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 : **Tabuncic Vladislav**

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 repositoriu.

* 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..

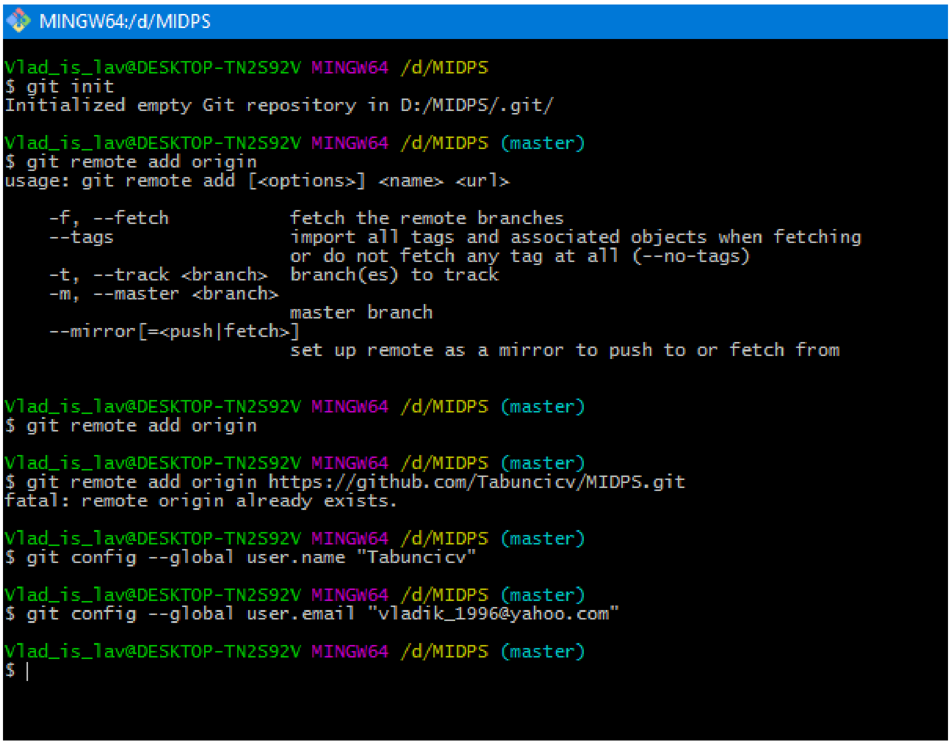
* Rezolvarea conflictelor.

3.2 Analiza lucrarii de laborator :

Linkul repositoriului https://github.com/Tabuncicv/MIDPS.git

Sunt mai multe modalitati de a initializa un repozitoriu pe github. Putem crea o mapa goala in care vom plasa gitul nostru prin intermediul comenzii **git init.**

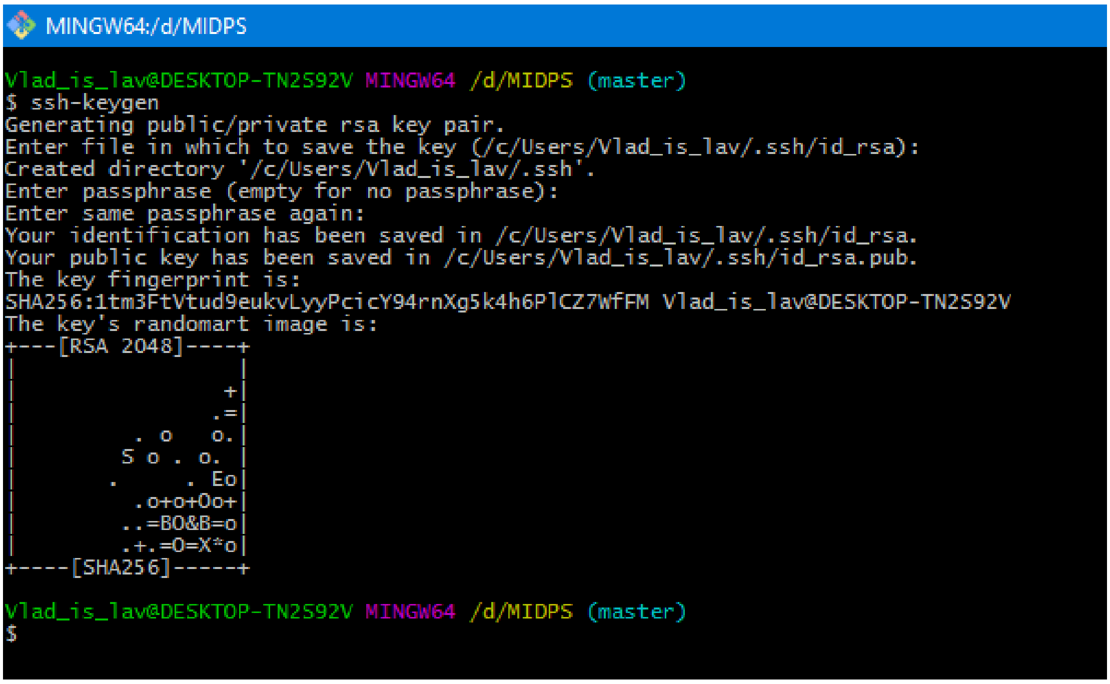
Urmatorul pas este crearea a noului repozitoriu pe care il vom crea utilizind urmatoarea comanda **curl – u 'USER' https.//api.github.com/user/repos/ -d '{"name":"NUME"}'.** Unde cuvintele scrise cu CAPS se vor inlocui cu numele utilizatorului si numele repozitoruui. Dupa aceasta este necesar sa unim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



```
Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS
$ git init
Initialized empty Git repository in D:/MIDPS/.git/

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS (master)
$ git remote add origin
usage: git remote add [<options>] <name> <url>

    -f, --fetch           fetch the remote branches
    --tags                import all tags and associated objects when fetching
                          or do not fetch any tag at all (--no-tags)
    -t, --track <branch>  branch(es) to track
    -m, --master <branch>
                          master branch
    --mirror[=<push|fetch>]
                          set up remote as a mirror to push to or fetch from


Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS (master)
$ git remote add origin

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS (master)
$ git remote add origin https://github.com/Tabuncicv/MIDPS.git
fatal: remote origin already exists.

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS (master)
$ git config --global user.name "Tabuncicv"

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS (master)
$ git config --global user.email "vladik_1996@yahoo.com"

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS (master)
$ |
```

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

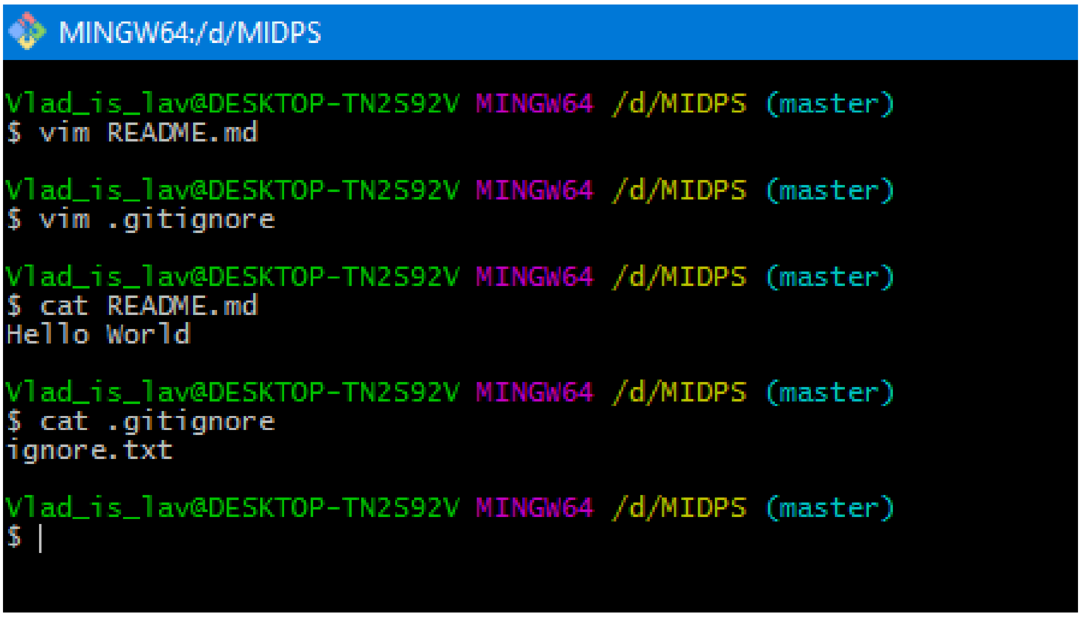
git config –global user.name “Numele”
git config –global user.email “Email”



```
Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS (master)
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Vlad_is_lav/.ssh/id_rsa):
Created directory '/c/Users/Vlad_is_lav/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Vlad_is_lav/.ssh/id_rsa.
Your public key has been saved in /c/Users/Vlad_is_lav/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:1tm3FtVtud9eukvLyyPcicY94rnXg5k4h6PlCZ7WfFM Vlad_is_lav@DESKTOP-TN2S92V
The key's randomart image is:
+---[RSA 2048]----+
|                 |
|                +|
|               .=|
|         . o   o.|
|        S o . o. |
|       .     . Eo|
|         .o+o+Oo+|
|        ..=BO&B=o|
|        .+.=O=X*o|
+----[SHA256]-----+

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS (master)
$
```

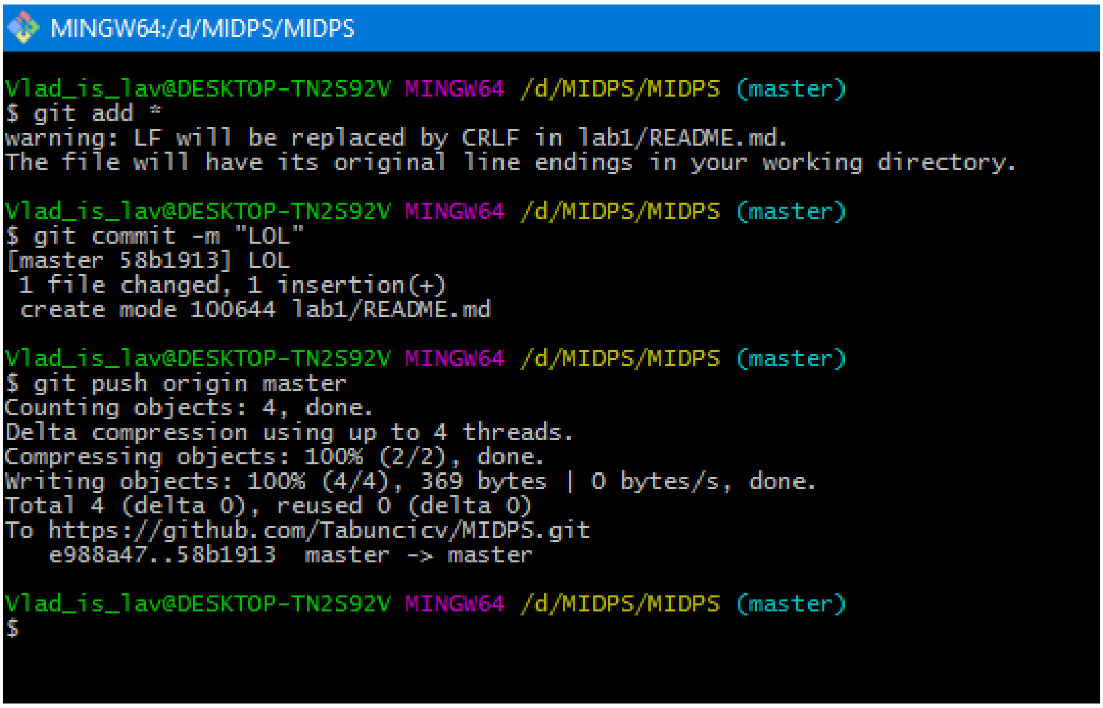
Urmatorul pas consta in generarea **SSH** key**.** Scriem **ssh-keygen,** iar cheia (publica) obtinuta o copiem in setarile noastre de pe github.com.

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).



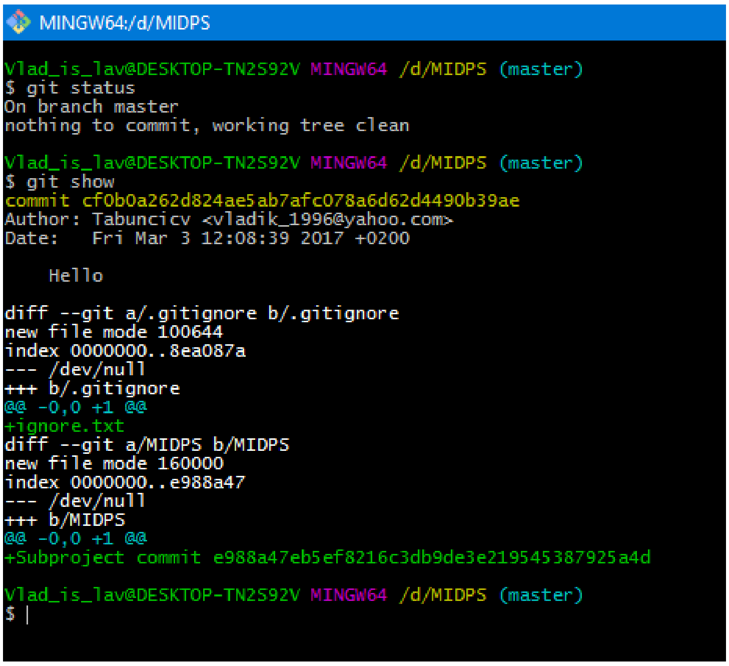
```
Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS (master)
$ vim README.md

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS (master)
$ vim .gitignore

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS (master)
$ cat README.md
Hello World

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS (master)
$ cat .gitignore
ignore.txt

Vlad_is_lav@DESKTOP-TN2S92V 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**



```
Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS/MIDPS (master)
$ git add *
warning: LF will be replaced by CRLF in lab1/README.md.
The file will have its original line endings in your working directory.

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS/MIDPS (master)
$ git commit -m "LOL"
[master 58b1913] LOL
 1 file changed, 1 insertion(+)
 create mode 100644 lab1/README.md

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS/MIDPS (master)
$ git push origin master
Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 369 bytes | 0 bytes/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/Tabuncicv/MIDPS.git
   e988a47..58b1913  master -> master

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS/MIDPS (master)
$
```

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

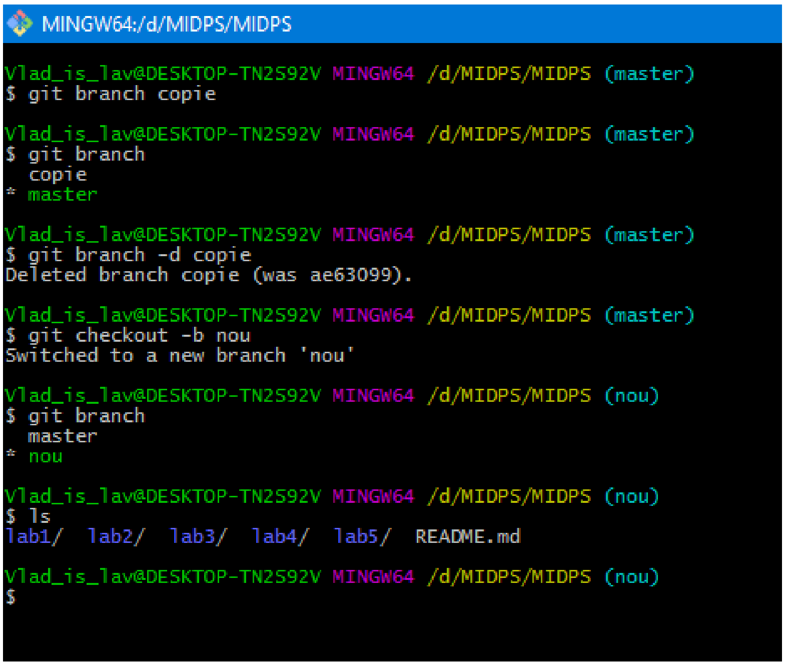


MINGW64:/d/MIDPS

```
Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS (master)
$ git status
On branch master
nothing to commit, working tree clean

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS (master)
$ git show
commit cf0b0a262d824ae5ab7afc078a6d62d4490b39ae
Author: Tabuncicv <vladik_1996@yahoo.com>
Date:   Fri Mar 3 12:08:39 2017 +0200

    Hello

diff --git a/.gitignore b/.gitignore
new file mode 100644
index 0000000..8ea087a
--- /dev/null
+++ b/.gitignore
@@ -0,0 +1 @@
+ignore.txt
diff --git a/MIDPS b/MIDPS
new file mode 160000
index 0000000..e988a47
--- /dev/null
+++ b/MIDPS
@@ -0,0 +1 @@
+Subproject commit e988a47eb5ef8216c3db9de3e219545387925a4d

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS (master)
$ |
```

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.

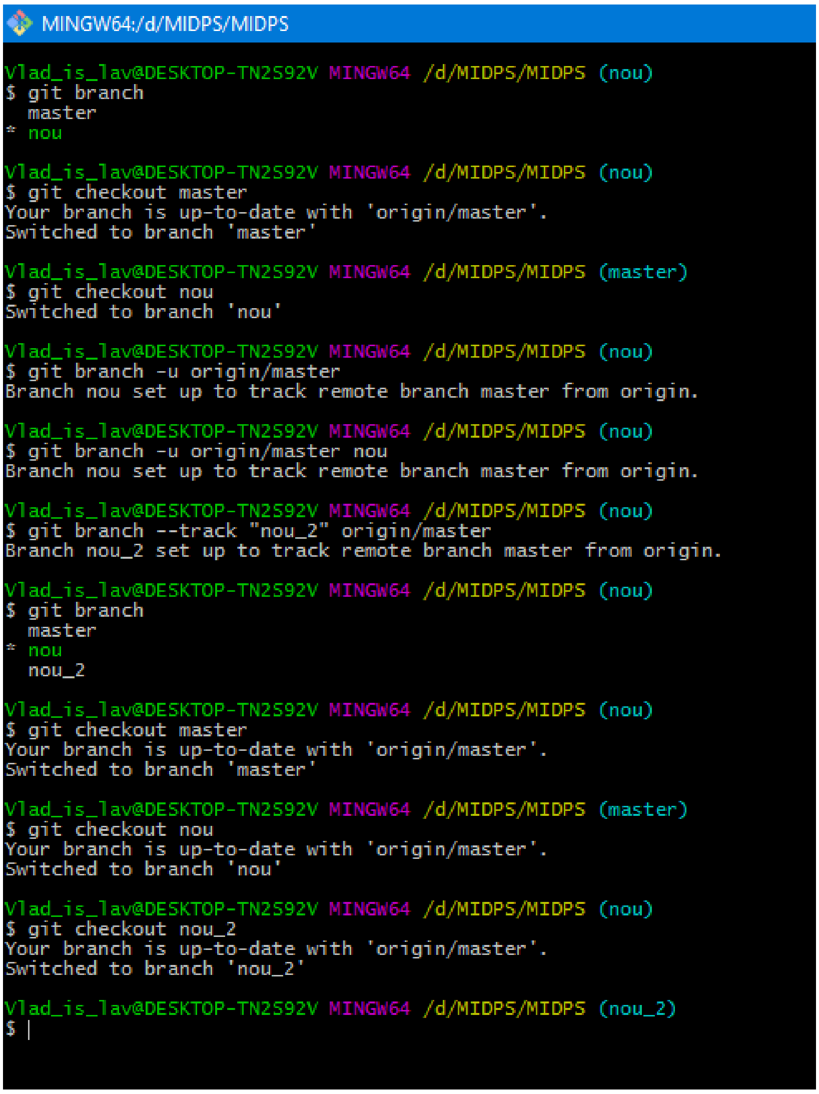
git branch “name” – creeaza un branch nou cu numele “name”.
git branch – vizualizarea branchurilor (* indica branchul curent).
git branch –d “name” – sterge branchul “name”.
git checkout –b “name” - creeaza un branch nou cu numele “name” si face switch la el.



MINGW64:/d/MIDPS/MIDPS

```
Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS/MIDPS (master)
$ git branch copie

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS/MIDPS (master)
$ git branch
  copie
* master

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS/MIDPS (master)
$ git branch -d copie
Deleted branch copie (was ae63099).

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS/MIDPS (master)
$ git checkout -b nou
Switched to a new branch 'nou'

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS/MIDPS (nou)
$ git branch
  master
* nou

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

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS/MIDPS (nou)
$
```

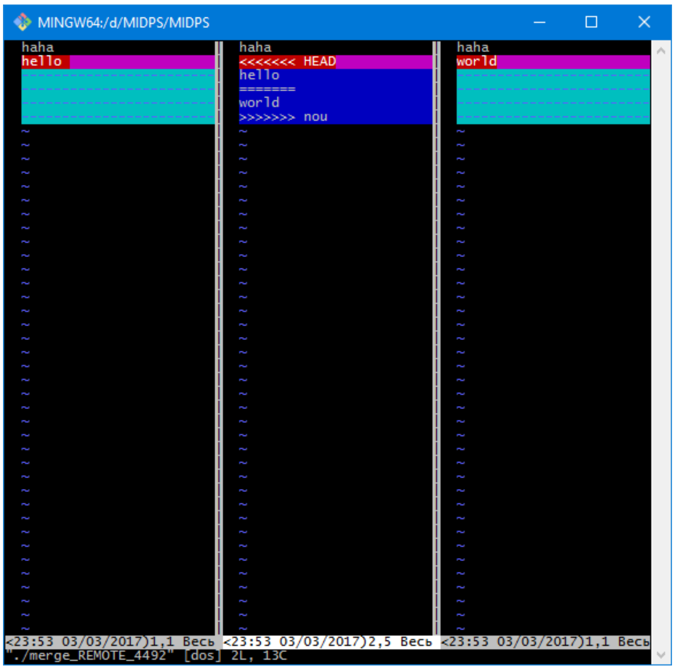
git checkout “name” – face switch la branchul “name”.
git branch –u upstream/name – face track la branchul indicat din branchul curent.
git branch –u upstream/name “name” – face track din branchul “name” la branchul indicat.
git branch –track “name” upstream/name – creeaza branchul “name” si ii face track la branchul indicat.
git branch –unset-upstream – scoate trackingul la branchul in care ne aflam.



```
Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS/MIDPS (nou)
$ git branch
  master
* nou

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS/MIDPS (nou)
$ git checkout master
Your branch is up-to-date with 'origin/master'.
Switched to branch 'master'

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS/MIDPS (master)
$ git checkout nou
Switched to branch 'nou'

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS/MIDPS (nou)
$ git branch -u origin/master
Branch nou set up to track remote branch master from origin.

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS/MIDPS (nou)
$ git branch -u origin/master nou
Branch nou set up to track remote branch master from origin.

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS/MIDPS (nou)
$ git branch --track "nou_2" origin/master
Branch nou_2 set up to track remote branch master from origin.

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS/MIDPS (nou)
$ git branch
  master
* nou
  nou_2

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS/MIDPS (nou)
$ git checkout master
Your branch is up-to-date with 'origin/master'.
Switched to branch 'master'

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS/MIDPS (master)
$ git checkout nou
Your branch is up-to-date with 'origin/master'.
Switched to branch 'nou'

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS/MIDPS (nou)
$ git checkout nou_2
Your branch is up-to-date with 'origin/master'.
Switched to branch 'nou_2'

Vlad_is_lav@DESKTOP-TN2S92V MINGW64 /d/MIDPS/MIDPS (nou_2)
$ |
```

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.

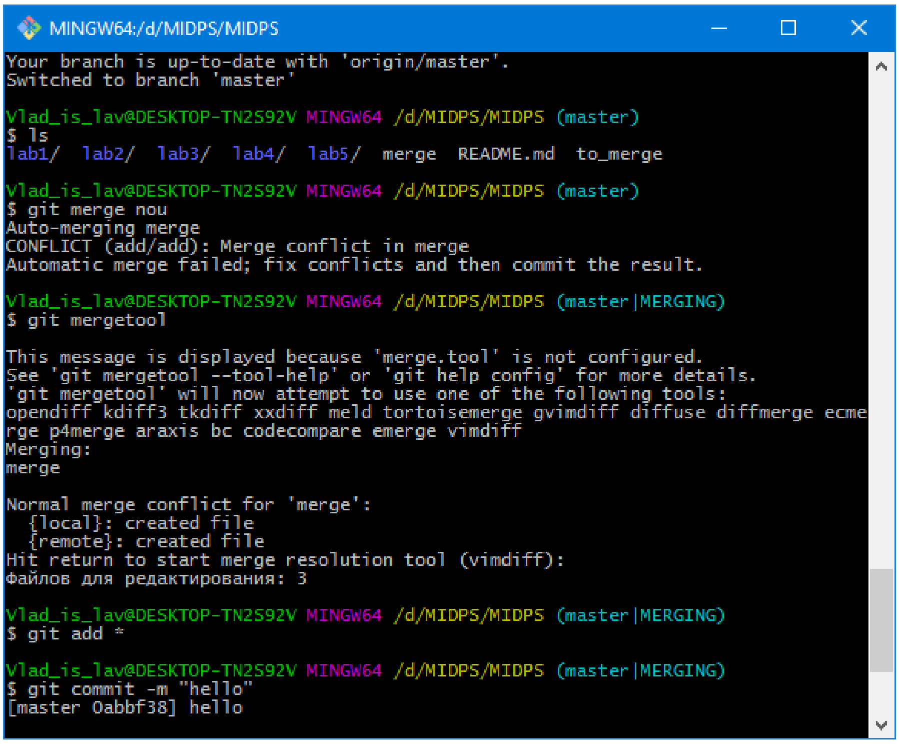


MINGW64:/d/MIDPS/MIDPS

```
haha                       ||haha                       ||haha
hello                      ||<<<<<<< HEAD               ||world
---------------------------||hello                      ||---------------------------
---------------------------||=======                    ||---------------------------
---------------------------||world                      ||---------------------------
---------------------------||>>>>>>> nou                ||---------------------------
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
~                          ||~                          ||~
<23:53 03/03/2017)1,1 Весь <23:53 03/03/2017)2,5 Весь <23:53 03/03/2017)1,1 Весь
"./merge_REMOTE_4492" [dos] 2L, 13C
```

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

Dupa acest pas rezovam conflictul co ajutorul **kdiff3.** De exemplu eu am ales sa fac merge in felul urmator.



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

2. https://github.com/Ernest96/MIDPS/blob/master/LAB1/Lab%231.pdf

3. https://www.atlassian.com/git/tutorials/git-merge

4. http://acs.ase.ro/Media/Default/documents/cts/SeminarZamfiroiu/GIT.pdf

5. https://www.google.com/   //recomand ^_^