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 Iordachi Cristian

A verificat: Gojin V.

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) •

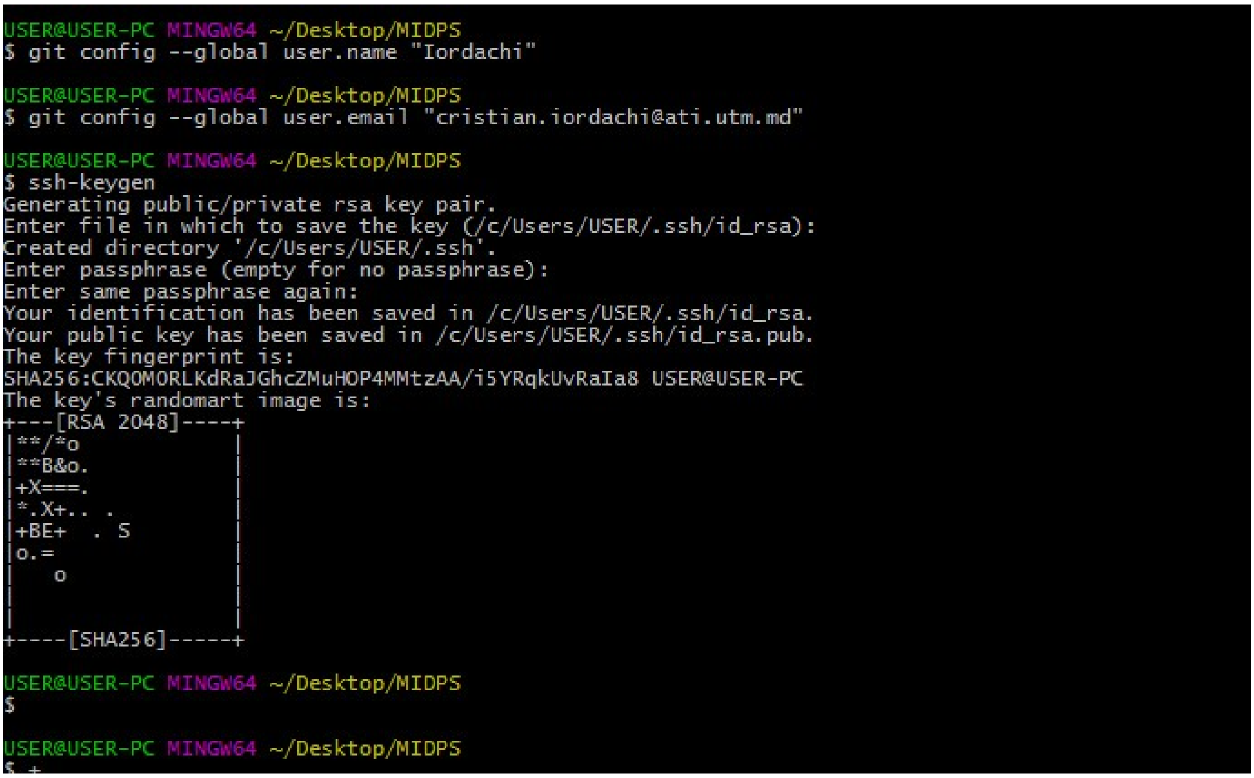
  Normal Level :
  - seteaza un branch to track a remote origin pe care vei putea sa faci push (ex. Github, Bitbucket or custom server)
  - reseteaza un branch la commit-ul anterior
  - salvarea temporara a schimbarilor care nu se vor face commit imediat. o folosirea fisierului .gitignore
- *Advanced Level* :
  - merge 2 branches
  - rezolvarea conflictelor a 2 branches
  - comezile git care trebuie cunoscute

Analiza Lucrării de laborator:

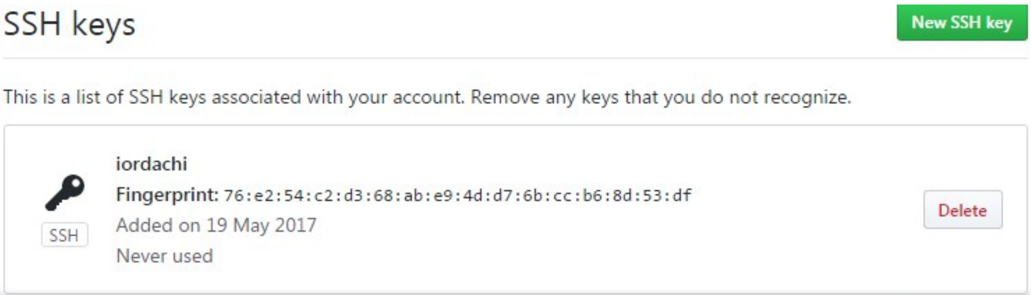
Link-ul la repozitoriu https://github.com/Pastuh2/MIDPS

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”** si generarea cheii SSH pe care o vom copia în setările de pe github.



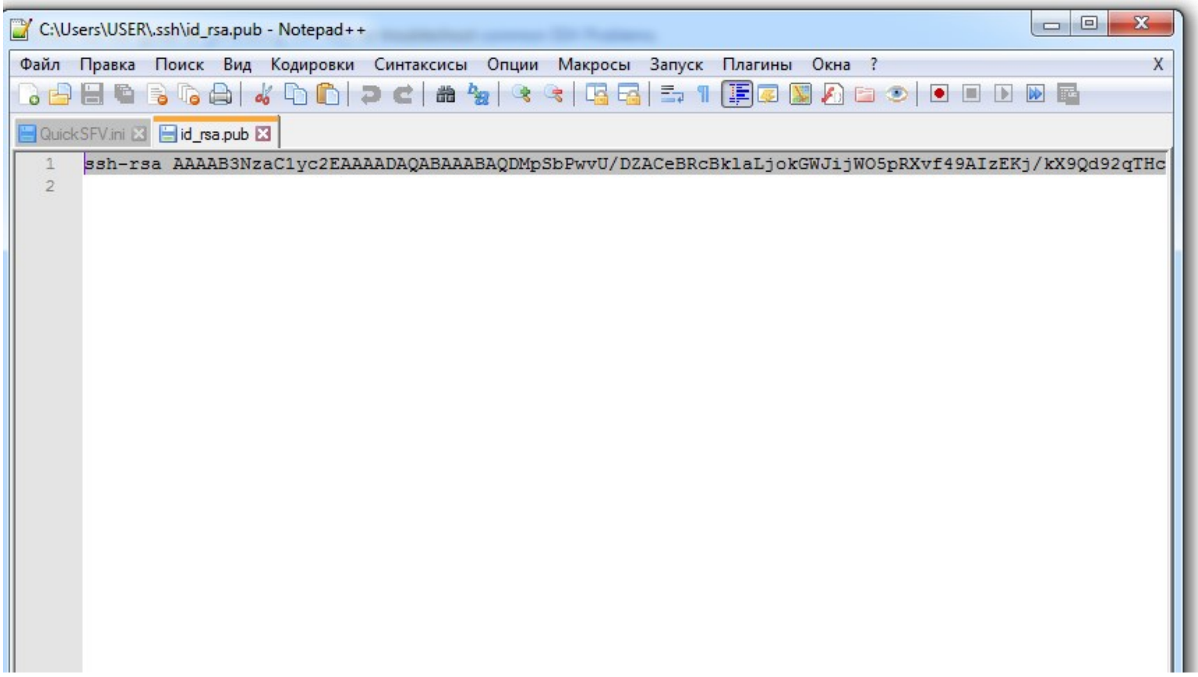
```
USER@USER-PC MINGW64 ~/Desktop/MIDPS
$ git config --global user.name "Iordachi"

USER@USER-PC MINGW64 ~/Desktop/MIDPS
$ git config --global user.email "cristian.iordachi@ati.utm.md"

USER@USER-PC MINGW64 ~/Desktop/MIDPS
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/USER/.ssh/id_rsa):
Created directory '/c/Users/USER/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/USER/.ssh/id_rsa.
Your public key has been saved in /c/Users/USER/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:CKQOMORLKdRaJGhcZMuHOP4MMtzAA/i5YRqkUvRaIa8 USER@USER-PC
The key's randomart image is:
+---[RSA 2048]----+
|**/*o            |
|**B&o.           |
|+X===.           |
|*.X+.. .         |
|+BE+   . S       |
|o.=              |
|    o            |
|                 |
|                 |
+----[SHA256]-----+

USER@USER-PC MINGW64 ~/Desktop/MIDPS
$

USER@USER-PC MINGW64 ~/Desktop/MIDPS
$ +
```



SSH keys

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

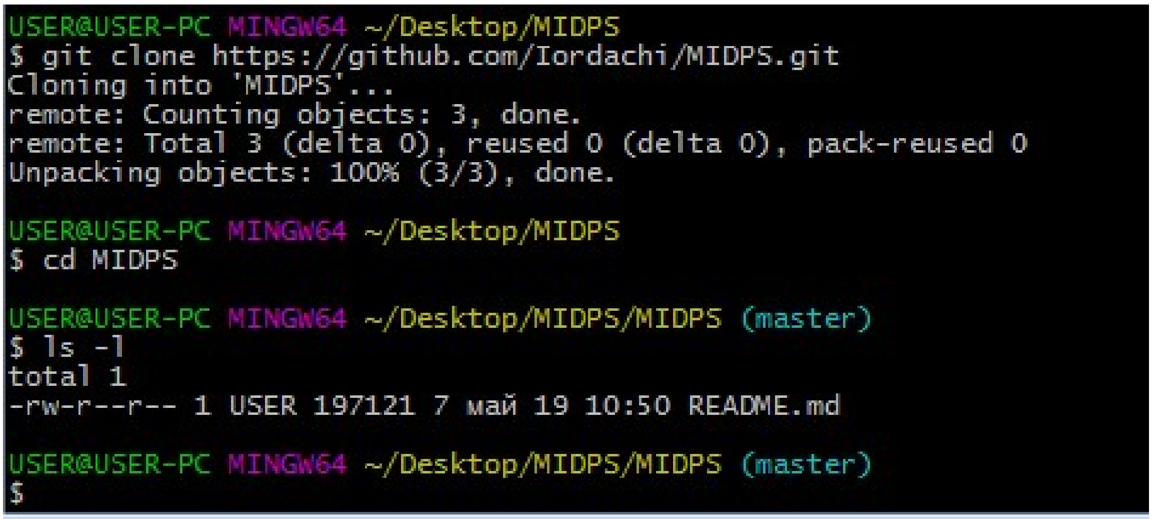
iordachi
Fingerprint: 76:e2:54:c2:d3:68:ab:e9:4d:d7:6b:cc:b6:8d:53:df
Added on 19 May 2017
Never used



C:\Users\USER\.ssh\id_rsa.pub - Notepad++

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDMpSbPwvU/DZACeBRcBklaLjokGWJijWO5pRXvf49AIzEKj/kX9Qd92qTHc
```

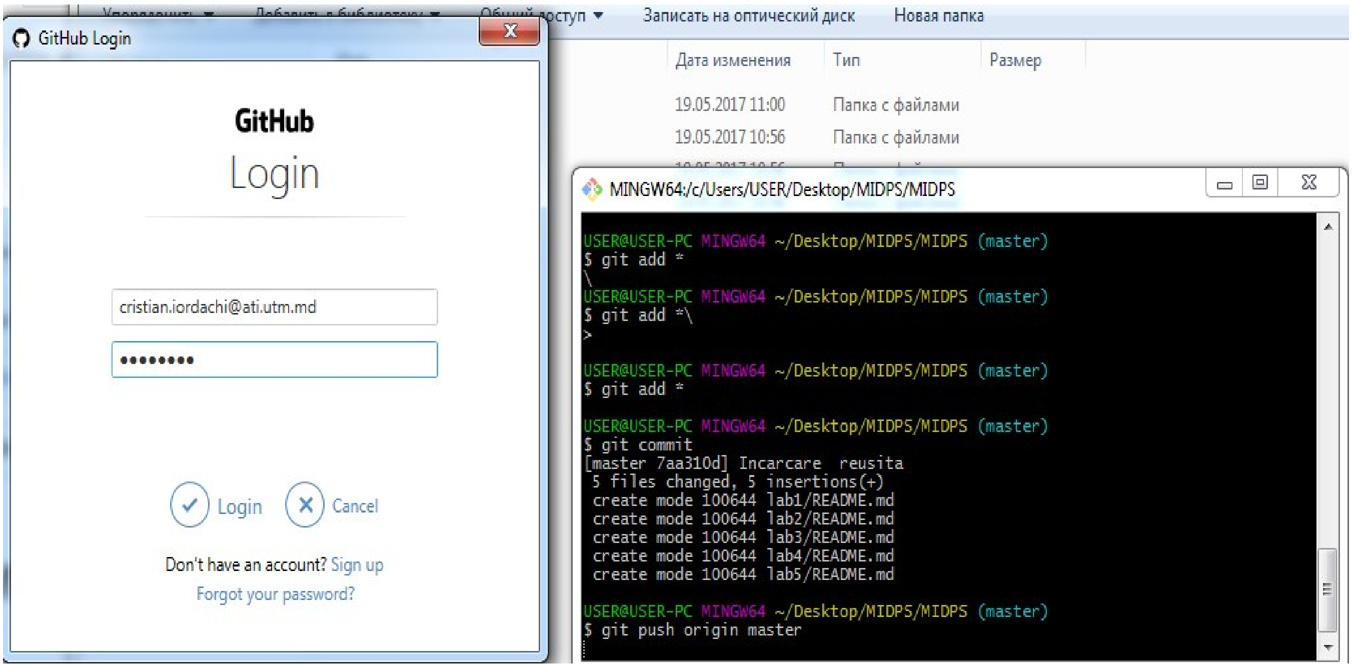
După ce am generat keygen-ul,clonăm repozitoriul pe mașina locală.



```
USER@USER-PC MINGW64 ~/Desktop/MIDPS
$ git clone https://github.com/Iordachi/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@USER-PC MINGW64 ~/Desktop/MIDPS
$ cd MIDPS

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (master)
$ ls -l
total 1
-rw-r--r-- 1 USER 197121 7 май 19 10:50 README.md

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

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.

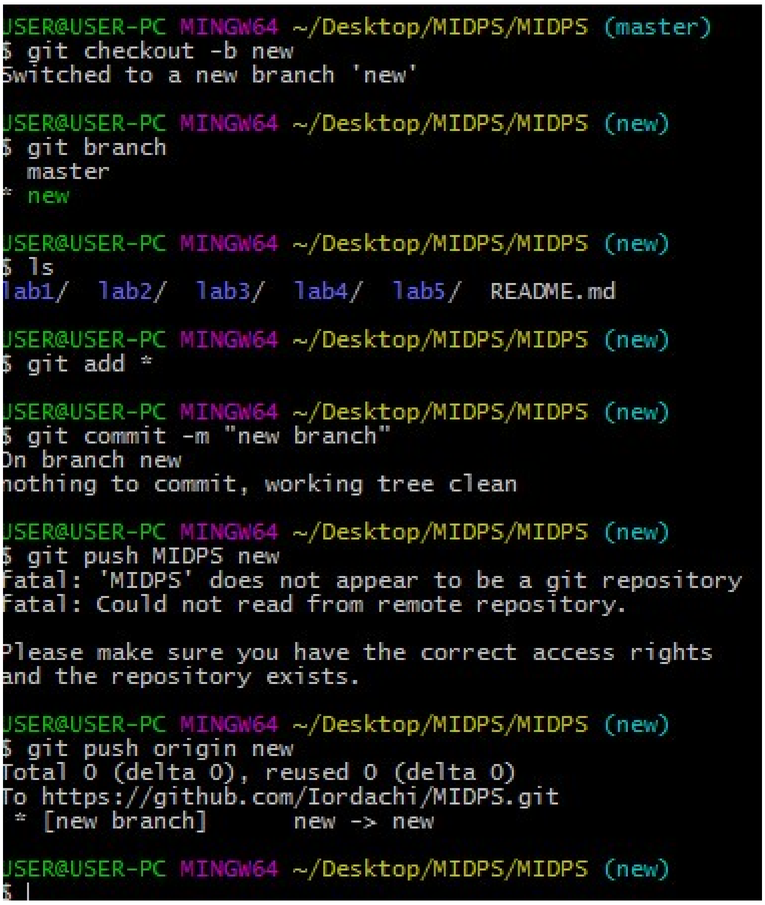
git push origin master - comanda încarcă toate fișierele indexate pe git.Totodată vom folosi **git status** și **git show** pentru a ne asigura că fișierele au fost adăugate în repozitoriu.



cristian.iordachi@ati.utm.md

••••••••

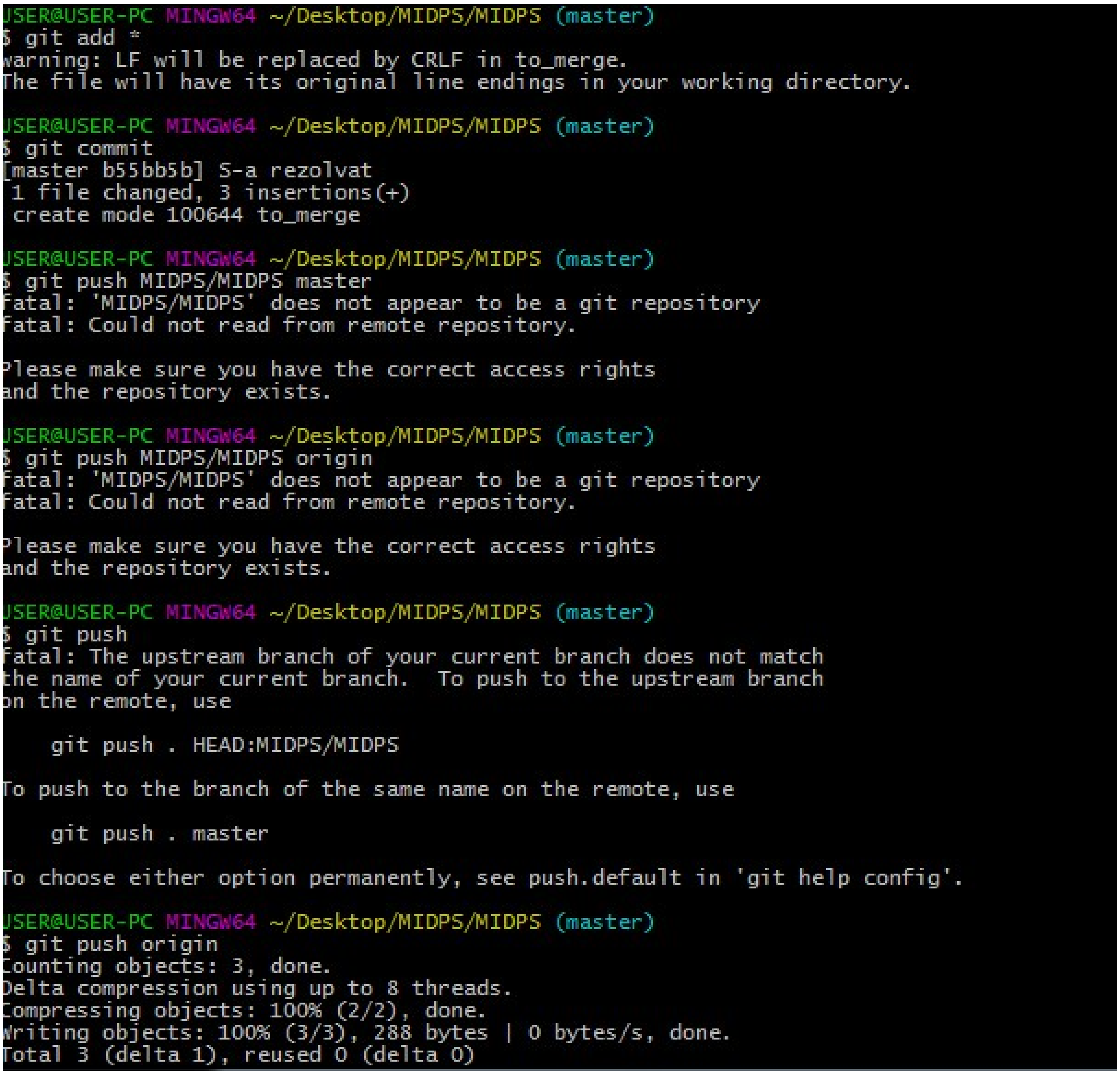
```
USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (master)
$ git add *
\
USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (master)
$ git add *\
>

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (master)
$ git add *

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (master)
$ git commit
[master 7aa310d] Incarcare  reusita
 5 files changed, 5 insertions(+)
 create mode 100644 lab1/README.md
 create mode 100644 lab2/README.md
 create mode 100644 lab3/README.md
 create mode 100644 lab4/README.md
 create mode 100644 lab5/README.md

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (master)
$ git push origin master
```



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

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (new)
$ git branch
  master
* new

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (new)
$ ls
lab1/  lab2/  lab3/  lab4/  lab5/  README.md

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (new)
$ git add *

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

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (new)
$ git push MIDPS new
fatal: 'MIDPS' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

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

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (new)
$ |
```

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.

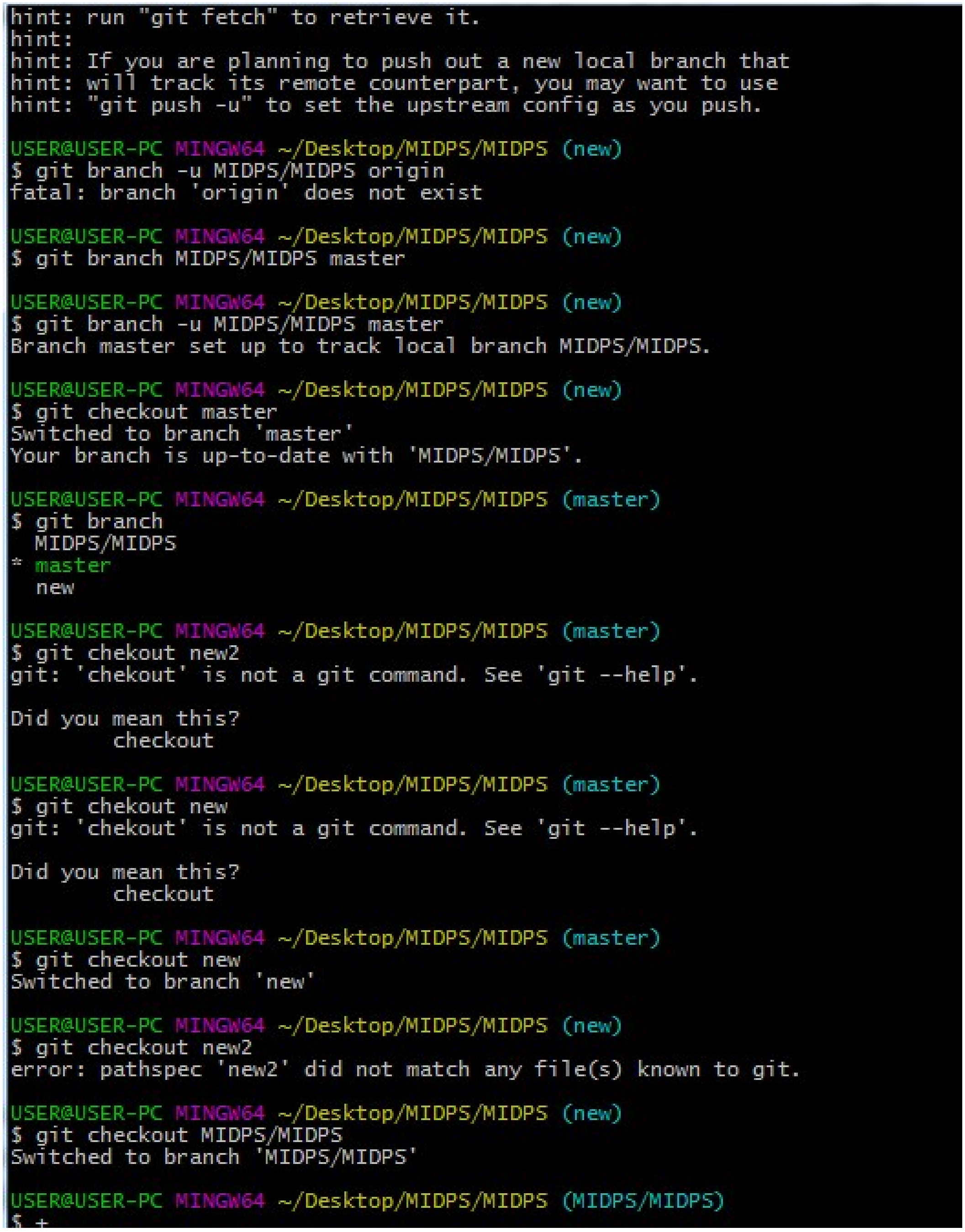
Am creat un fișier nou text.txt în versiunea 1. După care l-am șters și am făcut commit la versiunea 2 în care am sters fișierul test.txt.Dorim să revenim la versiunea1. La început vom lansa comanda **git log** care ne arată logul de commituri și codul pentru fiecare commit. Vom avea nevoie de primele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



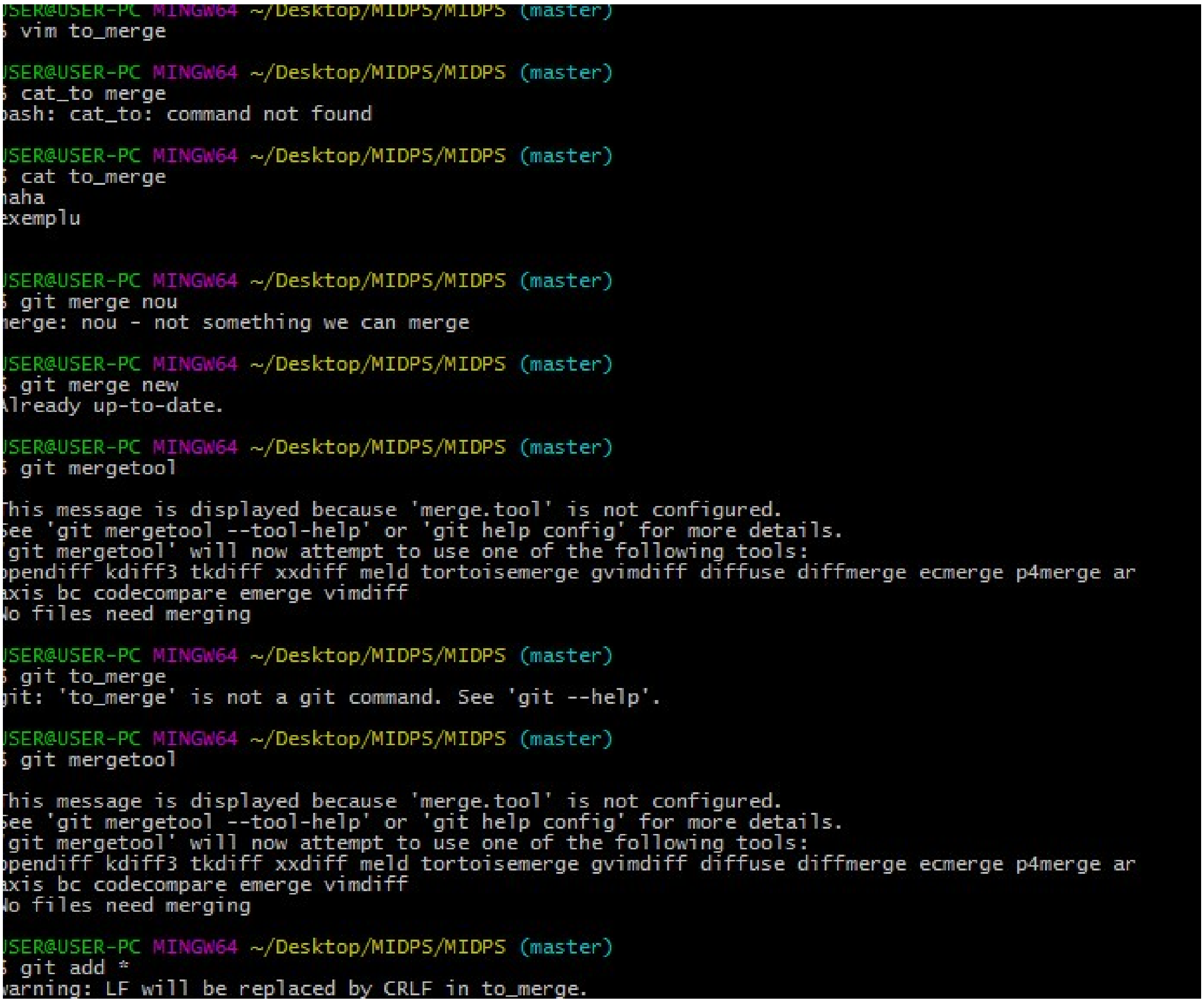
```
USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (master)
$ git add *
warning: LF will be replaced by CRLF in to_merge.
The file will have its original line endings in your working directory.

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (master)
$ git commit
[master b55bb5b] S-a rezolvat
 1 file changed, 3 insertions(+)
 create mode 100644 to_merge

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (master)
$ git push MIDPS/MIDPS master
fatal: 'MIDPS/MIDPS' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (master)
$ git push MIDPS/MIDPS origin
fatal: 'MIDPS/MIDPS' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (master)
$ git push
fatal: The upstream branch of your current branch does not match
the name of your current branch.  To push to the upstream branch
on the remote, use

    git push . HEAD:MIDPS/MIDPS

To push to the branch of the same name on the remote, use

    git push . master

To choose either option permanently, see push.default in 'git help config'.

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (master)
$ git push origin
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 288 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
```

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.

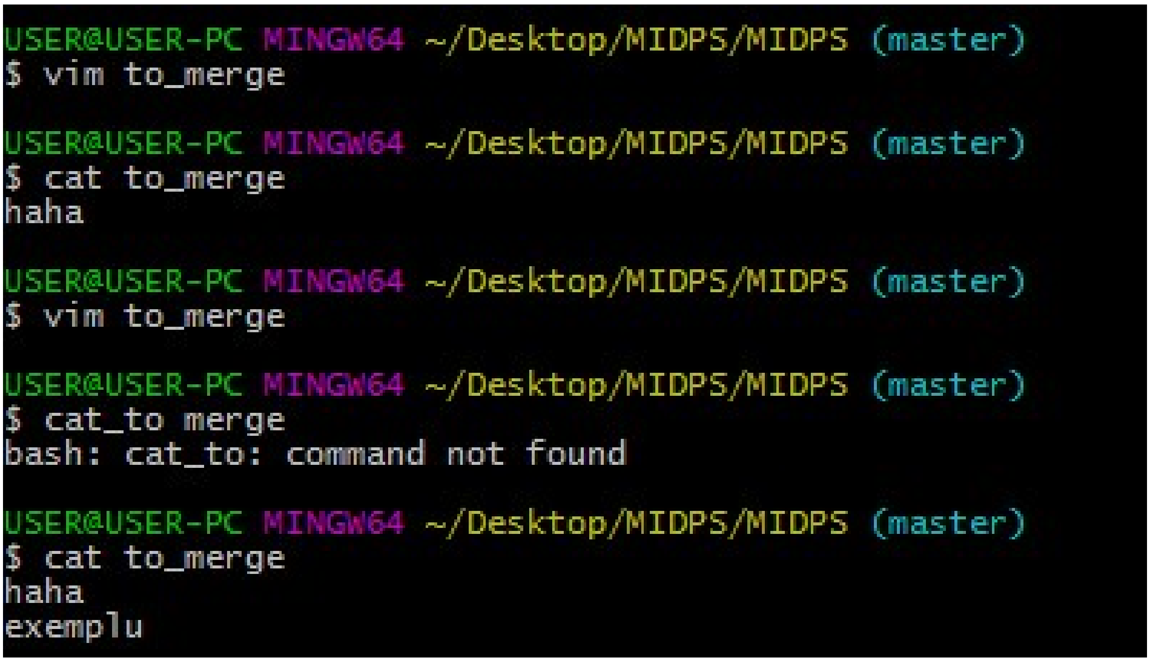
git branch ”name” - creează un branch nou cu numele ”name”. **git branch** - vizualizarea branch-urilor (* indică branch-ul curent). **git branch -d ”nume”** - șterge branch-ul ”nume”. **git checkout -b ”name”** - creează un branch nou cu numele ”name” și face switch la el



```
hint: run "git fetch" to retrieve it.
hint:
hint: If you are planning to push out a new local branch that
hint: will track its remote counterpart, you may want to use
hint: "git push -u" to set the upstream config as you push.

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (new)
$ git branch -u MIDPS/MIDPS origin
fatal: branch 'origin' does not exist

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (new)
$ git branch MIDPS/MIDPS master

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (new)
$ git branch -u MIDPS/MIDPS master
Branch master set up to track local branch MIDPS/MIDPS.

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

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (master)
$ git branch
  MIDPS/MIDPS
* master
  new

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (master)
$ git chekout new2
git: 'chekout' is not a git command. See 'git --help'.

Did you mean this?
        checkout

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (master)
$ git chekout new
git: 'chekout' is not a git command. See 'git --help'.

Did you mean this?
        checkout

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

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (new)
$ git checkout new2
error: pathspec 'new2' did not match any file(s) known to git.

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (new)
$ git checkout MIDPS/MIDPS
Switched to branch 'MIDPS/MIDPS'

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (MIDPS/MIDPS)
$
```



```
USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (master)
$ vim to_merge

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (master)
$ cat_to merge
bash: cat_to: command not found

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (master)
$ cat to_merge
haha
exemplu


USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (master)
$ git merge nou
merge: nou - not something we can merge

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (master)
$ git merge new
Already up-to-date.

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (master)
$ 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 ecmerge p4merge ar
axis bc codecompare emerge vimdiff
No files need merging

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (master)
$ git to_merge
git: 'to_merge' is not a git command. See 'git --help'.

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (master)
$ 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 ecmerge p4merge ar
axis bc codecompare emerge vimdiff
No files need merging

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (master)
$ git add *
warning: LF will be replaced by CRLF in to_merge.
```



```
USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (master)
$ vim to_merge

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (master)
$ cat to_merge
haha

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (master)
$ vim to_merge

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (master)
$ cat_to merge
bash: cat_to: command not found

USER@USER-PC MINGW64 ~/Desktop/MIDPS/MIDPS (master)
$ cat to_merge
haha
exemplu
```

Concluzie: Am studiat VCS.Mi-am aprofundat cunoştiinţele în GitHub.Am învăţat cum se creează mai multe branch-uri,cum se mută de la unul la altul,să fac operaţiile 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.
```