

FACULTATEA CALCULATOARE, INFORMATICA SI MICROELECTRONICA

UNIVERSITATEA TEHNICA A MOLDOVEI

MEDII INTERACTIVE DE DEZVOLTARE A PRODUSELOR SOFT

LUCRAREA DE LABORATOR#2

---

## Version Control Systems si modul de setare a unui server

---

*Autor:*

Croitoru IONEL

*lector asistent:*

Irina COJANU

*lector superior:*

Svetlana COJOCARU

## **Lucrarea de laborator #2**

### **1 Scopul lucrării de laborator**

Version Control Systems si modul de setare a unui server

### **2 Obiective**

- Intelegerea si folosirea CLI (basic level)
- Administrarea remote a masinilor linux machine folosind SSH (remote code editing)
- Version Control Systems (git — mercurial — svn)
- Compileaza codul C/C++/Java/Python prin intermediul CLI, folosind compilatoarele gcc/g++/javac/python

### **3 Laboratory work implementation**

#### **3.1 Tasks and Points**

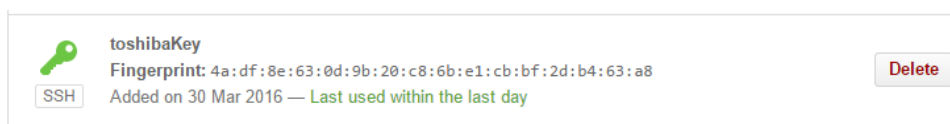
- Basic Level (nota 5 — 6) :
  - conecteaza-te la server folosind SSH
  - compileaza cel putin 2 sample programs din setul HelloWorldPrograms folosind CLI
  - executa primul commit folosind VCS
- Normal Level (nota 7 — 8):
  - initializeaza un nou repository
  - configureaza-ti VCS
  - crearea branch-urilor (creeaza cel putin 2 branches)
  - commit pe ambele branch-uri (cel putin 1 commit per branch)
- Advanced Level (nota 9 — 10):
  - 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
  - merge 2 branches
  - conflict solving between 2 branches

#### **3.2 Analiza lucrarii de laborator**

GitHub este un serviciu de găzduire web pentru proiecte de dezvoltare a software-ului care utilizează sistemul de control al versiunilor Git. GitHub oferă planuri tarifare pentru depozite private, și conturi gratuite pentru proiecte open source. Site-ul a fost lansat în 2008 de către Tom Preston-Werner, Chris Wanstrath, și PJ Hyett.



Pentru elaborarea acestui laborator m-am inregistrat pe github, am creat un repository si am instalat git bash-ul in calculator . Am reusit sa ma conectez la repositoryul creat folosind SSH prin git remote add origin email. Am creat o cheia ce ma autorizeaza automat prin ssh-keygen care creeaza o parola autentificarea de la acest device.



```
catea@catea-pc MINGW64 /b/MIDPS (master)
$ ssh -T git@github.com
Enter passphrase for key '/c/Users/catea/.ssh/id_rsa':
Hi croitoruion! You've successfully authenticated, but GitHub does not provide s
hell access.
```

Pentru compilarea si rulara unor programe hello world de exemplu scrise in limbajul java si C++ avem nevoie in primul rand de jdk shi gcc. In setari de la windows introducem calea "path" spre jdk si gcc, pentru compilarea acestor programe in continuare. Am creat un file java si cpp , am scris acele hello world programe. Dupa ce ne-am autentificat in github cu shh-key putem cu ajutorul comenzii javac sa compilam programul jhello iar apoi cu ajutorul comenzii java sa rulam acest program

```

catea@catea-pc MINGW64 /b/MIDPS/lab#2 (master)
$ javac jhello.java

catea@catea-pc MINGW64 /b/MIDPS/lab#2 (master)
$ java jhello
Hello GITHUB

```

Pentru a compila si rula un program C++ este comanda g++ "den fisierului" -o "denum file-ului exe" iar apoi sal rulam cu ajutorul comenzii ./"denumirea fisierului exe"

```

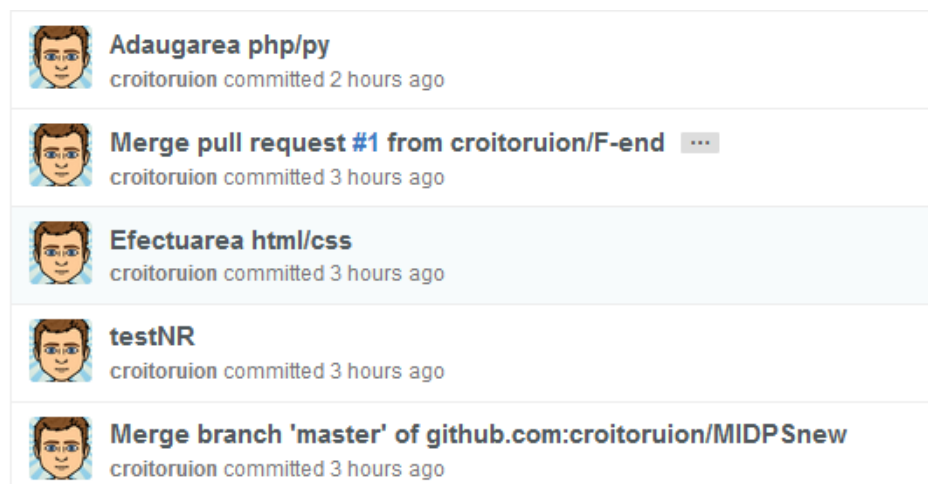
catea@catea-pc MINGW64 /b/MIDPS/lab#2 (master)
$ g++ hello.cpp -o C++hello

catea@catea-pc MINGW64 /b/MIDPS/lab#2 (master)
$ ./C++hello
Hello github!

catea@catea-pc MINGW64 /b/MIDPS/lab#2 (master)
$ F

```

La fiecare push se face si un commit care sistematizeaza orice schimbare pentru o usurare. Commiturile se fac prin comanda "git commit -m "COMENTARIU" ,iar apoi se face push pentru intrarea in vigoare a schimbarilor noastre



Am creat un repository nou MIDPSnew prin git init,apoi am configurat acest repository (name/email din github)

```

MINGW64:/b/midps2
Initialized empty Git repository in B:/midps2/.git/

catea@catea-pc MINGW64 /b/midps2 (master)
$ git config --global user.name "croitoruion"

catea@catea-pc MINGW64 /b/midps2 (master)
$ git config --global user.email "croitoruion1231@gmail.com"

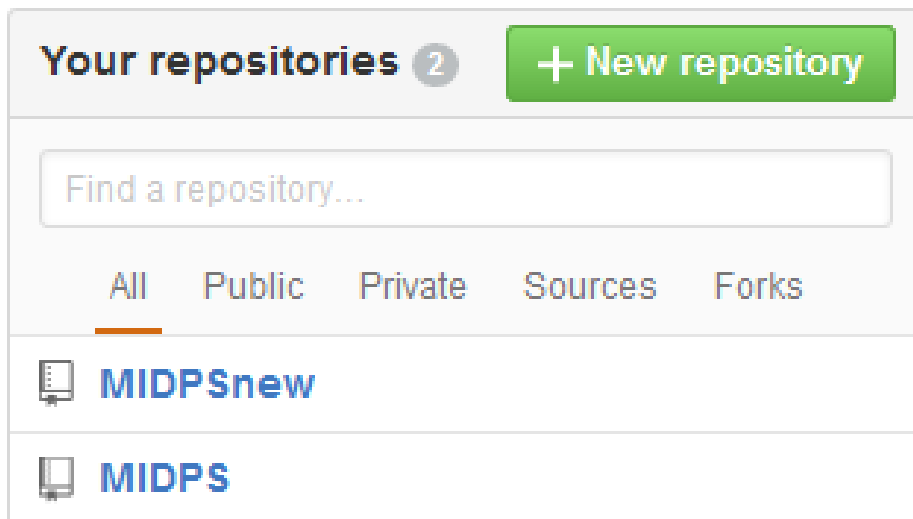
catea@catea-pc MINGW64 /b/midps2 (master)
$ git add .

catea@catea-pc MINGW64 /b/midps2 (master)
$ git commit -m "2"
[master (root-commit) 88f19e4] 2
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.txt

catea@catea-pc MINGW64 /b/midps2 (master)
$ git remote add origin git@github.com:croitoruion/MIDPSnew.git

catea@catea-pc MINGW64 /b/midps2 (master)
$ git status
On branch master
nothing to commit, working directory clean

```



Am creat 2 branch-uri prin ssh cu denumirile F-end si B-end

```
catea@catea-pc MINGW64 /b/midps2 (master)
$ git branch
* master

catea@catea-pc MINGW64 /b/midps2 (master)
$ git branch F-end

catea@catea-pc MINGW64 /b/midps2 (master)
$ git branch B-end

catea@catea-pc MINGW64 /b/midps2 (master)
$ git branch
  B-end
  F-end
* master

catea@catea-pc MINGW64 /b/midps2 (master)
$ |
```

Am creat 2 fisiere din branch-ul B-end si le-am adaugat in repozitoriu

```
catea@catea-pc MINGW64 /b/midps2 (B-end)
$ git status
On branch B-end
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    3.php
    4.py

nothing added to commit but untracked files present (use "git add" to track)

catea@catea-pc MINGW64 /b/midps2 (B-end)
$ git add .

catea@catea-pc MINGW64 /b/midps2 (B-end)
$ git commit -m "Adaugarea php/py"
[B-end 8dcf858] Adaugarea php/py
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 3.php
create mode 100644 4.py
```

Observam ca push-ul se face cu ajutorul ssh-key

```

catea@catea-pc MINGW64 /b/midps2 (B-end)
$ git push origin B-end
Enter passphrase for key '/c/Users/catea/.ssh/id_rsa':
Counting objects: 2, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 283 bytes | 0 bytes/s, done.
Total 2 (delta 0), reused 0 (delta 0)
To git@github.com:croitoruion/MIDPSnew.git
 * [new branch]      B-end -> B-end

catea@catea-pc MINGW64 /b/midps2 (B-end)
$

```

La fel si pentru branch-ul F-end am creat alte 2 fisiere si le-am adaugat in repozitoriu

```

MINGW64:/b/midps2
catea@catea-pc MINGW64 /b/midps2 (master)
$ git checkout F-end
Switched to branch 'F-end'

catea@catea-pc MINGW64 /b/midps2 (F-end)
$ git status
On branch F-end
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:    1.txt
        deleted:    2.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        1.css
        2.html

no changes added to commit (use "git add" and/or "git commit -a")

catea@catea-pc MINGW64 /b/midps2 (F-end)
$

```

```

catea@catea-pc MINGW64 /b/midps2 (F-end)
$ git push origin F-end
Enter passphrase for key '/c/Users/catea/.ssh/id_rsa':
Counting objects: 2, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 276 bytes | 0 bytes/s, done.
Total 2 (delta 0), reused 0 (delta 0)
To git@github.com:croitoruion/MIDPSnew.git
 * [new branch]      F-end -> F-end

catea@catea-pc MINGW64 /b/midps2 (F-end)
$

```

Cand intram pe github in calitate de master observam ca branch-urile noastre create au facut push,putem accepta acest push putem sa nu-l acceptam astfel in repozitoriu nu se va schimba nimic.Putem adauga comentarii la acel care a facut acest push/commit

base: master ... compare: F-end ✓ Able to merge. These branches can be automatically merged.

Efectuarea html/css

Write Preview AA B i “ < > ⌂ ☰ ☷ ☹ ↶ @ 📌

good job|

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Styling with Markdown is supported

Create pull request

1 commit 2 files changed 0 commit comments

Astfel in urma acceptarii ambelor comituri din partea ambelor branch-uri observam ca in repozitoriu a aparut 4 fisiere cate 2 din partea fiecarui branch

1.css	Efectuarea html/css
2.html	Efectuarea html/css
3.php	Adaugarea php/py
4.py	Adaugarea php/py

Setam un branch to track remote origin pe care vom putea face push Am selectat branch-ul B-end Am facut o schimbare,astfel am facut un commit

```
catea@catea-pc MINGW64 /b/midps2 (B-end)
$ git branch
* B-end
  F-end
  master

catea@catea-pc MINGW64 /b/midps2 (B-end)
$ git push --set-upstream origin B-end
Enter passphrase for key '/c/Users/catea/.ssh/id_rsa':
Branch B-end set up to track remote branch B-end from origin.
Everything up-to-date
```

Am intrat in log-urile unde se afla toate comiturile si ne-am reintors la starea anterioara adica la penultimul commit



```
MINGW64:/b/midps2

Efectuarea html/css
* commit daae7c621e6e3d138e186fc2f177f82e2d1d239f (origin/F-end, F-end)
/ Author: croitoruion <croitoruion1231@gmail.com>
Date: Wed Mar 30 20:21:52 2016 +0300

Efectuarea html/css
:
catea@catea-pc MINGW64 /b/midps2 (B-end)
$ git checkout 8dcf858
Note: checking out '8dcf858'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

    git checkout -b <new-branch-name>

HEAD is now at 8dcf858... Adaugarea php/py
catea@catea-pc MINGW64 /b/midps2 ((8dcf858...))
$
```

Merge 2 branches. Pentru a face merge la 2 branch-uri utilizam comanda merge.

```
catea@catea-pc MINGW64 /b/midps2 (F-end)
$ git merge master
Updating 92124a1..1838da9
Fast-forward
 common.txt | 2 --
 conflict.txt | 2 +-
 2 files changed, 1 insertion(+), 3 deletions(-)
 delete mode 100644 common.txt
```

Rezolvarea conflictelor a 2 branch-uri

Am creat 2 fisiere identice dar cu continut diferit in fiecare din branch astfel cand dam merge observam ca apare conflictul

```
catea@catea-pc MINGW64 /b/midps2 (B-end)
$ git merge F-end
Auto-merging conflict.txt
CONFLICT (add/add): Merge conflict in conflict.txt
Removing 2.txt
Removing 1.txt
Automatic merge failed; fix conflicts and then commit the result.

catea@catea-pc MINGW64 /b/midps2 (B-end|MERGING)
$ |
```

Github-ul ne indica si ne ajuta sa rezolvam acest conflict

```
Файл  Правка  Формат  Вид  Справка
<<<<<< HEAD
hello
=====
not hello
>>>>>> F-end
```

Stergem ce nu ne trebuie si lasam ceea de ce avem nevoie apoi ii dam commit

## Concluzie

În urma efectuării acestei lucrări de laborator am studiat serviciile de bază a rețelei sociale "IT" numite github.com . Githubul oferă gazduire web pentru proiecte de dezvoltare a software-ului care utilizează sistemul de control al Git-ului. Am efectuat pe rând toate task-urile care au fost propuse în condițiile laboratorului precum initializarea și setarea repositoryului, autentificarea ssh, compilarea și rularea programelor Java și C++, adăugarea commiturilor, crearea a mai multor branch-uri, lucrul cu aceste branch-uri precum, commiturile de pe branch și acceptarea lor de master, efectuarea a merge la branch-uri, diminuarea conflictelor de fișiere. Totul se efectuează prin git Bash care este un terminal destul de comod cu comenzi destul de ușoare și eficiente Githubul reprezintă un instrument ce vine în ajutorul echipelor/firmelor/comuniilor de dezvoltare a tuturor serviciilor IT.

## References

- 1 [stackoverflow.com/repository init](https://stackoverflow.com/questions/10474840/git-repository-init)
- 2 [google.com/git set previous commit](https://google.com/search?q=git+set+previous+commit)
- 3 [google.com/ls equivalent cmd](https://google.com/search?q=ls+equivalent+cmd)
- 4 <https://www.youtube.com/watch?v=ZFYhW3kBjnE>
- 5 [helpgithub.com](https://help.github.com/)