

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:

Ciulcov DUMITRU

lector asistent:

Irina COJANU

lector superior:

Svetlana COJOCARU

Laboratory work #2

1 Scopul lucrarii 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 Realizarea lucrarii de laborator

3.1 Tasks and Points

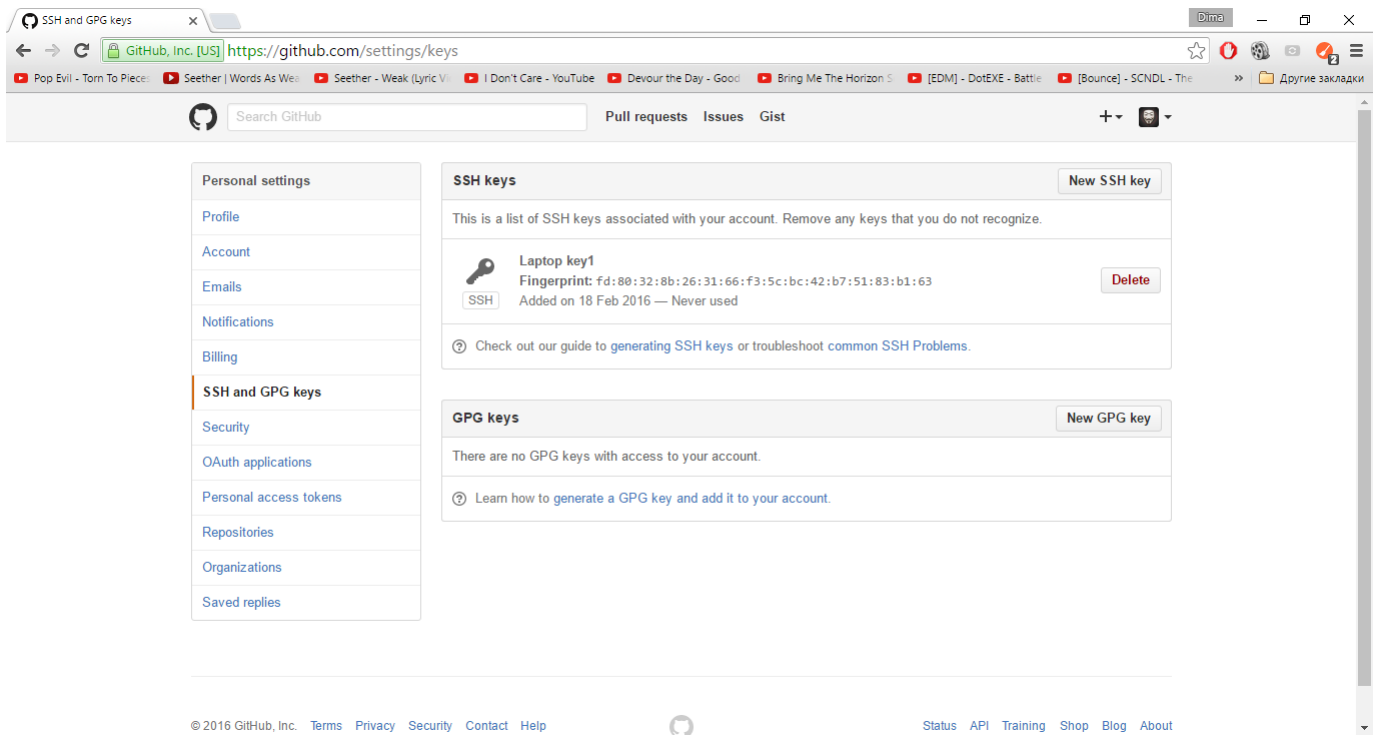
- Basic Level (nota 5 — 6) :
 - conecteaza-te la server folosind SSH
 - compileaza cel putin 2 sample programs din setul HelloWorldPrograms folosind CLI
 - executa primul commit folosind VCS
- Normal Level (nota 7 — 8):
 - initializeaza un nou repository
 - configureaza-ti VCS
 - crearea branch-urilor (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
 - rezolvarea conflictelor a 2 branches

3.2 Analiza lucrarii de laborator

Linkul la repositoryul GITHUB:

<https://github.com/dmitrii724/MIDPS>

Pentru a realiza aceasta lucrare de laborator *m – am* inregistrat pe github.com si am instalat *git – bash*, am generat o cheie SSH si am adaugat aceasta cheie publica pe github pentru a identifica acest calculator.



Pentru a compila programe scrise in $C++$, $Java$ avem nevoie de a seta directiile spre $g++$, $javac$ in fisierul `bash_profile` din directoriul unde este instalat *Git – Bash*. Pentru a compila programul scris in $Java$ utilizam `javac` pentru compilare si `java HelloWorld` pentru a rula programul nostru, in cazul programuli $C++$ utilizam comanda `g++ hello.cpp -o hello`, si `./hello`.

```

MINGW64/d/Univer/Anul 2/Semestru 2/midps lab 2
user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/midps lab 2
$ g++ HelloWorld.cpp -o HelloWorld

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/midps lab 2
$ ls -la
total 1102
drwxr-xr-x 1 user 197608 0 май 21 20:14 ./
drwxr-xr-x 1 user 197608 0 май 21 20:05 ../
-rw-r--r-- 1 user 197608 85990 май 7 19:51 1.png
-rw-r--r-- 1 user 197608 22882 май 21 20:13 2.png
-rw-r--r-- 1 user 197608 426 май 21 20:12 HelloWorld.class
-rw-r--r-- 1 user 197608 103 май 21 20:07 HelloWorld.cpp
-rwxr-xr-x 1 user 197608 1001851 май 21 20:14 HelloWorld.exe
-rw-r--r-- 1 user 197608 192 май 21 20:12 HelloWorld.java

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/midps lab 2
$ ./HelloWorld
Hello World!

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/midps lab 2
$

```

```
MINGW64:/d/Univer/Anul 2/Semestru 2/midps lab 2
$ ./HelloWorld
Hello World!

user@WIN-P5ACQNT3OHH MINGW64 /d/Univer/Anul 2/Semestru 2/midps lab 2
$ javac HelloWorld.java

user@WIN-P5ACQNT3OHH MINGW64 /d/Univer/Anul 2/Semestru 2/midps lab 2
$ ls -la
total 1102
drwxr-xr-x 1 user 197608 0 май 21 20:14 ./
drwxr-xr-x 1 user 197608 0 май 21 20:05 ../
-rw-r--r-- 1 user 197608 85990 май 7 19:51 1.png
-rw-r--r-- 1 user 197608 24376 май 21 20:15 2.png
-rw-r--r-- 1 user 197608 426 май 21 20:16 HelloWorld.class
-rw-r--r-- 1 user 197608 103 май 21 20:07 HelloWorld.cpp
-rwxr-xr-x 1 user 197608 1001851 май 21 20:14 HelloWorld.exe#
-rw-r--r-- 1 user 197608 192 май 21 20:12 HelloWorld.java

user@WIN-P5ACQNT3OHH MINGW64 /d/Univer/Anul 2/Semestru 2/midps lab 2
$ java HelloWorld
Hello, World

user@WIN-P5ACQNT3OHH MINGW64 /d/Univer/Anul 2/Semestru 2/midps lab 2
$ |
```

Pentru fiecare schimbare pe care o facem pe repository putem lasa un mesaj folosind comanda `git commit -m "mesaj"` astfel organizam mai bine repositoryul si putem vedea ce schimbari au avut loc. Pentru a facea primul add, commit si push am utilizat urmatoarele comenzi necesare.

```
MINGW64:/d/Univer/Anul 2/Semestru 2/MIDPS
user@WIN-P5ACQNT3OHH MINGW64 /d/Univer/Anul 2/Semestru 2/MIDPS (master)
$ git add
Nothing specified, nothing added.
Maybe you wanted to say 'git add .'
user@WIN-P5ACQNT3OHH MINGW64 /d/Univer/Anul 2/Semestru 2/MIDPS (master)
$ git add MIDPS\ lab\ 2/example.txt
user@WIN-P5ACQNT3OHH MINGW64 /d/Univer/Anul 2/Semestru 2/MIDPS (master)
$ git commit -m "example.txt"
[master 91cb942] example.txt
1 file changed, 1 insertion(+)
create mode 100644 MIDPS lab 2/example.txt
user@WIN-P5ACQNT3OHH MINGW64 /d/Univer/Anul 2/Semestru 2/MIDPS (master)
$ git push
warning: push.default is unset; its implicit value has changed in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the traditional behavior, use:

    git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

    git config --global push.default simple

When push.default is set to 'matching', git will push local branches
to the remote branches that already exist with the same name.

Since Git 2.0, Git defaults to the more conservative 'simple'
behavior, which only pushes the current branch to the corresponding
remote branch that 'git pull' uses to update the current branch.

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

Counting objects: 4, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 317 bytes | 0 bytes/s, done.
Total 4 (delta 1), reused 0 (delta 0)
To http://github.com/dmitrii724/MIDPS.git
 8c09590..91cb942 master -> master
user@WIN-P5ACQNT3OHH MINGW64 /d/Univer/Anul 2/Semestru 2/MIDPS (master)
$
```

Am initializat un nou repository cu numele NewRepository cu git init, si am configurat acest repository cu git config -global user.name si user.email.

```
MINGW64:/d/Univer/Anul 2/Semestru 2/NewRepository
user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository
$ echo "# NewRepository" >> README.md

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository
$ git init
Initialized empty Git repository in D:/Univer/Anul 2/Semestru 2/NewRepository/.git/

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (master)
$ git add README.md
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory.

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (master)
$ git commit -m "New Repository"
[master (root-commit) 433c9c8] New Repository
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory.
1 file changed, 1 insertion(+)
create mode 100644 README.md

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (master)
$ git remote add origin http://github.com/dmitrii724/NewRepository.git

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (master)
$ git push -u origin master
Counting objects: 3, done.
Writing objects: 100% (3/3), 241 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To http://github.com/dmitrii724/NewRepository.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (master)
$
```

```
MINGW64:/d/Univer/Anul 2/Semestru 2/NewRepository
user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (master)
$ git config --global user.name "dmitrii724"

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (master)
$ git config --global user.email dmitrii122988@gmail.com

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (master)
$
```

Am creat doua branch-uri cu numele 1 si 2 folosind comanda git branch "numele".

```
MINGW64:/d/Univer/Anul 2/Semestru 2/NewRepository
user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (master)
$ git branch
* master

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (master)
$ git branch 1
1
* master

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (master)
$ git branch 2

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (master)
$ |
```

Am adaugat un fisier pe branch-ul 1.

```
MINGW64:/d/Univer/Anul 2/Semestru 2/NewRepository
user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (master)
$ git checkout 1
Switched to branch '1'

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (1)
$ notepad new.txt

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (1)
$ ls
HelloWorld.java new.txt README.md

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (1)
$ git add .

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (1)
$ git commit -m "First Branch"
[1 ea383d8] First Branch
2 files changed, 10 insertions(+)
create mode 100644 HelloWorld.java
create mode 100644 new.txt
```

Am adaugat si un fisier pe branch-ul 2.

```
MINGW64:/d/Univer/Anul 2/Semestru 2/NewRepository
user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (1)
$ git checkout 2
Switched to branch '2'

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (2)
$ notepad example.txt

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (2)
$ git add example.txt

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (2)
$ git add HelloWorld.cpp

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (2)
$ git commit -m "Second branch"
> git commit -m "Second branch"
> ^C

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (2)
$ git commit -m "Second Branch"
[2 7307d4d] Second Branch
2 files changed, 8 insertions(+)
create mode 100644 HelloWorld.cpp
create mode 100644 example.txt

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (2)
$ git push origin 2
Counting objects: 4, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (3/3), done.
```

Cind accesam github.com ca master putem accepta schimbarile de pe celelalte branch-uri astfel fisierele vor fi adaugate pe master.La fel putem lasa si un comentariu pentru acel commit.

```
MINGW64:/d/Univer/Anul 2/Semestru 2/NewRepository
user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (2)
$ git checkout 1
Switched to branch '1'

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (1)
$ git push origin 1
Counting objects: 4, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 460 bytes | 0 bytes/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To http://github.com/dmitrii724/NewRepository.git
 * [new branch]      1 -> 1

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (1)
$
```

Am setat branch-ul 1 track a remote.


```
MINGW64:/d/Univer/Anul 2/Semestru 2/NewRepository
user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (1)
$ notepad empty.txt

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (1)
$ git status
On branch 1
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        empty.txt

nothing added to commit but untracked files present (use "git add" to track)
user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (1)
$ git add .

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (1)
$ git commit -m "Empty txt"
[1 a0b096c] Empty txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 empty.txt

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (1)
$ git push --set-upstream origin 1
Counting objects: 3, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 264 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To http://github.com/dmitrii724/NewRepository.git
    ea383d8..a0b096c 1 -> 1
Branch 1 set up to track remote branch 1 from origin.

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (1)
$
```

Am resetat branch-ul 1 la un commit anterior.

```
MINGW64:/d/Univer/Anul 2/Semestru 2/NewRepository
user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (1)
$ git log --oneline
a0b096c Empty txt
ea383d8 First Branch
433c9c8 New Repository

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (1)
$ git status
On branch 1
Your branch is up-to-date with 'origin/1'.
nothing to commit, working directory clean

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (1)
$ git reset --hard ea383d8
HEAD is now at ea383d8 First Branch

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (1)
$
```

Am facut merge la branch-ul 1 cu master.

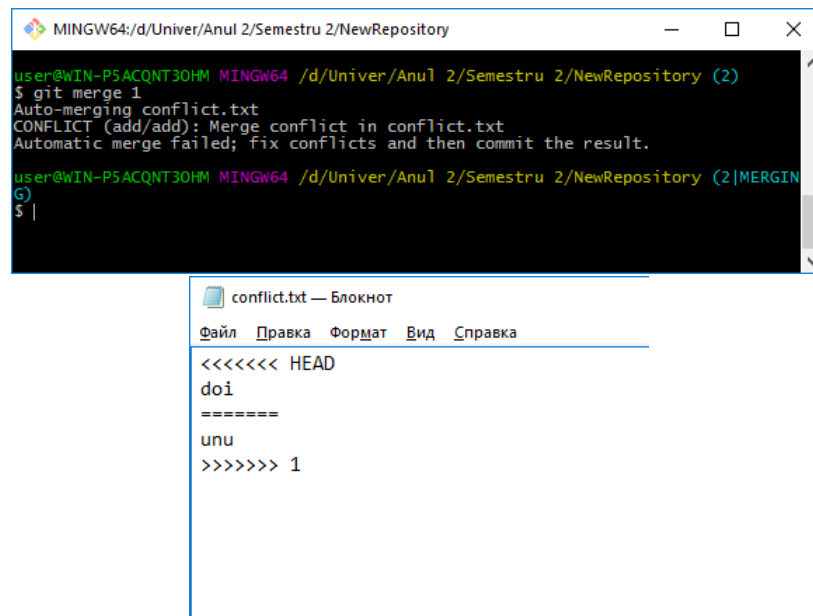
```
MINGW64:/d/Univer/Anul 2/Semestru 2/NewRepository
user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (1)
$ git status
On branch 1
Your branch is up-to-date with 'origin/1'.
nothing to commit, working directory clean

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (1)
$ git checkout master
Switched to branch 'master'
Your branch is up-to-date with 'origin/master'.

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (master)
$ git merge 1
Updating 433c9c8..a0b096c
Fast-forward
 HelloWorld.java | 9 ++++++++
 empty.txt       | 0
 new.txt         | 1 +
 3 files changed, 10 insertions(+)
 create mode 100644 HelloWorld.java
 create mode 100644 empty.txt
 create mode 100644 new.txt

user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (master)
$
```

În cazul cînd pe un branch avem un fișier cu un conținut oarecare și pe al alt branch același fișier dar cu conținut diferit atunci cînd încercăm să facem merge a acestor două branch-uri atunci primim un mesaj de conflict. Dacă deschidem fișierul acolo vor fi afișate problemele care trebuie înlăturate.



```
MINGW64:/d/Univer/Anul 2/Semestru 2/NewRepository
user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (2)
$ git merge 1
Auto-merging conflict.txt
CONFLICT (add/add): Merge conflict in conflict.txt
Automatic merge failed; fix conflicts and then commit the result.
user@WIN-P5ACQNT30HM MINGW64 /d/Univer/Anul 2/Semestru 2/NewRepository (2)MERGIN
$ |
```

```
conflict.txt — Блокнот
Файл  Правка  Формат  Вид  Справка
<<<<<<< HEAD
doi
=====
unu
>>>>>>> 1
```

Pentru a rezolva această problemă putem modifica conținutul fișierului și după care facem din nou git add și commit astfel rezolvăm acest conflict.

Concluzie

În această lucrare de laborator am studiat Version Control System numit github.com. Github-ul oferă posibilitate de a ține proiectul online, care poate fi de tip public și privat. Am efectuat task-urile propuse precum ar fi compilare unor mici programe C++, Java de tipul HELLO WORLD, efectuare commiturilor, initializarea unui repository nou și altele. Pentru a efectua aceste operații am utilizat Git-Bash care este un terminal cu comenzi asemănătoare cu cel din Linux. Comenzile sunt simple și eficiente pentru a găzdui un proiect.