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 Moraru Dumitru

A verificat: Cojanu Irina

Chișinău 2017

Obiective:

- Version Control Systems (git)

Cerințele laboratorului:

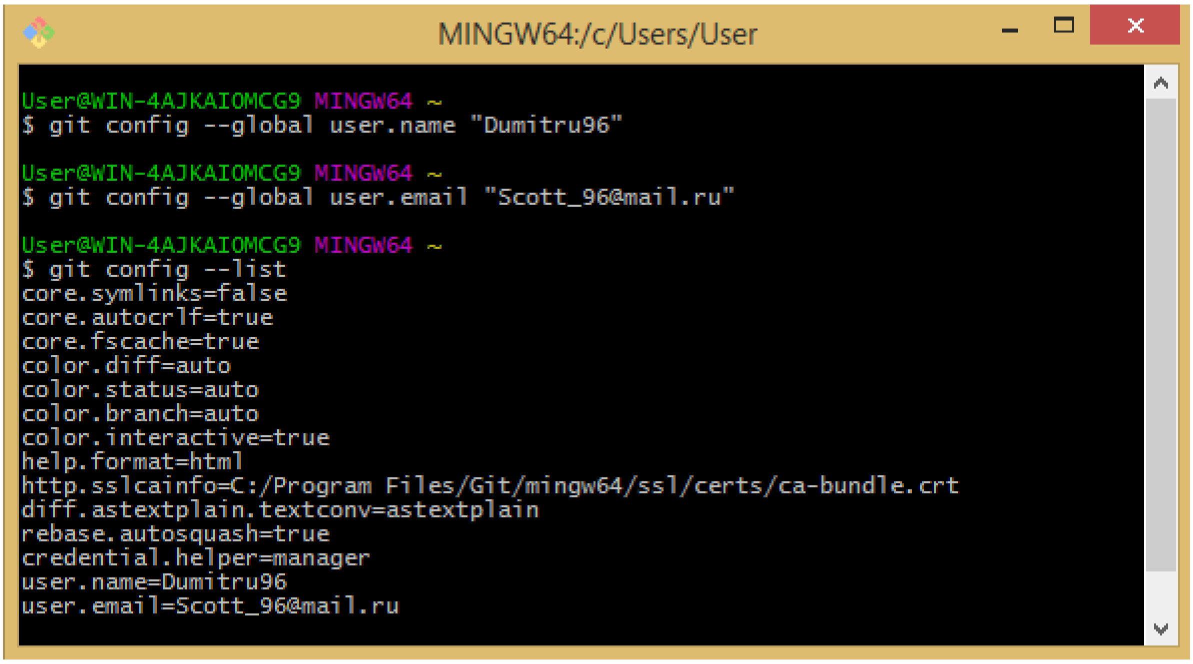
- *Basic Level* :
    - initializeaza un nou repositoriu
    - configureaza-ti VCS
    - crearea branch-urilor (creeaza cel putin 2 branches)
    - commit pe ambele branch-uri (cel putin 1 commit per branch)
- *Normal Level* :
    - seteaza un branch to track a remote origin pe care vei putea sa faci 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
    - comezile git care trebuie cunoscute

Analiza Lucrării de laborator:

Link-ul la repozitoriu **https://github.com/DocHumanity/MIDPS**

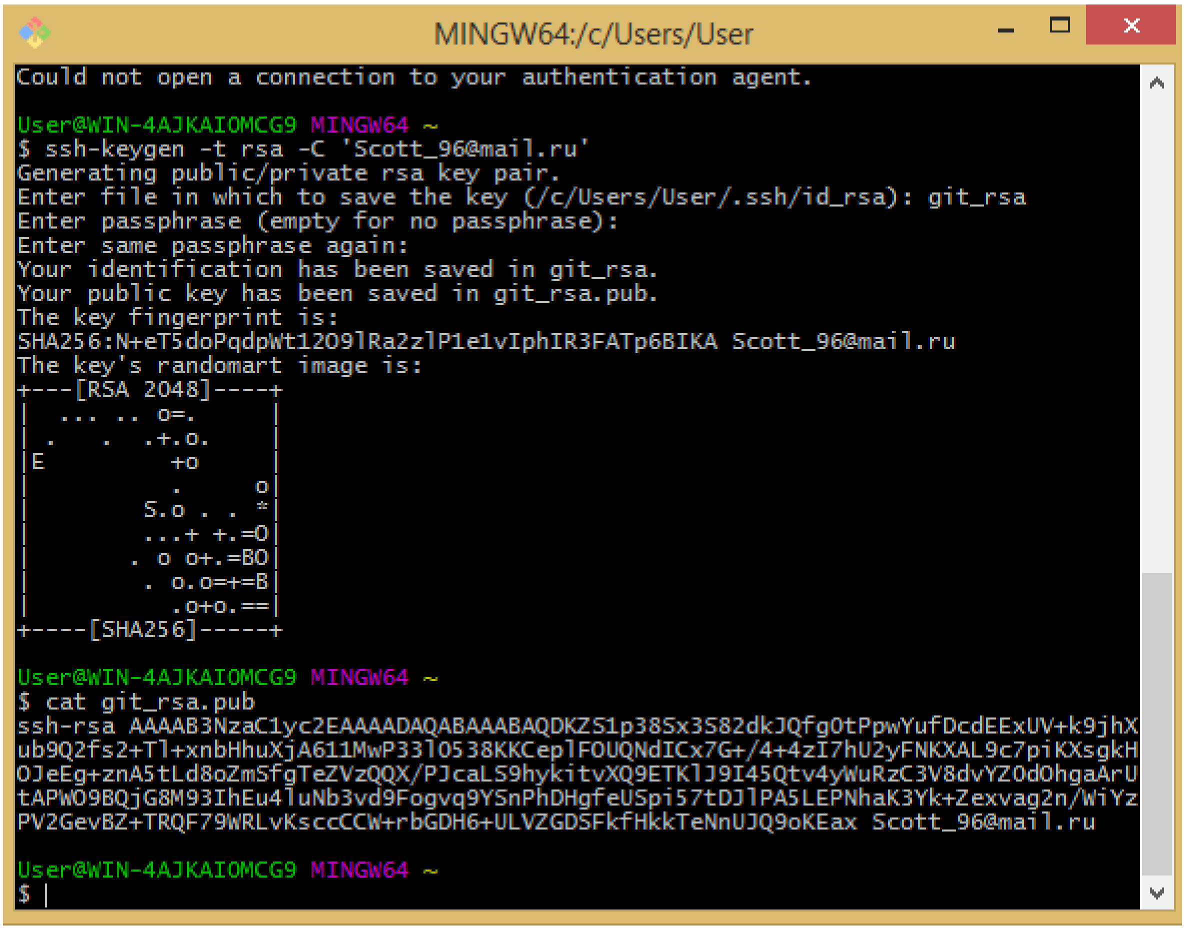
Am creat repozitoriul prin metoda online.Am deschis pagina mea pe github.com,click pe Repositories și apoi pe butonul New.Atunci când am creat repozitoriul MIDPS,l-am inițializat cu un fișier README.

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”.**

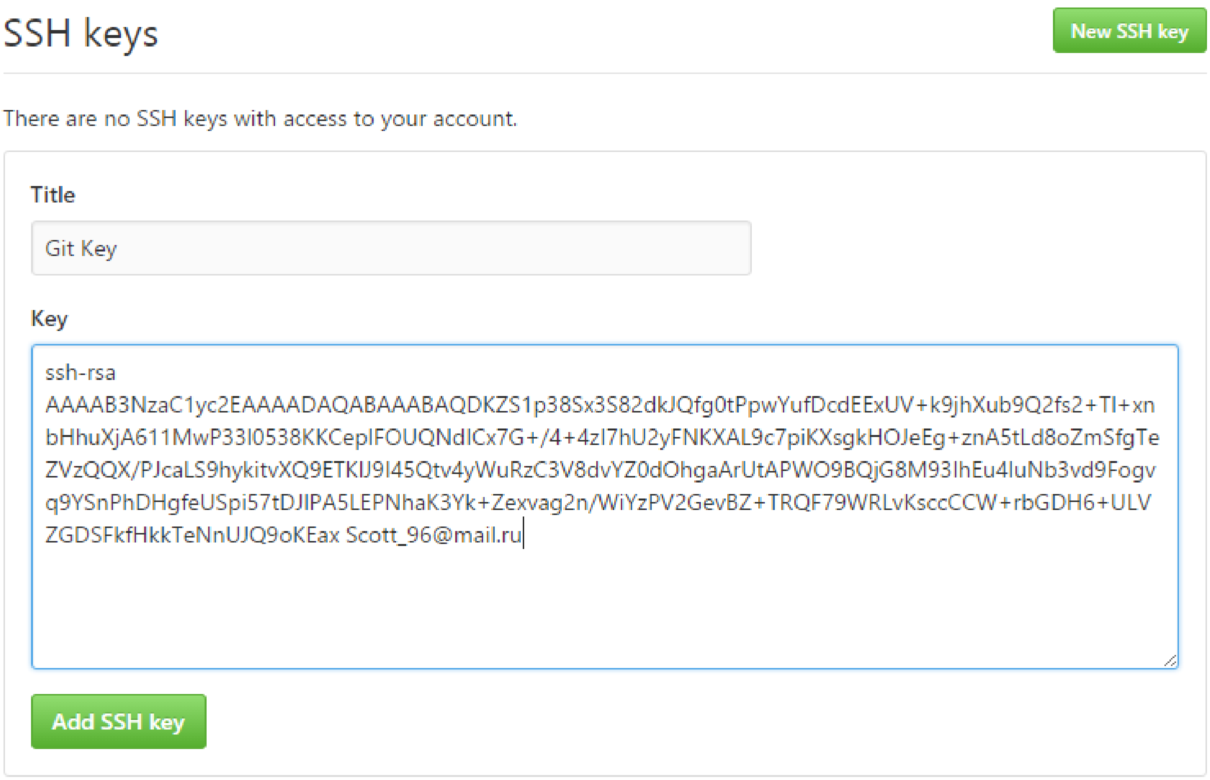


```
User@WIN-4AJKAIOMCG9 MINGW64 ~
$ git config --global user.name "Dumitru96"

User@WIN-4AJKAIOMCG9 MINGW64 ~
$ git config --global user.email "Scott_96@mail.ru"

User@WIN-4AJKAIOMCG9 MINGW64 ~
$ 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=Dumitru96
user.email=Scott_96@mail.ru
```

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



```
Could not open a connection to your authentication agent.

User@WIN-4AJKAIOMCG9 MINGW64 ~
$ ssh-keygen -t rsa -C 'Scott_96@mail.ru'
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/User/.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:N+eT5doPqdpWt12O9lRa2zlP1e1vIphIR3FATp6BIKA Scott_96@mail.ru
The key's randomart image is:
+---[RSA 2048]----+
|  ... .. o=.     |
| .   .  .+.o.    |
|E         +o     |
|          .     o|
|        S.o . . *|
|        ...+ +.=O|
|       . o o+.=BO|
|        . o.o=+=B|
|          .o+o.==|
+----[SHA256]-----+

User@WIN-4AJKAIOMCG9 MINGW64 ~
$ cat git_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDKZS1p38Sx3582dkJQfg0tPpwYufDcdEExUV+k9jhX
ub9Q2fs2+Tl+xnbHhuXjA611MwP33l0538KKCeplFOUQNdICx7G+/4+4zI7hU2yFNKXAL9c7piKXsgkH
OJeEg+znA5tLd8oZmSfgTeZVzQQX/PJcaLS9hykitvXQ9ETKlJ9I45Qtv4yWuRzC3V8dvYZOdOhgaArU
tAPWO9BQjG8M93IhEu4luNb3vd9Fogvq9YSnPhDHgfeUSpi57tDJlPA5LEPNhaK3Yk+Zexvag2n/WiYz
PV2GevBZ+TRQF79WRLvKsccCCW+rbGDH6+ULVZGDSFkfHkkTeNnUJQ9oKEax Scott_96@mail.ru

User@WIN-4AJKAIOMCG9 MINGW64 ~
$ |
```



There are no SSH keys with access to your account.

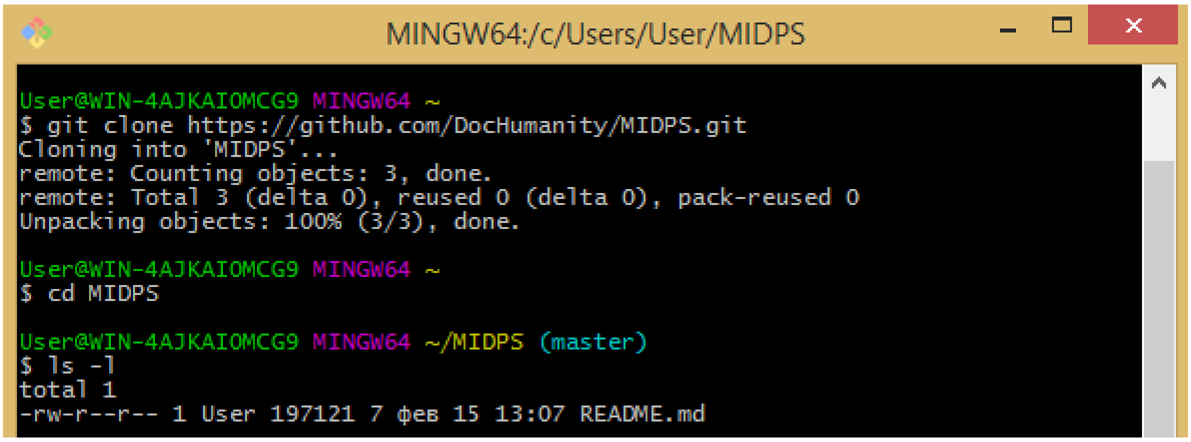
Title

Git Key

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDKZS1p38Sx3S82dkJQfg0tPpwYufDcdEExUV+k9jhXub9Q2fs2+Tl+xn
bHhuXjA611MwP33l0538KKCeplFOUQNdlCx7G+/4+4zI7hU2yFNKXAL9c7piKXsgkHOJeEg+znA5tLd8oZmSfgTe
ZVzQQX/PJcaLS9hykitvXQ9ETKIJ9I45Qtv4yWuRzC3V8dvYZ0dOhgaArUtAPWO9BQjG8M93IhEu4luNb3vd9Fogv
q9YSnPhDHgfeUSpi57tDJIPA5LEPNhaK3Yk+Zexvag2n/WiYzPV2GevBZ+TRQF79WRLvKsccCCW+rbGDH6+ULV
ZGDSFkfHkkTeNnUJQ9oKEax Scott_96@mail.ru
```

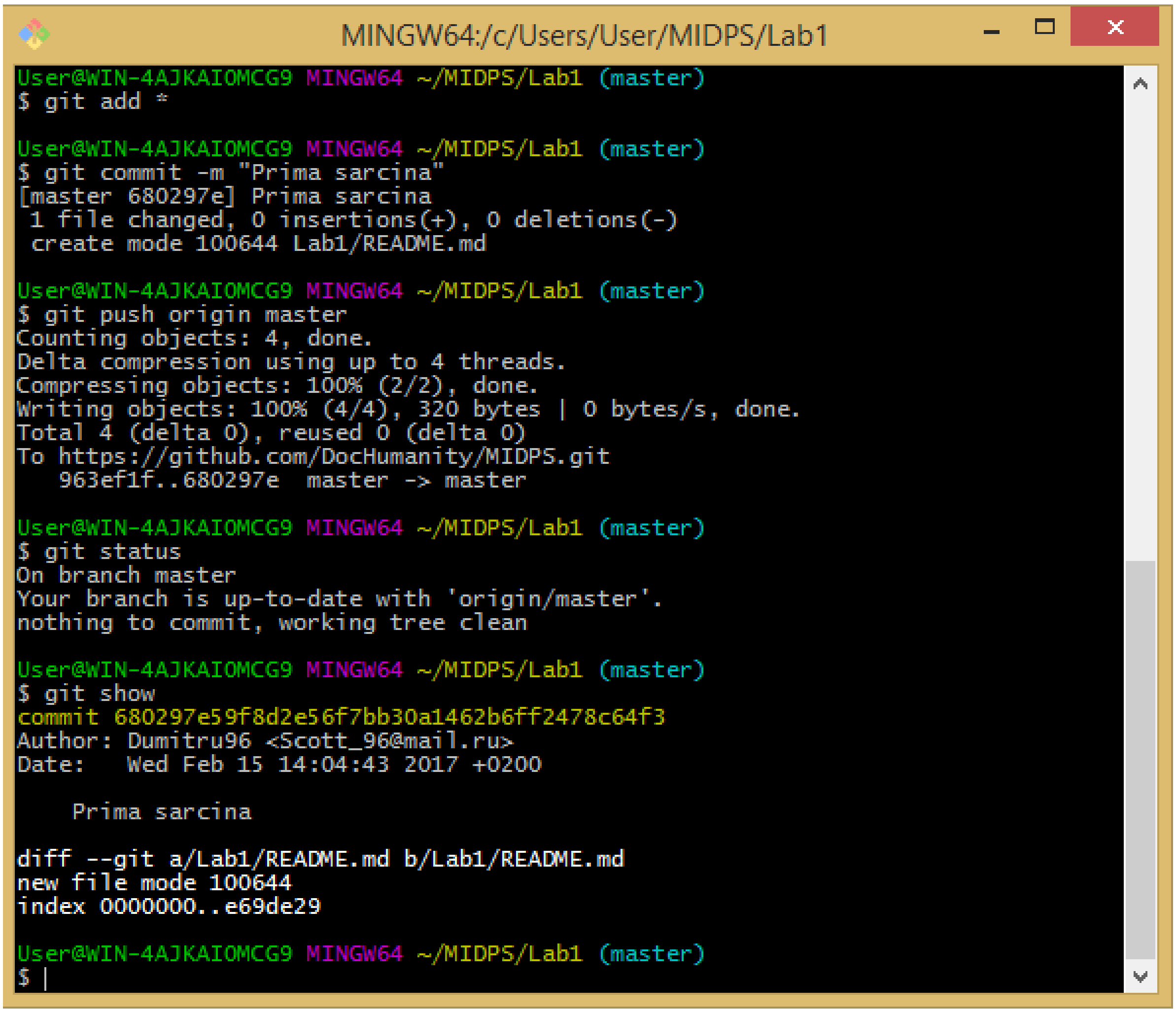
După ce am generat keygen-ul,clonăm repozitoriul pe mașina locală.



```
User@WIN-4AJKAIOMCG9 MINGW64 ~
$ git clone https://github.com/DocHumanity/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.

User@WIN-4AJKAIOMCG9 MINGW64 ~
$ cd MIDPS

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS (master)
$ ls -l
total 1
-rw-r--r-- 1 User 197121 7 фев 15 13:07 README.md
```

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.

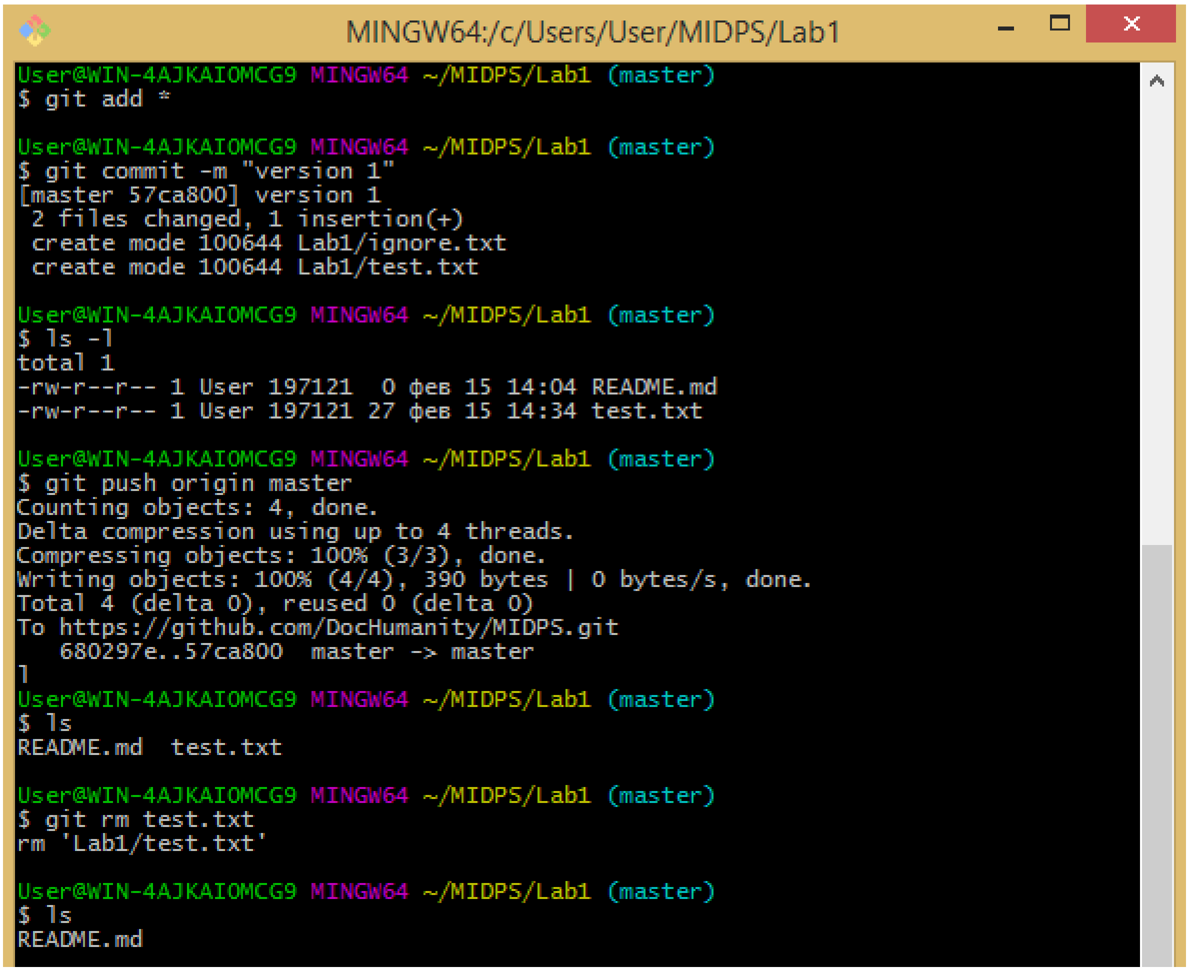
git push origin master - comanda încarcă toate fișierele indexate pe git.Totodată vom folosi **git status** și **git show** pentru a ne asigura că fișierele au fost adăugate în repozitoriu.



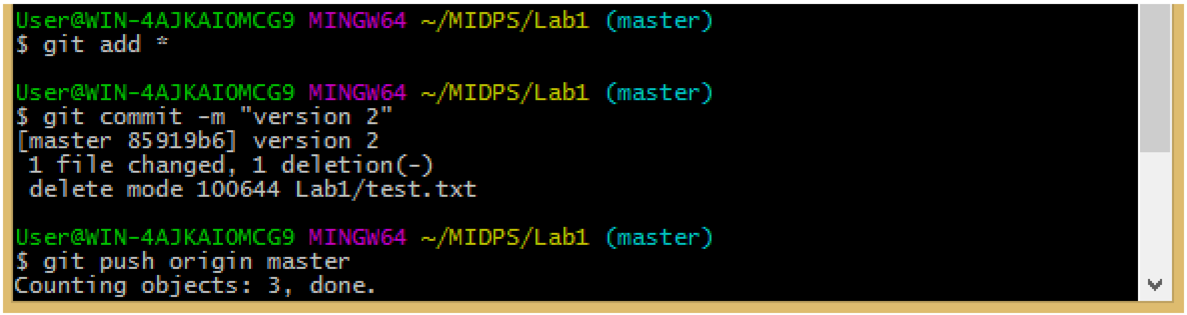
```
User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
$ git add *

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
$ git commit -m "Prima sarcina"
[master 680297e] Prima sarcina
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Lab1/README.md

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (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), 320 bytes | 0 bytes/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/DocHumanity/MIDPS.git
   963ef1f..680297e  master -> master

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working tree clean

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
$ git show
commit 680297e59f8d2e56f7bb30a1462b6ff2478c64f3
Author: Dumitru96 <Scott_96@mail.ru>
Date:   Wed Feb 15 14:04:43 2017 +0200

    Prima sarcina

diff --git a/Lab1/README.md b/Lab1/README.md
new file mode 100644
index 0000000..e69de29

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (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.

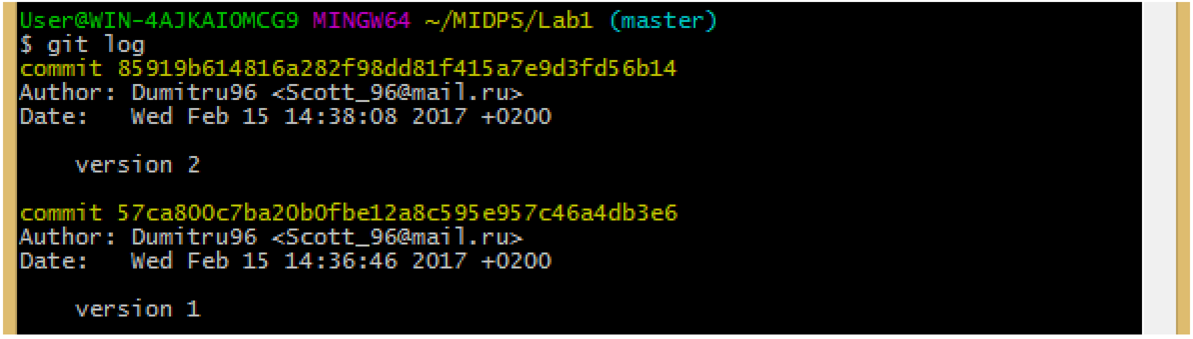
Am creat un fișier nou text.txt în versiunea 1. După care l-am șters și am făcut commit la versiunea 2 în care am sters fișierul test.txt.Dorim să revenim la versiunea1. La început vom lansa comanda **git log** care ne arată logul de commituri și codul pentru fiecare commit. Vom avea nevoie de primele 7 cifre la commitul anterior.



```
User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
$ git add *

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
$ git commit -m "version 1"
[master 57ca800] version 1
 2 files changed, 1 insertion(+)
 create mode 100644 Lab1/ignore.txt
 create mode 100644 Lab1/test.txt

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
$ ls -l
total 1
-rw-r--r-- 1 User 197121  0 фев 15 14:04 README.md
-rw-r--r-- 1 User 197121 27 фев 15 14:34 test.txt

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
$ git push origin master
Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 390 bytes | 0 bytes/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/DocHumanity/MIDPS.git
   680297e..57ca800  master -> master
]
User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
$ ls
README.md  test.txt

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
$ git rm test.txt
rm 'Lab1/test.txt'

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
$ ls
README.md
```

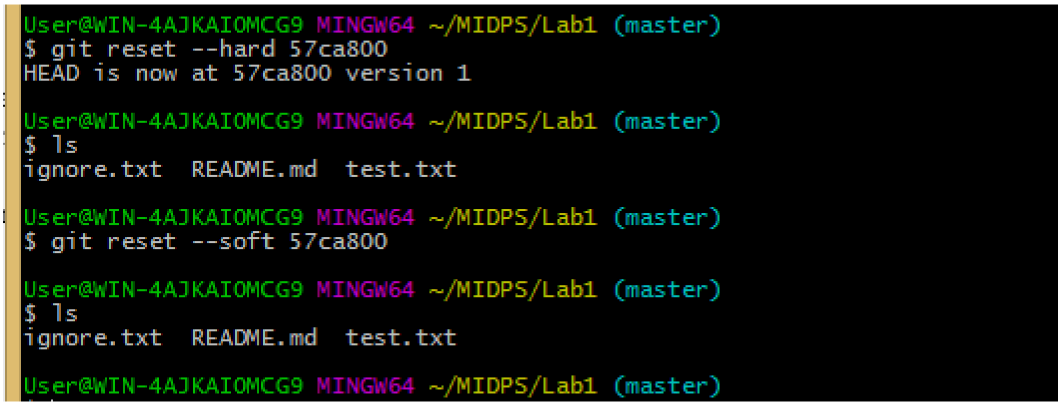


```
User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
$ git add *

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
$ git commit -m "version 2"
[master 85919b6] version 2
 1 file changed, 1 deletion(-)
 delete mode 100644 Lab1/test.txt

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
$ git push origin master
Counting objects: 3, done.
```



```
User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
$ git log
commit 85919b614816a282f98dd81f415a7e9d3fd56b14
Author: Dumitru96 <Scott_96@mail.ru>
Date:   Wed Feb 15 14:38:08 2017 +0200

    version 2

commit 57ca800c7ba20b0fbe12a8c595e957c46a4db3e6
Author: Dumitru96 <Scott_96@mail.ru>
Date:   Wed Feb 15 14:36:46 2017 +0200

    version 1
```

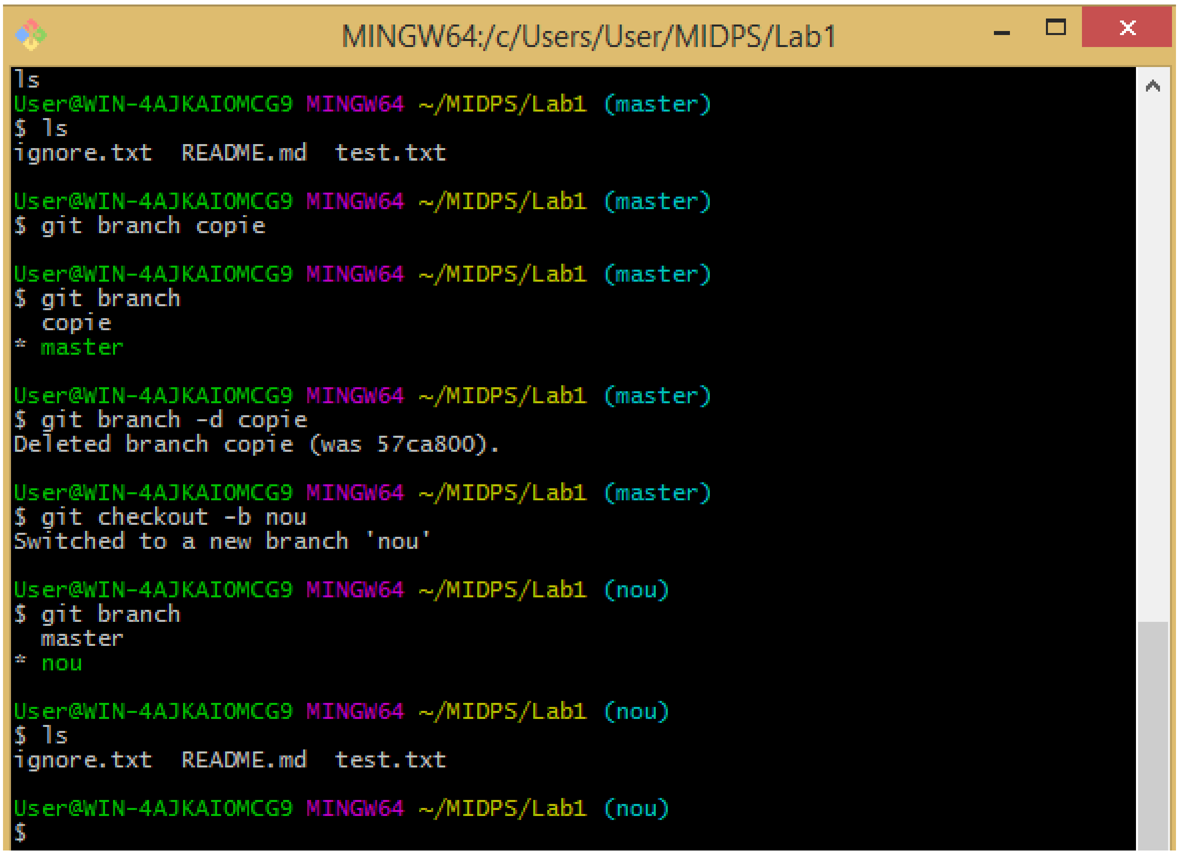
Acum folosim comenzile **git reset --hard** și **git reset --soft**



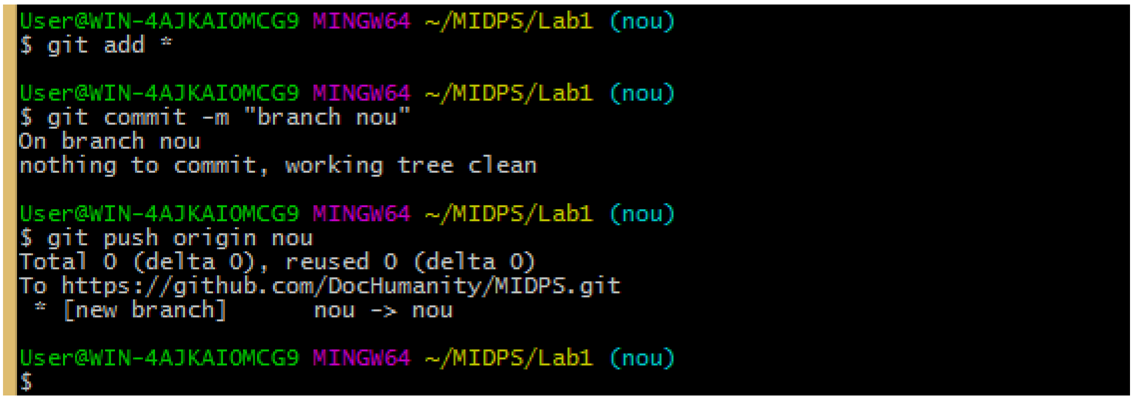
```
User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
$ git reset --hard 57ca800
HEAD is now at 57ca800 version 1

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
$ ls
ignore.txt  README.md  test.txt

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
$ git reset --soft 57ca800

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
$ ls
ignore.txt  README.md  test.txt

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (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 modificarile.

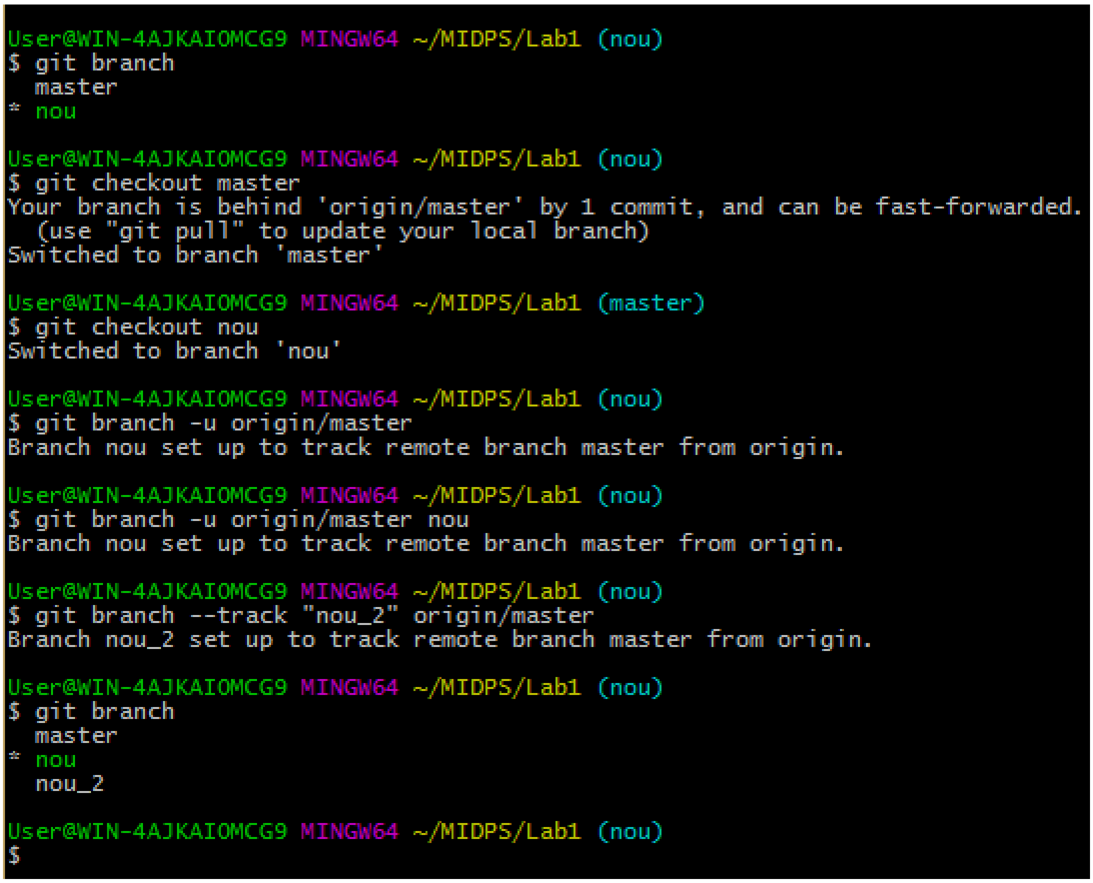
git branch ”name” - creează un branch nou cu numele ”name”. **git branch** - vizualizarea branch-urilor (* indică branch-ul curent). **git branch -d ”nume”** - șterge branch-ul ”nume”. **git checkout -b ”name”** - creează un branch nou cu numele ”name” și face switch la el.



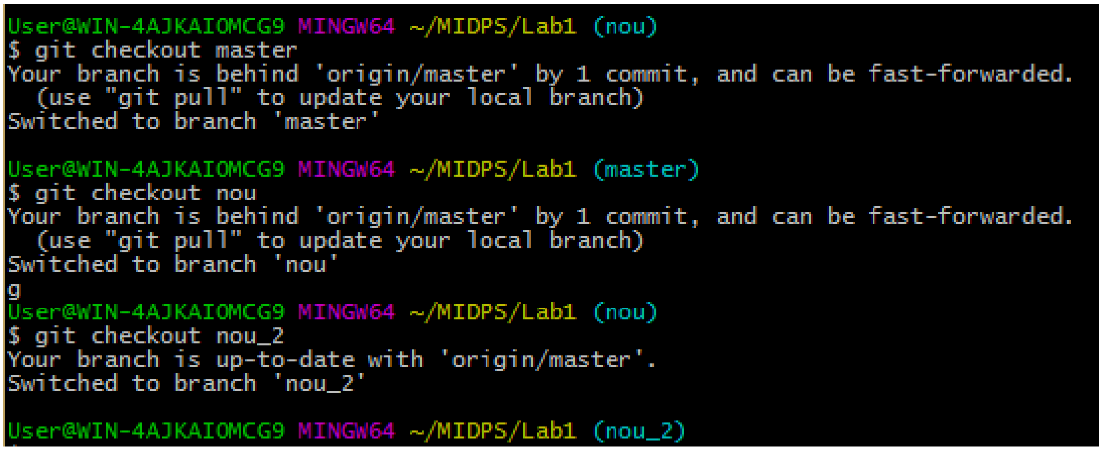
```
ls
User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
$ ls
ignore.txt  README.md  test.txt

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
$ git branch copie

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
$ git branch
  copie
* master

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
$ git branch -d copie
Deleted branch copie (was 57ca800).

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
$ git checkout -b nou
Switched to a new branch 'nou'

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (nou)
$ git branch
  master
* nou

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (nou)
$ ls
ignore.txt  README.md  test.txt

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (nou)
$
```



```
User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (nou)
$ git add *

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (nou)
$ git commit -m "branch nou"
On branch nou
nothing to commit, working tree clean

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (nou)
$ git push origin nou
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/DocHumanity/MIDPS.git
 * [new branch]      nou -> nou

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (nou)
$
```

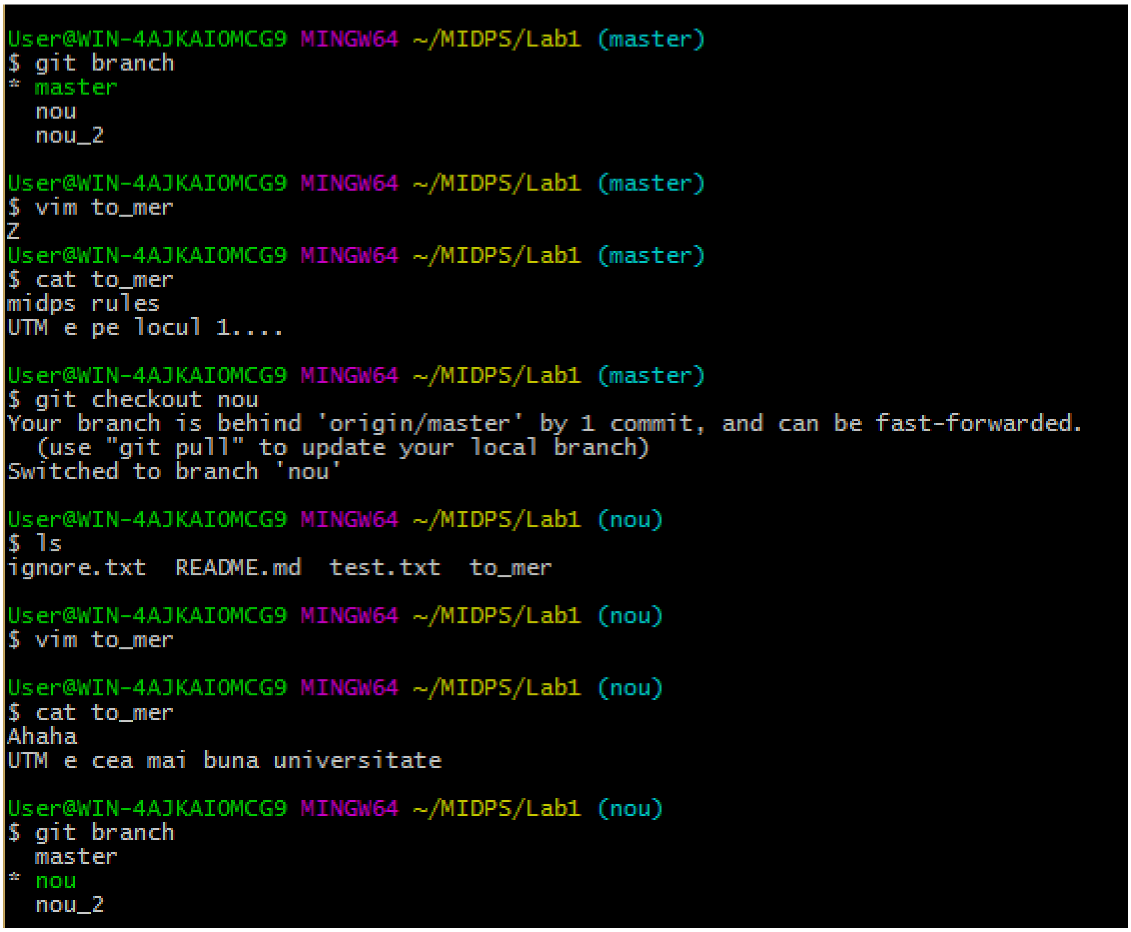


```
User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (nou)
$ git branch
  master
* nou

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (nou)
$ git checkout master
Your branch is behind 'origin/master' by 1 commit, and can be fast-forwarded.
  (use "git pull" to update your local branch)
Switched to branch 'master'

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
$ git checkout nou
Switched to branch 'nou'

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (nou)
$ git branch -u origin/master
Branch nou set up to track remote branch master from origin.

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (nou)
$ git branch -u origin/master nou
Branch nou set up to track remote branch master from origin.

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (nou)
$ git branch --track "nou_2" origin/master
Branch nou_2 set up to track remote branch master from origin.

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (nou)
$ git branch
  master
* nou
  nou_2

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (nou)
$
```

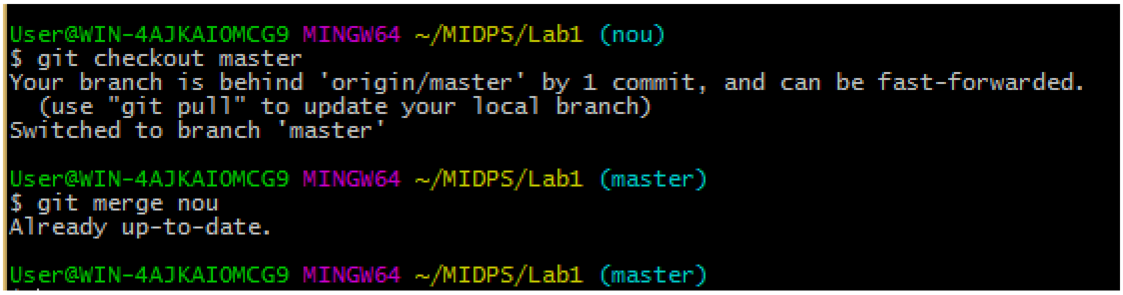


```
User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (nou)
$ git checkout master
Your branch is behind 'origin/master' by 1 commit, and can be fast-forwarded.
  (use "git pull" to update your local branch)
Switched to branch 'master'

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
$ git checkout nou
Your branch is behind 'origin/master' by 1 commit, and can be fast-forwarded.
  (use "git pull" to update your local branch)
Switched to branch 'nou'
g
User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (nou)
$ git checkout nou_2
Your branch is up-to-date with 'origin/master'.
Switched to branch 'nou_2'

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (nou_2)
```

Vom lucra cu 2 branch-uri - ”master” si ”nou”. Vom crea în fiecare branch câte un fișier ”to_mer”,dar conținutul fiecăruia va fi diferit.



```
User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
$ git branch
* master
  nou
  nou_2

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
$ vim to_mer
Z
User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
$ cat to_mer
midps rules
UTM e pe locul 1....

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
$ git checkout nou
Your branch is behind 'origin/master' by 1 commit, and can be fast-forwarded.
  (use "git pull" to update your local branch)
Switched to branch 'nou'

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (nou)
$ ls
ignore.txt  README.md  test.txt  to_mer

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (nou)
$ vim to_mer

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (nou)
$ cat to_mer
Ahaha
UTM e cea mai buna universitate

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (nou)
$ git branch
  master
* nou
  nou_2
```



```
User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (nou)
$ git checkout master
Your branch is behind 'origin/master' by 1 commit, and can be fast-forwarded.
  (use "git pull" to update your local branch)
Switched to branch 'master'

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
$ git merge nou
Already up-to-date.

User@WIN-4AJKAIOMCG9 MINGW64 ~/MIDPS/Lab1 (master)
```

Concluzie: Am studiat VCS.Mi-am aprofundat cunoștiințele 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.