

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:

Comanda ARTUR

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

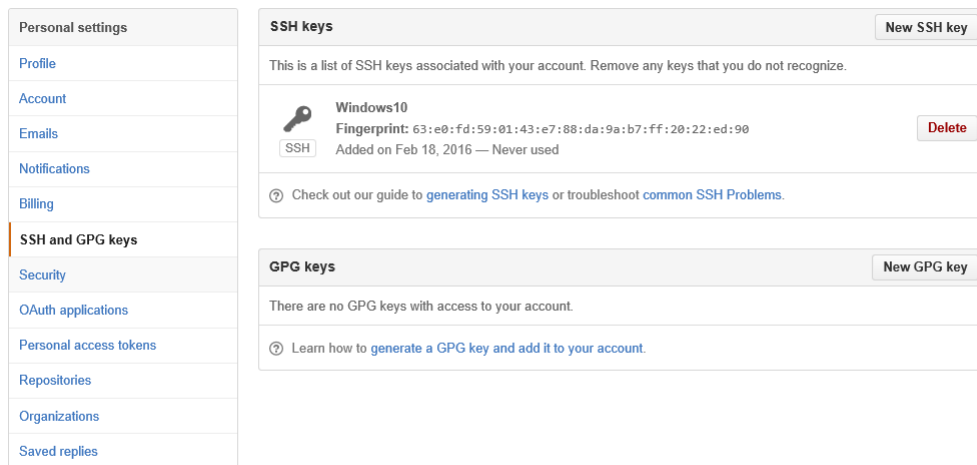
- Basic Level (nota 5 — 6) :
 - conecteaza-te la server folosind SSH
 - compileaza cel putin 2 sample programs din setul HelloWorldPrograms folosind CLI
 - executa primul commit folosind VCS
- Normal Level (nota 7 — 8):
 - initializeaza un nou repository
 - configureaza-ti VCS
 - crearea branch-urilor (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/aillyroredshi/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/MIDPS LAB 2 foto
ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto
$ g++ hello.cpp -o helloworld

ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto
$
ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto
$ ls
1.png hello.cpp hello.java helloworld.exe*
ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto
$ ./helloworld
Hello World!
ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto
$
```

```
MINGW64:/d/MIDPS LAB 2 foto
ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto
$ javac hello.java
hello.java:2: error: class HelloWorld is public, should be declared in a file na
med HelloWorld.java
public class HelloWorld {
    ^
1 error
ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto
$ javac HelloWorld.java
ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto
$ ls
1.png 2.png hello.cpp HelloWorld.class helloworld.exe* HelloWorld.java
ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto
$ java HelloWorld
Hello, World
ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto
$
```

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/MIDPS LAB 2 foto
ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (master)
$ git add example.txt

ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (master)
$ git commit -m "TXT"
[master 19cb8eb] TXT
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 example.txt

ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (master)
$ git push origin master
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 272 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/aillyroredshi/new-repository.git
d3a1fa8..19cb8eb master -> master

ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (master)
$
```

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

```
MINGW64:/d/MIDPS LAB 2 foto
ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto
$ echo "# new repository" >> README.md

ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto
$ git init
Initialized empty Git repository in D:/MIDPS LAB 2 foto/.git/

ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (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.

ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (master)
$ git commit -m "LAB 2"
[master (root-commit) d3a1fa8] LAB 2
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

ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (master)
$ git remote add origin https://github.com/aillyroredshi/new-repository.git

ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (master)
$ git push -u origin master
Counting objects: 3, done.
Writing objects: 100% (3/3), 226 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/aillyroredshi/new-repository.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
```

```
MINGW64:/d/MIDPS LAB 2 foto
--replace-all      replace all matching variables: name value [value_regex]
x] --add            add a new variable: name value
--unset            remove a variable: name [value-regex]
--unset-all        remove all matches: name [value-regex]
--rename-section    rename section: old-name new-name
--remove-section    remove a section: name
-l, --list          list all
-e, --edit          open an editor
--get-color         find the color configured: slot [default]
--get-colorbool     find the color setting: slot [stdout-is-tty]

Type
--bool             value is "true" or "false"
--int              value is decimal number
--bool-or-int      value is --bool or --int
--path             value is a path (file or directory name)

Other
-z, --null         terminate values with NUL byte
--name-only        show variable names only
--includes         respect include directives on lookup

ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (master)
$ git config --global user.name "aillyroredshi"

ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (master)
$ git config --global user.email comanda.artur@gmail.com

ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (master)
$
```

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

```
MINGW64:/d/MIDPS LAB 2 foto
ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (master)
$ git branch first

ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (master)
$ git branch second

ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (master)
$ git branch
 1
  first
* master
  second

ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (master)
$ |
```

Am adaugat un fisier pe branch-ul 1.

```
MINGW64:/d/MIDPS LAB 2 foto
ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (second)
$ notepad example.txt

ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (second)
$ git add example.txt

ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (second)
$ git add hello
hello.cpp      helloworld.exe

ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (second)
$ git add hello.cpp

ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (second)
$ git commit -m "branch 2"
[second a45bf96] branch 2
 2 files changed, 8 insertions(+)
 create mode 100644 hello.cpp

ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (second)
$ |
```

Am adaugat si un fisier pe branch-ul 2.

```
MINGW64:/d/MIDPS LAB 2 foto
ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (second)
$ git checkout first
Switched to branch 'first'

ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (first)
$ git push origin first
Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 503 bytes | 0 bytes/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/aillyroredshi/new-repository.git
 * [new branch]      first -> first

ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (first)
$ |
```

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/MIDPS LAB 2 foto
ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (second)
$ git commit -m "Branch 2"
On branch second
Untracked files:
  1.png
 10.png
 2.png
 3.png
 4.png
 5.png
 6.png
 7.png
 8.png
 9.png
 HelloWorld.class
 empty.txt
 helloworld.exe

nothing added to commit but untracked files present

ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (second)
$ git push origin second
Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 416 bytes | 0 bytes/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/aillyroredshi/new-repository.git
 * [new branch]      second -> second

ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (second)
$
```

Am setat branch-ul 1 track a remote.

```
MINGW64; d:/MIDPS LAB 2 foto
ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (second)
$ git log --oneline
a45bf96 branch 2
19cb8eb TXT
d3a1fa8 LAB 2

ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (second)
$ git status
On branch second
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        1.png
        10.png
        11.png
        2.png
        3.png
        4.png
        5.png
        6.png
        7.png
        8.png
        9.png
        HelloWorld.class
        empty.txt
        helloworld.exe

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

ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (second)
$ git reset --hard 19cb8eb
HEAD is now at 19cb8eb TXT
```

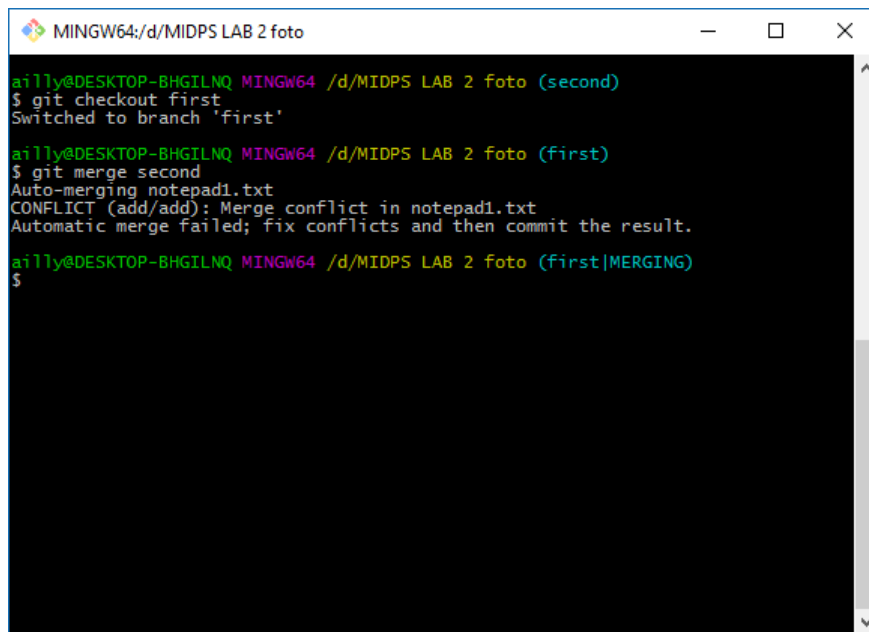
Am resetat branch-ul 1 la un commit anterior.

```
MINGW64; d:/MIDPS LAB 2 foto
ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (master)
$ git merge first
Updating 19cb8eb..f7f19a8
Fast-forward
 HelloWorld.java | 9 ++++++++
 new.txt         | 1 +
 2 files changed, 10 insertions(+)
 create mode 100644 HelloWorld.java
 create mode 100644 new.txt

ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (master)
$
```

Am facut merge la branch-ul 1 cu master.

In cazul cind pe un branch avem un fisier cu un continut oarecare si pe al branch acelasi fisier dar cu continut diferit atunci cind incercam sa facem merge a acestor doua branch-uri atunci primim un mesaj de conflict. Daca deschidem fisierul acolo vor fi afisate problemele care trebuie inlaturate.

A screenshot of a terminal window titled "MINGW64:/d/MIDPS LAB 2 foto". The terminal shows a user named "ailly@DESKTOP-BHGILNQ" in a "MINGW64" environment. The user is in the directory "/d/MIDPS LAB 2 foto" and is on the "second" branch. They run the command "\$ git checkout first", which switches them to the "first" branch. Then, they run "\$ git merge second", which attempts to merge the "second" branch into the "first" branch. The terminal shows "Auto-merging notepad1.txt" and then a "CONFLICT (add/add): Merge conflict in notepad1.txt". It also shows "Automatic merge failed; fix conflicts and then commit the result." The terminal ends with the prompt "\$" and the branch name "(first|MERGING)".

```
MINGW64:/d/MIDPS LAB 2 foto
ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (second)
$ git checkout first
Switched to branch 'first'

ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (first)
$ git merge second
Auto-merging notepad1.txt
CONFLICT (add/add): Merge conflict in notepad1.txt
Automatic merge failed; fix conflicts and then commit the result.

ailly@DESKTOP-BHGILNQ MINGW64 /d/MIDPS LAB 2 foto (first|MERGING)
$
```

Pentru a rezolva aceasta problema putem modifica continutul fisierului si dupa care faceem din nou git add si commit astfel rezolvam acest conflict.

Concluzie

În lucrarea de laborator am studiat sistemul github.com. Github-ul oferă posibilitate de a ține proiectul online. Am efectuat task-urile propuse, compilarea 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 pentru utilizare.