

**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***

***Efectuat : TI-151 Poseletchi Cristian***

***Verificat : lector asistent Cojanu Irina***

**2017**

# 1. Scopul lucrarii de laborator :

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

## 2. Obiective

Studierea Version Control Systems (git).

Intelegerea si aplicarea comenzilor GIT.

## 3. Mersul lucrarii de laborator

### 3.1 Cerintele :

- \* Initializare unui nou repositoryu.
- \* Configurarea VCS.
- \* Crearea branch-urilor si commit pe ambele branch-uri
- \* Resetarea branch-urilor la commit-urile anterioare
- \* Merge la 2 branchuri.
- \* Folosirea fisierului .gitignore..
- \* Rezolvarea conflictelor.

### 3.2 Analiza lucrarii de laborator :

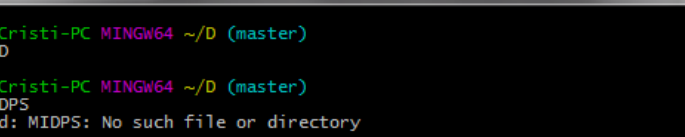
Linkul repositoryului <https://github.com/megustador/MIDPS>

**Configurarea gitului** consta in mai multe etape. La inceput vom configura numele si emailul prin intermediul urmatoarelor comenzi :

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

```
Cristi@Cristi-PC MINGW64 ~ (master)
$ git config --global user.name "megustador"
Cristi@Cristi-PC MINGW64 ~ (master)
$ git config --global user.email "cristiposeletchi@gmail.com"
>
Cristi@Cristi-PC MINGW64 ~ (master)
$ |
```

Exista mai multe metode de a crea un repozitoriu. Eu am creat repozitoriul direct pe github apoi cu **comanda git clone repo\_url** si **SSH adresa** la repo mi-am creat o copie a repozitoriului pe local. Se putea de facut acest lucru si cu comanda **git init**



```
MINGW64:c:/Users/Cristi/D

Cristi@Cristi-PC MINGW64 ~\D (master)
$ cd ~\D

Cristi@Cristi-PC MINGW64 ~\D (master)
$ cd MIDPS
bash: cd: MIDPS: No such file or directory

Cristi@Cristi-PC MINGW64 ~\D (master)
$ git clone git@github.com:megustador/MIDPS.git
Cloning into 'MIDPS'...
Enter passphrase for key 'c:/Users/Cristi/.ssh/id_rsa':
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.

Cristi@Cristi-PC MINGW64 ~\D (master)
$ |
```

Urmatorul pas consta in generarea **SSH key**. Scriem **ssh-keygen**, iar cheia (publica) obtinuta o copiem in setarile noastre de pe github.com.

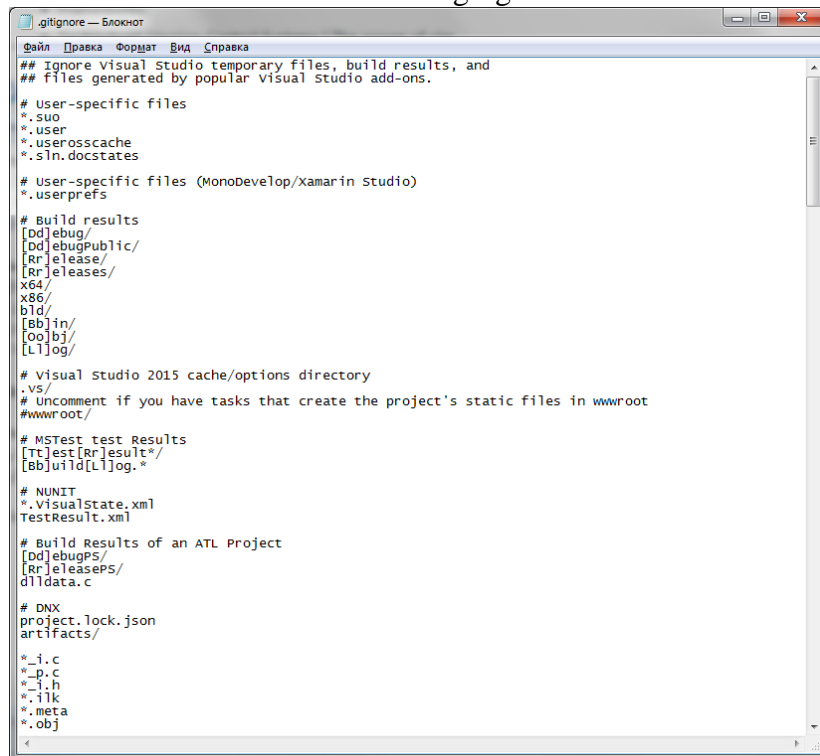
Cum e mentinut si in conditiile laboratorului, este de dorit sa initializam repozitorul nostru cu un fisier **README.md** si un **.gitignore**. In fisierul README.md vom adauga informatii pentru cei care se vor folosi de repozitoriu iar in fisierul .gitignore vom adauga toate fisierele ce trebuiesc ignorate (adica sa nu fie incarcate la moment ).

```

MINGW64:/c:/Users/Cristi
Enter file in which to save the key (/c:/Users/Cristi/.ssh/id_rsa): midps
midps already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in midps.
Your public key has been saved in midps.pub.
The key fingerprint is:
SHA256:kDdpkAkXpmZc5oyAVp5PuR8hpmvF6jssOdtKharcPvI Cristi@Cristi-PC
The key's randomart image is:
+---[RSA 2048]-----+
|          ..          |
|       .   ...o       |
|    .o ..+o=          |
| ..oo.=*=.   .        |
|..  +*Xo S           |
|..  *=o.           |
|..oo+ . .          |
|+=oB .             |
|. +@E+             |
+---[SHA256]-----+
Cristi@Cristi-PC MINGW64 ~ (master)
$

```

## Fisierul gitignore



```
.gitignore — Блокнот
Файл  Правка  Формат  Вид  Справка

## Ignore Visual Studio temporary files, build results, and
## files generated by popular Visual Studio add-ons.

# User-specific files
*.suo
*.user
*.useroscachе
*.sln.docstates

# User-specific files (MonoDevelop/Xamarin Studio)
*.userprefs

# Build results
[dd]ebug/
[dd]ebugPublic/
[Rr]elease/
[Rr]eleases/
x64/
x86/
bld/
[bb]in/
[oo]bj/
[Ll]og/

# Visual Studio 2015 cache/options directory
.vs/
# Uncomment if you have tasks that create the project's static files in wwwroot
#wwwroot/

# MSTest test Results
[tt]est[Rr]esult*/
[bb]uild[Ll]og.*

# NUNIT
*.visualState.xml
TestResult.xml

# Build Results of an ATL Project
[dd]ebugPS/
[Rr]eleasePS/
dlldata.c

# DNX
project.lock.json
artifacts/

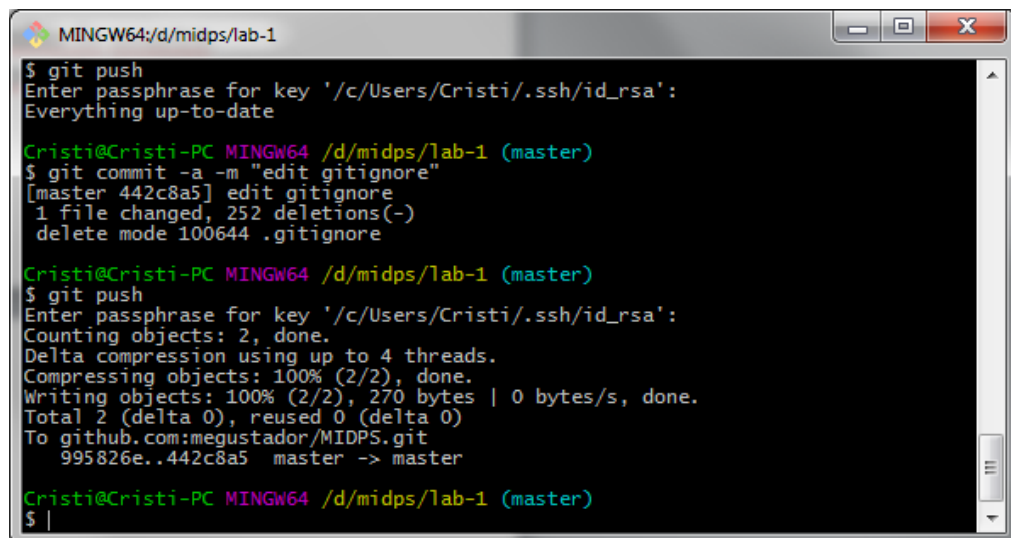
*.i.c
*.p.c
*.i.h
*.ilk
*.meta
*.obj
```

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/lab-1
$ git push
Enter passphrase for key '/c/Users/Cristi/.ssh/id_rsa':
Everything up-to-date

Cristi@Cristi-PC MINGW64 /d/midps/lab-1 (master)
$ git commit -a -m "edit gitignore"
[master 442c8a5] edit gitignore
1 file changed, 252 deletions(-)
delete mode 100644 .gitignore

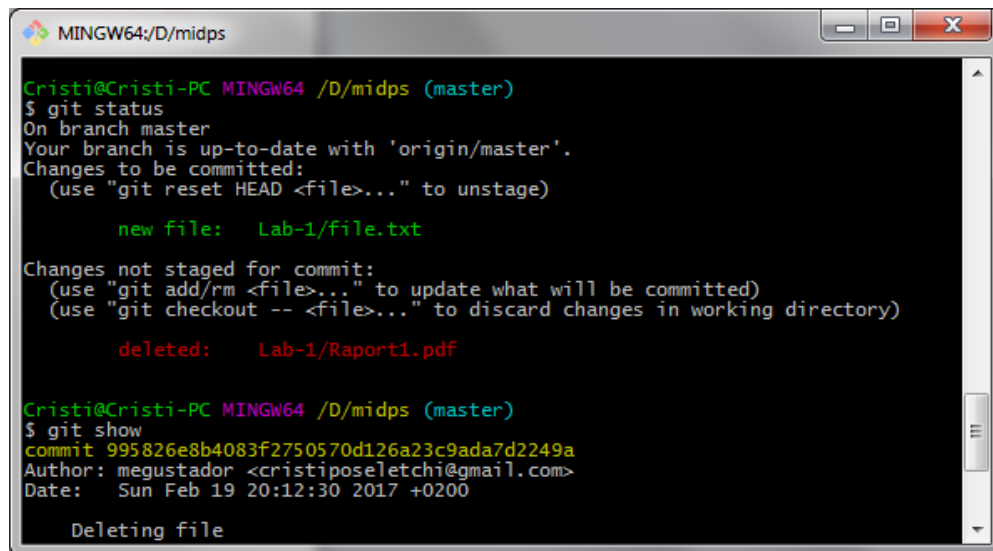
Cristi@Cristi-PC MINGW64 /d/midps/lab-1 (master)
$ git push
Enter passphrase for key '/c/Users/Cristi/.ssh/id_rsa':
Counting objects: 2, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 270 bytes | 0 bytes/s, done.
Total 2 (delta 0), reused 0 (delta 0)
To github.com:megustador/MIDPS.git
995826e..442c8a5  master -> master

Cristi@Cristi-PC MINGW64 /d/midps/lab-1 (master)
$
```

Pentru a ne asigura ca am facut totul bine si nu avem probleme utilizam urmatoarele comenzi git:

\***git status**

\***git show**



```
MINGW64:/D/midps

Cristi@Cristi-PC MINGW64 /D/midps (master)
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   Lab-1/file.txt

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    deleted:    Lab-1/Raport1.pdf

Cristi@Cristi-PC MINGW64 /D/midps (master)
$ git show
commit 995826e8b4083f2750570d126a23c9ada7d2249a
Author: megustador <crisposeletchi@gmail.com>
Date:   Sun Feb 19 20:12:30 2017 +0200

    Deleting file
```

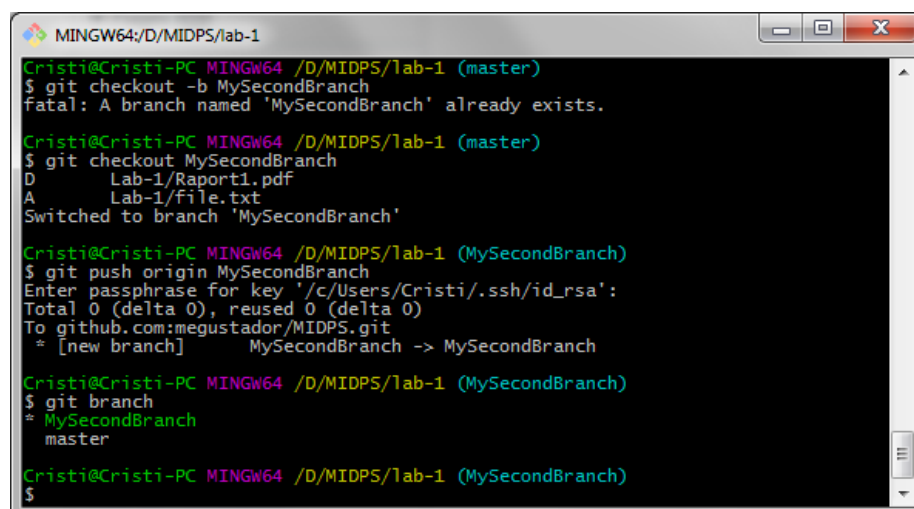
VCS ne permite sa avem mai multe **branchuri**. Din ENG branch semnifica “creanga”. Branch-urile sunt utilizate cind lucram paralel la un proiect si apoi dorim sa combinam toate modificarile.

**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/lab-1

Cristi@Cristi-PC MINGW64 /D/MIDPS/lab-1 (master)
$ git checkout -b MySecondBranch
fatal: A branch named 'MySecondBranch' already exists.

Cristi@Cristi-PC MINGW64 /D/MIDPS/lab-1 (master)
$ git checkout MySecondBranch
D    Lab-1/Raport1.pdf
A    Lab-1/file.txt
Switched to branch 'MySecondBranch'

Cristi@Cristi-PC MINGW64 /D/MIDPS/lab-1 (MySecondBranch)
$ git push origin MySecondBranch
Enter passphrase for key '/c/Users/Cristi/.ssh/id_rsa':
Total 0 (delta 0), reused 0 (delta 0)
To github.com:megustador/MIDPS.git
 * [new branch]      MySecondBranch -> MySecondBranch

Cristi@Cristi-PC MINGW64 /D/MIDPS/lab-1 (MySecondBranch)
$ git branch
* MySecondBranch
  master

Cristi@Cristi-PC MINGW64 /D/MIDPS/lab-1 (MySecondBranch)
$
```

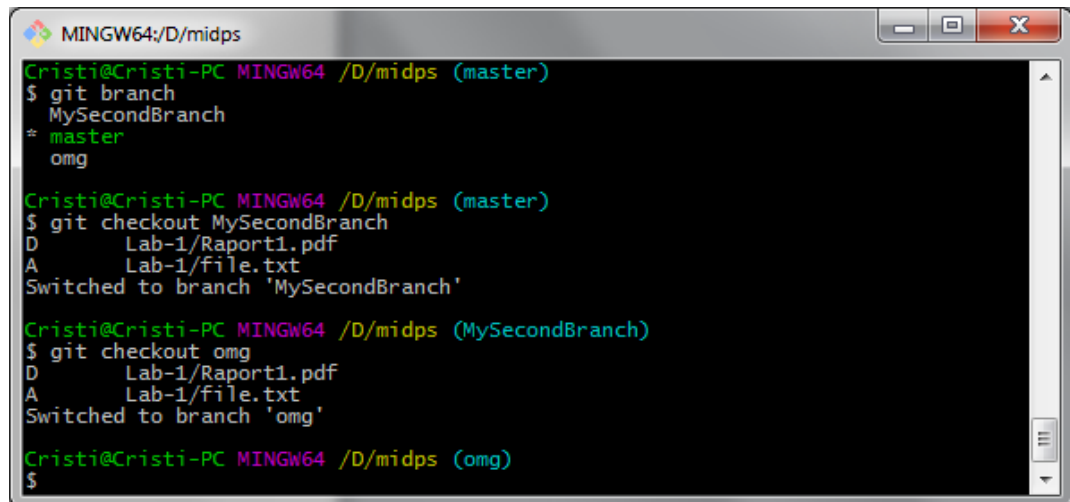
**git checkout “name”** – face switch la branchul “name”.

**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.

A terminal window titled 'MINGW64:/D/midps' showing a series of git commands. The user starts on the 'master' branch, lists branches (showing 'MySecondBranch' and 'omg'), then checks out 'MySecondBranch'. From there, they check out 'omg'.

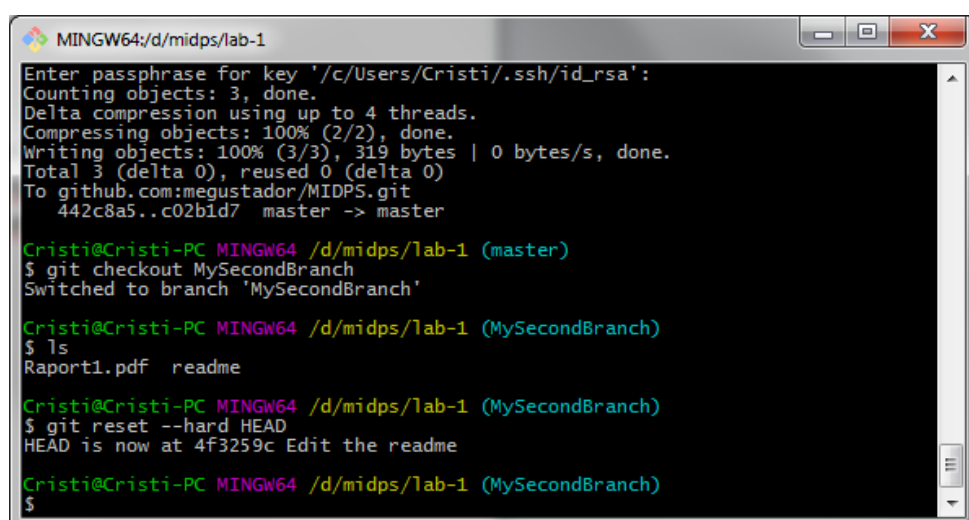
```
Cristi@Cristi-PC MINGW64 /D/midps (master)
$ git branch
  MySecondBranch
* master
  omg

Cristi@Cristi-PC MINGW64 /D/midps (master)
$ git checkout MySecondBranch
D       Lab-1/Raport1.pdf
A       Lab-1/file.txt
Switched to branch 'MySecondBranch'

Cristi@Cristi-PC MINGW64 /D/midps (MySecondBranch)
$ git checkout omg
D       Lab-1/Raport1.pdf
A       Lab-1/file.txt
Switched to branch 'omg'

Cristi@Cristi-PC MINGW64 /D/midps (omg)
$
```

In caz ca dorim sa schimbam istoria unui commit, sau sa **resetam un branch la commitul anterior**. Pentru asta putem folosi comanda **git reset commit\_index**. Pentru a demonstra asta am ales branchul “MySecondBranch” de pe repo-ul meu si l-am resetat la ultimul commit facut.

A terminal window titled 'MINGW64:/d/midps/lab-1' showing a git push followed by a checkout to 'MySecondBranch', listing files, and then performing a hard reset to HEAD. The terminal output shows the push details and the state of the branch after the reset.

```
MINGW64:/d/midps/lab-1
Enter passphrase for key '/c/Users/Cristi/.ssh/id_rsa':
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 319 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:megustador/MIDPS.git
   442c8a5..c02b1d7  master -> master

Cristi@Cristi-PC MINGW64 /d/midps/lab-1 (master)
$ git checkout MySecondBranch
Switched to branch 'MySecondBranch'

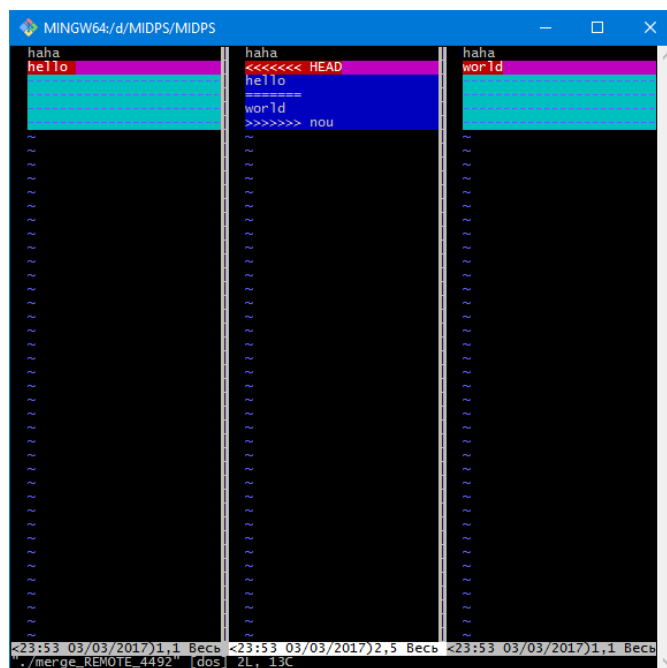
Cristi@Cristi-PC MINGW64 /d/midps/lab-1 (MySecondBranch)
$ ls
Raport1.pdf  readme

Cristi@Cristi-PC MINGW64 /d/midps/lab-1 (MySecondBranch)
$ git reset --hard HEAD
HEAD is now at 4f3259c Edit the readme

Cristi@Cristi-PC MINGW64 /d/midps/lab-1 (MySecondBranch)
$
```

Pot aparea conflicte in cazul cind dorim sa facem **merge** la 2 branch-uri si unele rinduri sunt diferite. In asa caz, pentru a elimina conflictele, folosim **mergetool**. Drept mergetool am ales **kdiff3**. Pentru kdiff3, in mod implicit folosim comanda : **git config --global merge.tool kdiff3**.

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



In continuare facem merge si incercam sa rezolvam acest conflict cu ajutorul **kdiff3**.

```
MINGW64:/d/MIDPS/MIDPS
Your branch is up-to-date with 'origin/master'.
Switched to branch 'master'

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS/MIDPS (master)
$ ls
lab1/ lab2/ lab3/ lab4/ lab5/ merge README.md to_merge

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS/MIDPS (master)
$ git merge nou
Auto-merging merge
CONFLICT (add/add): Merge conflict in merge
Automatic merge failed; fix conflicts and then commit the result.

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS/MIDPS (master|MERGING)
$ git mergetool

This message is displayed because 'merge.tool' is not configured.
See 'git mergetool --tool-help' or 'git help config' for more details.
'git mergetool' will now attempt to use one of the following tools:
opendiff kdiff3 tkdiff xxdiff meld tortoisemerge gvimdiff diffuse diffmerge ecme
rge p4merge araxis bc codecompare emerge vimdiff
Merging:
merge

Normal merge conflict for 'merge':
{local}: created file
{remote}: created file
Hit return to start merge resolution tool (vimdiff):
Файлов для редактирования: 3

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS/MIDPS (master|MERGING)
$ git add *

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS/MIDPS (master|MERGING)
$ git commit -m "hello"
[master 0abbf38] hello
```

## 4. Concluzie

In lucrarea nr.1 la MIDPS am studiat lucrul cu VCS. Am facut cunostinta cu o alta fata a **github-ului** (eu il facusem doar pentru freecodecamp, fara a-i studia functiile). Toate comenzile git le-am indeplinit in terminal pe Windows. Fara VCS elaborarea si partajarea produselor soft ar fi foarte lenta si problematica. El ne permite lucrul paralel intre proiecte, menajarea versiunelor, revenire la versiuni anterioare. In lucrare am practicat majoritatea comenzilor esentiale. Companiile de success recomanda cunoasterea unui VCS, deci e si el parte din cheia succesului. El contribuie nu doar la dezvoltarea hard si soft-skillurilor dar si personala, in cazul meu. : In sfirsit am invatat sa folosesc Google, cautind multimea notiunilor si comenzilor mai variate decit cele de baza, acesta fiind si motivul pentru care am intirziat, dar cred ca e mult mai bine in comparatie cu devenirea unui (citez) "*bîdlo-coder*".

## 5. Referinte :

1. [https://github.com/BestMujik/MIDPS-labs/blob/master/MIDPS\\_LAB%231.md](https://github.com/BestMujik/MIDPS-labs/blob/master/MIDPS_LAB%231.md)
2. <https://github.com/Ernest96/MIDPS/blob/master/LAB1/Lab%231.pdf>
3. <https://www.atlassian.com/git/tutorials/>
4. <http://acs.ase.ro/Media/Default/documents/cts/SeminarZamfiroiu/GIT.pdf>
5. <https://www.google.com/> //recomand ^\_^