fisierele pe repozitoriul meu fara probleme.

### **Am lucrat cu comenzile de linie cum ar fi :’’ current state**

```
git status list which (unstaged) files have changed
git diff list (unstaged) changes to files
git log list recent commits
git add fn stage file
git commit -m 'message' commit file
git commit -am 'message' add/commit all changes from all tracked files (no untracked files) in one go
git status
git fetch origin
etc.
```