

**Facultatea Calculatoare, Informatica si
Microelectronica
Universitatea Tehnica a Moldovei**

Medii Interactive de Dezvoltare a
Produselor Soft
Lucrarea de laborator Nr.1

Version Control Systems si modul de setare a unui server

A efectuat : **Dobrin Eduard**

lector asistent : **Cojanu Irina**

lector superior : **Melnic Radu**

Lucrarea de laborator Nr.1

1. Scopul lucrarii de laborator :

De a se invata utilizarea unui Version Control System si modul de setare a unui server.

2. Obiective

Studiarea Version Control Systems (git).

3. Mersul lucrării de laborator

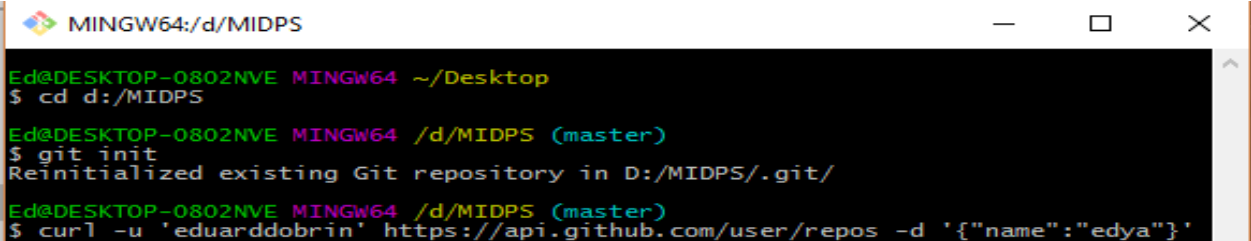
3.1 Cerintele :

- * Initializare unui nou repository.
- * Configurarea VCS.
- * Commit, Push branch.
- * Folosirea fisierului .gitignore.
- * Revenire la versiunile anterioare.
- * Crearea branch-urilor noi.
- * Commit pe ambele branch-uri.
- * Merge la 2 branchuri.
- * Rezolvarea conflictelor.

3.2 Analiza lucrării de laborator :

Linkul la repository <https://github.com/eduarddobrin/MIDPS>
Sunt mai multe modalitati de a initializa un repository pe github. Putem crea o mapa goala in care vom plasa gitul nostru prin intermediul comenzii **git init**.

Urmatorul pas este crearea a noului repository pe care il vom crea utilizand urmatoarea comanda **curl -u 'USER' https://api.github.com/user/repos/ -d '{"name":"NUME"}'**. Unde cuvintele scrise cu CAPS se vor inlocui cu numele utilizatorului si numele repositoryului. Dupa aceasta este necesar sa unim gitul nostru gol cu repositoryul creat. Vom folosi urmatoarea comanda **git remote add origin "Linkul la repositoryul nostru"**



```
MINGW64:/d/MIDPS
Ed@DESKTOP-0802NVE MINGW64 ~/Desktop
$ cd d:/MIDPS
Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ git init
Reinitialized existing Git repository in D:/MIDPS/.git/
Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ curl -u 'eduarddobrin' https://api.github.com/user/repos -d '{"name":"edya"}'
```

Configurarea gitului consta in mai multe etape. La inceput vom configura numele si emailul. Scriem urmatoarele comenzi :

```
git config --global user.name "Numele"
git config --global user.email "Email"
```

MINGW64:/d/MIDPS

```
Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ git config --global user.name "eduarddobrin"

Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ git config --global user.email "dobryy.edya@mail.ru"

Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ git config --list
core.symlinks=false
core.autocrlf=true
core.fscache=true
color.diff=auto
color.status=auto
color.branch=auto
color.interactive=true
help.format=html
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
diff.astextplain.textconv=astextplain
rebase.autosquash=true
credential.helper=manager
user.name=eduarddobrin
user.email=dobryy.edya@mail.ru
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true
remote.MIDPS.url=https://github.com/eduarddobrin/MIDPS.git
remote.MIDPS.fetch=+refs/heads/*:refs/remotes/MIDPS/*
remote.origin.url=https://github.com/eduarddobrin/MIDPS.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*

Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ |
```

Urmatorul pas consta in generarea la cheis **SSH**. Scriem in CLI **ssh-keygen**, iar cheia obtinuta o copiem in setarile noastre de pe github.com.

Este de dorit sa initializam repozitorul nostru cu un fisier **README.md** si un **.gitignore**. In fisierul README.md vom adauga niste informatie pentru cei care se vor folosi de repozitoriu iar in fisierul .gitignore vom adauga toate fisierele ce trebuiesc ignorate (adica sa nu fie incarcate).

MINGW64:/d/MIDPS

```
Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ vim README.md

Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ vim .gitignore

Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ cat README.md

<br>Hello World</br>

Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ vim README.md

Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ cat README.md
MIDPS
Dobrin Eduard. Grupa TI-153

Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ cat .gitignore
ignore.txt

Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ |
```

Vom adauga fisierele noi create pe repozitoriul nostru. Pentru aceasta vom avea nevoie de urmatoarele comenzi :

git add * - comanda indexeaza toate fisierele.

git commit -m "TEXT" – comanda face un snapshot la toate schimbarile noastre.

git push origin master – comanda incarca toate fisierele indexate pe **github.com**

MINGW64:/d/MIDPS

```
Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ git add *

Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ git commit -m "Edya Dobryy"
[master 9898383] Edya Dobryy
3 files changed, 2 insertions(+), 2 deletions(-)
create mode 100644 lab1/Lab1.docx
create mode 100644 lab1/~$Lab1.docx

Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ git push MIDPS master
Counting objects: 6, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 577 bytes | 0 bytes/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To https://github.com/eduarddobrin/MIDPS.git
 b2592aa..9898383 master -> master

Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$
```

Pentru a ne asigura ca am facut totul bine si nu avem probleme vom utiliza :

***git status**

***git show**

```
Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ git status
On branch master
nothing to commit, working tree clean

Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ git show
commit 9898383b2fc23a463283c1f9aab3cb143de515c1
Author: eduarddobrin <dobryy.edya@mail.ru>
Date: Thu Feb 16 22:46:59 2017 +0300

    Edya Dobryy

diff --git a/README.md b/README.md
index f1d95f0..ea73c8d 100644
--- a/README.md
+++ b/README.md
@@ -1,2 +1,2 @@
```

VCS ne permite sa avem mai multe **branchuri**. Din traducere branch semnifica “creanga”. Branchurile sunt foarte comod de folosit cind dorim sa lucram paralel la un proiect si apoi dorim sa unim toate modificarile.

git branch “name” – creeaza un branch nou cu numele “name”.

git branch – vizualizarea branchurilor (* indica branchul curent).

git branch -d “name” – sterge branchul “name”.

git checkout -b “name” - creeaza un branch nou cu numele “name” si face switch la el.

```

MINGW64:/d/MIDPS
Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ git branch Oups
Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ git branch
Oups
* master
Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ git branch -d Oups
Deleted branch Oups (was 9898383).
Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ git checkout -b new
Switched to a new branch 'new'
Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (new)
$ git branch
master
* new
Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (new)
$ ls
lab1/  README.md
Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (new)
$ git add *
Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (new)
$ git commit -m "new branch"
On branch new
nothing to commit, working tree clean
Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (new)
$ git push MIDPS new
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/eduarddobrin/MIDPS.git
 * [new branch]      new -> new
Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (new)
$

```

git checkout “name” – face switch la branchul “name”.

git branch –u upstream/name – face track la branchul indicat din branchul curent.

git branch –u upstream/name “name” – face track din branchul “name” la branchul indicat.

git branch –track “name” upstream/name – creeaza branchul “name” si ii face track la branchul indicat.

git branch –unset-upstream – scoate trackingul la branchul in care ne aflam.

```

MINGW64:/d/MIDPS
Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ git checkout new
Branch new set up to track remote branch new from MIDPS.
Switched to a new branch 'new'
Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (new)
$ git branch -u MIDPS/master
Branch new set up to track remote branch master from MIDPS.
Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (new)
$ git branch -u MIDPS/master new
Branch new set up to track remote branch master from MIDPS.
Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (new)
$ git branch --track "new2" MIDPS/master
Branch new2 set up to track remote branch master from MIDPS.
Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (new)
$ git branch
master
* new
  new2
Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (new)
$ git checkout master
Switched to branch 'master'
Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ git checkout new
Your branch is up-to-date with 'MIDPS/master'.
Switched to branch 'new'
Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (new)
$ git checkout new2
Your branch is up-to-date with 'MIDPS/master'.
Switched to branch 'new2'
Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (new2)
$ git checkout master
Switched to branch 'master'
Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ |

```

Putem avea conflicte in cazul cind dorim sa facem merge la 2 branchuri si unele rinduri sunt diferite. In asa caz ne vin in ajutor mergetool. Drept mergetool am ales **kdiff3**. Pentru kdiff3 ca mergetool default folosim comanda : **git config –global merge.tool kdiff3**

In continuare vom lucra cu 2 branchuri – “master” si “new”. Vom crea in fiecare branch cite un fisier “to_merge” continutul caruia va fi diferit.

```
MINGW64:/d/MIDPS

Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ git branch
* master
  new
  new2

Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ vim to_merge

Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ cat to_merge
edya
exemplu

Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ git checkout new
switched to branch 'new'

Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (new)
$ vim to_merge

Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (new)
$ cat to_merge
edya
exemplu 2

Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (new)
$ git checkout master
switched to branch 'master'
```

In continuare vom incerca sa facem merge si sa rezolvam acest conflict.

```
Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ git branch
* master
  new
  new2

Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ git merge new
Already up-to-date.

Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ git mergetool
```

Dupa acest pas rezovam conflictul co ajutorul **kdifff3**. De exemplu eu am ales sa fac merge in felul urmator.

```
MINGW64:/d/MIDPS

Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ ls
lab1/  README.md  to_merge

Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ cat to_merge
edya
exemplu 2

Ed@DESKTOP-0802NVE MINGW64 /d/MIDPS (master)
$ |
```

4. Concluzie

In lucrarea nr.1 la MIDPS am studiat lucrul cu **VCS**. Am cunoscut platforma **github**. Toate lucrurile, comenzile le-am indeplinit in terminal pe Windows. Sunt o multime de plusuri in folosirea VCS. Fara VCS elaborarea produselor soft ar fi foarte lenta si problematica. El ne permite lucrul paralel, menajarea versiunelor, revenire la versiuni anterioare. In lucrare am practicat majoritatea comenzilor esentiale. Este prima mea experienta cu github.com si mi-am imbunatatit nespuse de mult lucrul pe aceasta platforma. Am cunoscut branchurile, merge la branchuri si rezolvarea conflictelor. Dupa parerea mea orice programator contemporan necesita cunostinta unui VCS. El contribuie nu doar la dezvoltarea hard-skillurilor dar si a celor soft.

5. Referinte :

1. https://github.com/BestMujik/MIDPS-labs/blob/master/MIDPS_LAB%201.md
2. <https://github.com/Ernest96/MIDPS/blob/master/LAB1/Lab%201.pdf>