

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 Onta Vladislav

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"**.

```
Godzila@Godzila MINGW64 ~/Desktop (master)
$ git config --global user.name "Onta"

Godzila@Godzila MINGW64 ~/Desktop (master)
$ git config --global user.email "vladislav.onta@gmail.com"

Godzila@Godzila MINGW64 ~/Desktop (master)
$
```

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

```
Godzila@Godzila MINGW64 ~/Desktop (master)
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Godzila/.ssh/id_rsa):
Created directory '/c/Users/Godzila/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Godzila/.ssh/id_rsa.
Your public key has been saved in /c/Users/Godzila/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:WsdBVw/hp6GPd0YSCvPQFaxjJPqpJD8J2ISF1tCRGuk Godzila@Godzila
The key's randomart image is:
+----[RSA 2048]-----+
|  .*.o . .o*o |
|  =  = ..o.o.o |
| o =  = .o..o.. |
| E . . . *+..o |
| + S.o.o.o.o.o |
| . +o..o oo. |
| . = o . oo |
|  =  = ... |
|  .  .  |
+----[SHA256]-----+

Godzila@Godzila MINGW64 ~/Desktop (master)
$
```

Personal settings

Profile

Account

Emails

Notifications

Billing

SSH and GPG keys

Security

Blocked users

Repositories

Organizations

SSH keys

New SSH key

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

Onta

Fingerprint: 27:39:f8:36:61:c0:80:75:05:58:b4:43:67:71:33:ef

Added on 19 May 2017

Never used

Delete

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH Problems](#).

GPG keys

New GPG key

There are no GPG keys with access to your account.

Learn how to [generate a GPG key](#) and add it to your account.

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

```
MINGW64/c/Users/Godzila/Desktop/MIDPS
Godzila@Godzila MINGW64 ~/Desktop (master)
$ git clone https://github.com/VladislavOnta/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.

Godzila@Godzila MINGW64 ~/Desktop (master)
$ cd MIDPS

Godzila@Godzila MINGW64 ~/Desktop/MIDPS (master)
$ ls -l
total 1
-rw-r--r-- 1 Godzila 197609 7 mai 19 10:50 README.md

Godzila@Godzila MINGW64 ~/Desktop/MIDPS (master)
$ -
```

Pentru a adăuga fișiere pe repository, 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 repository.

```
MINGW64/c/Users/Godzila/Desktop/MIDPS
Godzila@Godzila MINGW64 ~/Desktop/MIDPS (master)
$ git add *

Godzila@Godzila MINGW64 ~/Desktop/MIDPS (master)
$ git commit
[master b9a0a87] Incarcare reusita!
5 files changed, 5 insertions(+)
create mode 100644 lab 1/README.md
create mode 100644 lab 2/README.md
create mode 100644 lab 3/README.md
create mode 100644 lab 4/README.md
create mode 100644 lab 5/README.md

Godzila@Godzila MINGW64 ~/Desktop/MIDPS (master)
$ git push origin master
Counting objects: 2, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 284 bytes | 0 bytes/s, done.
Total 2 (delta 0), reused 0 (delta 0)
To https://github.com/VladislavOnta/MIDPS.git
 4205522..b9a0a87  master -> master

Godzila@Godzila 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/Godzila/Desktop/MIDPS
Godzila@Godzila MINGW64 ~/Desktop/MIDPS (master)
$ git branch
* master
  versiune1

Godzila@Godzila MINGW64 ~/Desktop/MIDPS (master)
$ git branch -d "versiune1"
Deleted branch versiune1 (was b9a0a87).

Godzila@Godzila MINGW64 ~/Desktop/MIDPS (master)
$ git branch "origin"

Godzila@Godzila MINGW64 ~/Desktop/MIDPS (master)
$ git branch
* master
  origin

Godzila@Godzila MINGW64 ~/Desktop/MIDPS (master)
$ git checkout -b 'new'
Switched to a new branch 'new'

Godzila@Godzila MINGW64 ~/Desktop/MIDPS (new)
$ git branch
* new
  master
  origin

Godzila@Godzila MINGW64 ~/Desktop/MIDPS (new)
$
```

```
MINGW64/c/Users/Godzila/Desktop/MIDPS
Godzila@Godzila MINGW64 ~/Desktop/MIDPS (new)
$ ls
'lab 1'/ 'lab 2'/ 'lab 3'/ 'lab 4'/ 'lab 5'/ README.md

Godzila@Godzila MINGW64 ~/Desktop/MIDPS (new)
$ git add *

Godzila@Godzila MINGW64 ~/Desktop/MIDPS (new)
$ git commit -m "new branch"
On branch new
nothing to commit, working tree clean

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

Godzila@Godzila MINGW64 ~/Desktop/MIDPS (new)
$
```

```
MINGW64:/c/Users/Godzila/Desktop/MIDPS
Godzilla@Godzilla MINGW64 ~/Desktop/MIDPS (new)
$ git branch
  master
* new
  origin

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

Godzilla@Godzilla MINGW64 ~/Desktop/MIDPS (master)
$ git checkout new
Switched to branch 'new'

Godzilla@Godzilla MINGW64 ~/Desktop/MIDPS (new)
$
```

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.

```
MINGW64: c:/Users/Godzila/Desktop/MIDPS

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

Godzila@Godzila MINGW64 ~/Desktop/MIDPS (master)
$ vim to_merge

Godzila@Godzila MINGW64 ~/Desktop/MIDPS (master)
$ cat to_merge
hai odata

Godzila@Godzila MINGW64 ~/Desktop/MIDPS (master)
$ vim to_merge

Godzila@Godzila MINGW64 ~/Desktop/MIDPS (master)
$ cat to_merge
hai odata
hai de 2 ori

Godzila@Godzila MINGW64 ~/Desktop/MIDPS (master)
$
```

```
MINGW64: c:/Users/Godzila/Desktop/MIDPS

Godzila@Godzila MINGW64 ~/Desktop/MIDPS (master)
$ git push origin
Counting objects: 3, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 298 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/VladislavOnta/MIDPS.git
   b9a0a87..d8b884a  master -> master

Godzila@Godzila MINGW64 ~/Desktop/MIDPS (master)
$
```

```
Выбрав MINGW64/c/Users/Godzilla/Desktop/MIDPS
Godzilla@Godzilla MINGW64 ~/Desktop/MIDPS (master)
$ git checkout new
Switched to branch 'new'

Godzilla@Godzilla MINGW64 ~/Desktop/MIDPS (new)
$ git commit -m "hello"
On branch new
nothing to commit, working tree clean

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

Godzilla@Godzilla MINGW64 ~/Desktop/MIDPS (master)
$ git commit -m "buna seara"
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working tree clean

Godzilla@Godzilla MINGW64 ~/Desktop/MIDPS (master)
$ git push origin
Everything up-to-date

Godzilla@Godzilla MINGW64 ~/Desktop/MIDPS (master)
$
```



**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.
```