FACULTATEA CALCULATOARE, INFORMATICĂ ȘI MICROELECTRONICĂ

UNIVERSITATEA TEHNICĂ A MOLDOVEI

MEDII INTERACTIVE DE DEZVOLTARE A PRODUSELOR SOFT

LUCRARE DE LABORATOR #2

VERSION CONTROL SYSTEMS ȘI MODUL DE SETARE A UNUI SERVER

Autor:

st. gr. TI-141

Valeria BALINSCHI

lector asistent:

Irina COJANU

lector superior:

Svetlana COJOCARU

LUCRARE DE LABORATOR #2

1. Scopul lucrării

Însușirea noțiunii de Version Control Systems și a modului de setare a unui server.

2. Obiectivele lucrării

a) Înțelegerea și folosirea CLI (basic level)
b) Administrarea remote a mașinilor linux machine folosind SSH (remote code editing)
c) Version Control Systems (git || mercurial || svn)
d) Compilează codul C/C++/Java/Python prin intermediul CLI, folosind compilatoarele gcc/g++/javac/python

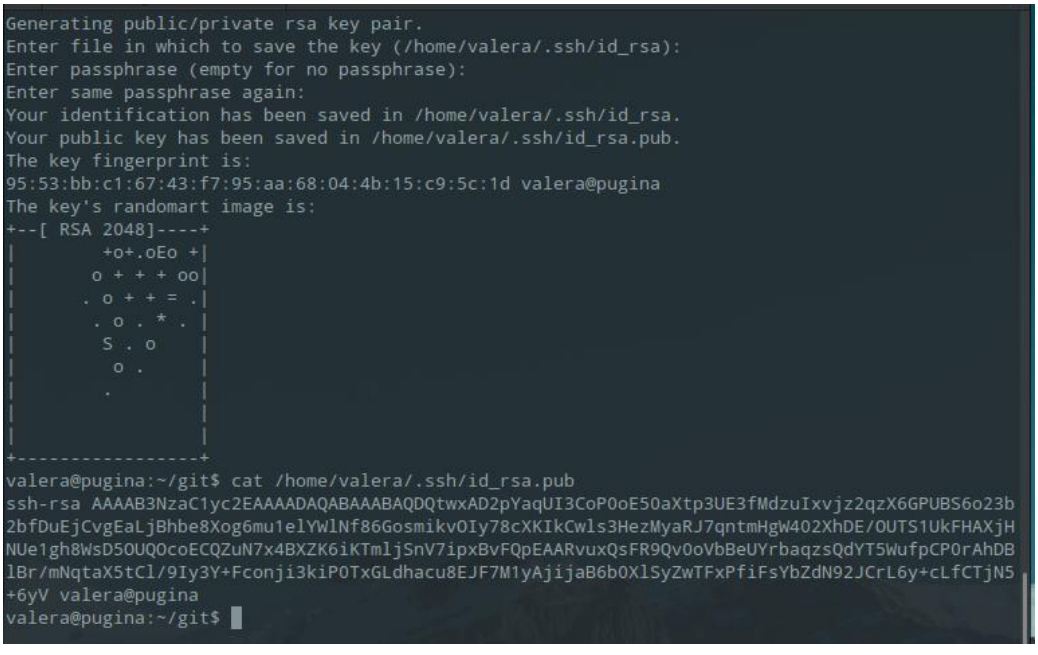
3. Efectuarea lucrării de laborator

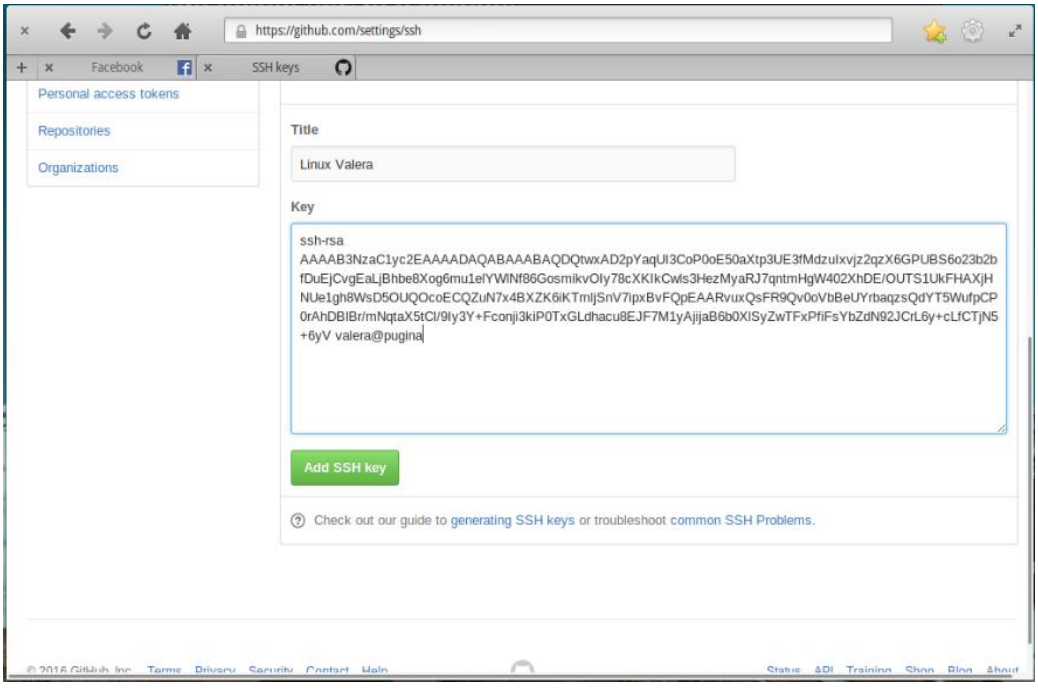
3.1. Task-uri implementate

- *Basic Level* (nota 5 || 6):
  - conectează-te la server folosind SSH
  - compilează cel puțin 2 sample programs din setul HelloWolrdPrograms folosind CLI
  - execută primul commit folosind VCS
- *Normal Level* (nota 7 || 8):
  - inițializează un nou repositoriu
  - configurează-ți VCS
  - crearea branch-urilor (creează cel puțin 2 branches)
  - commit pe ambele branch-uri (cel puțin 1 commit per branch)
- *Advanced Level* (nota 9 || 10):
  - setează un branch to track a remote origin pe care vei putea sa faci push (ex. Github, Bitbucket or custom server)
  - resetează un branch la commit-ul anterior
  - merge 2 branches
  - conflict solving between 2 branches
- *Bonus Point*:
  - Scrie un script care va compila HelloWolrdPrograms projects: c, cpp, java, pyton, ruby.

3.2. Realizarea lucrării de laborator

- *Basic Level* (nota 5 || 6) + *Bonus Point:*
  - conectează-te la server folosind SSH





- compilează cel puțin 2 sample programs din setul HelloWolrdPrograms folosind CLI

```c
#include <stdio.h>

int main()
{
   printf("Hello World!\n");
   return 0;
}
```

hello.c

```cpp
#include <iostream>

using namespace std;

int main()
{
    cout << "Hello World!" << endl;
    return 0;
}
```

hello.cpp

```
 GNU nano 2.2.6                                          File: hello.py

print "Hello World!"
```

hello.py

```
 GNU nano 2.2.6                                          File: hello.rb

puts "Hello World!"
```

hello.rb

```
 GNU nano 2.2.6                                          File: hello.java

class HelloWorld {
        public static void main(String args[]) {
                System.out.println("Hello World!");

}
```

hello.java

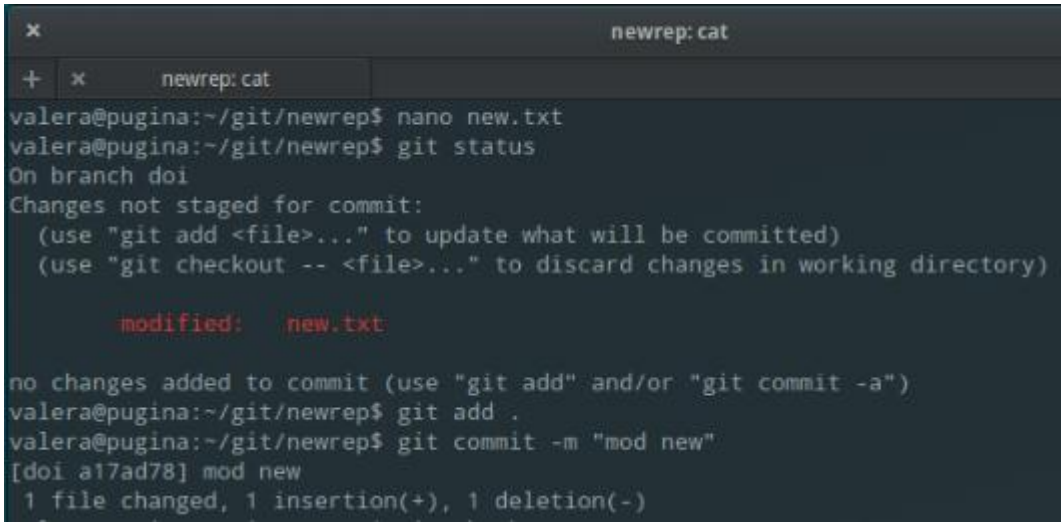
```
 GNU nano 2.2.6                                          File: helloscript.sh

#!/bin/bash

gcc -o helloc.out hello.c && ./helloc
g++ -o hellocpp.out hello.cpp &&  ./hellocpp
javac hello.java && java -cp /home/valera/git/lab2 HelloWorld
python hello.py
ruby hello.rb
```

helloscript.sh

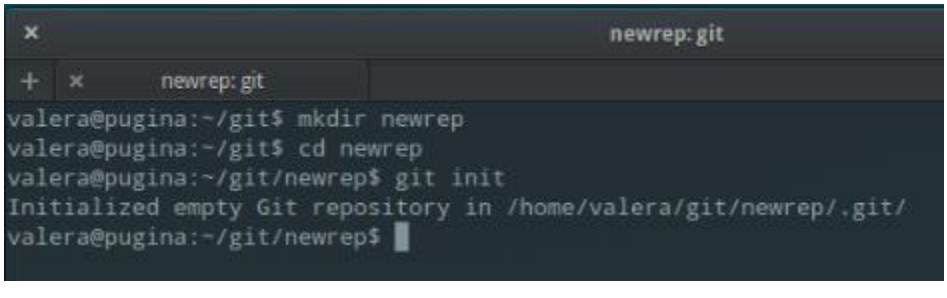
```
root@pugina:/home/valera/git/lab2# ./helloscript.sh
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
```

Rezultatele rulării programelor

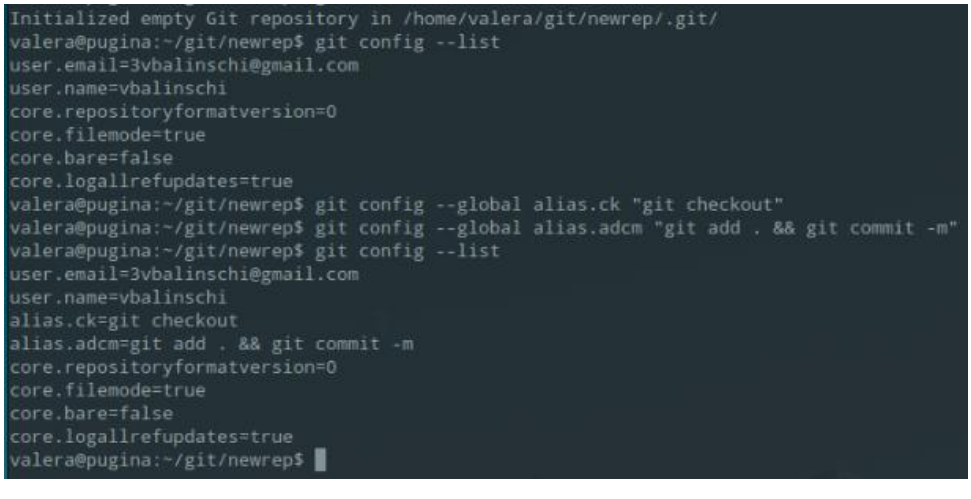
- execută primul commit folosind VCS



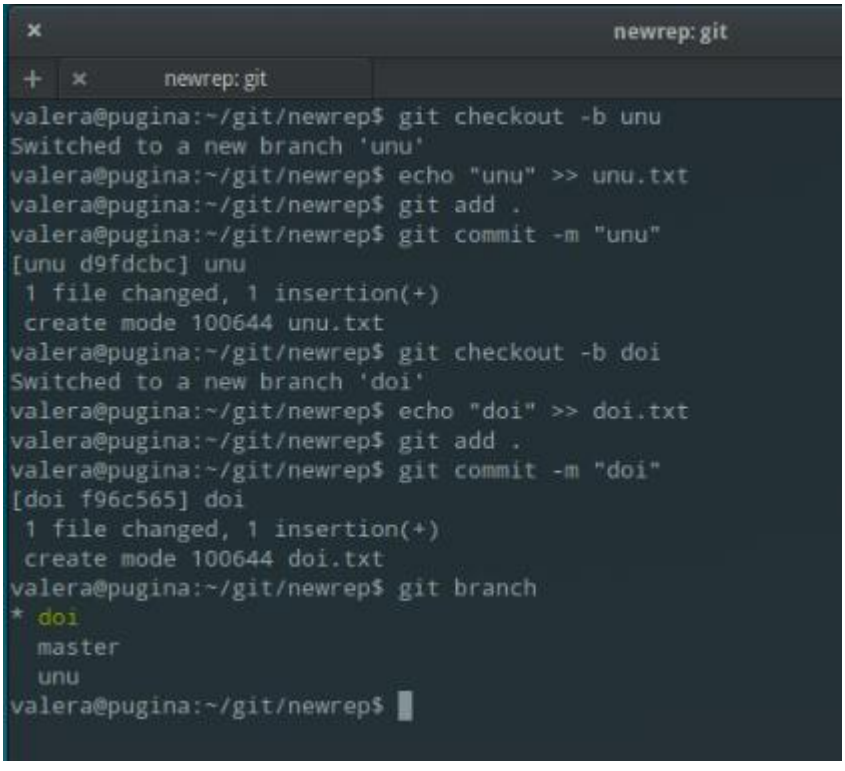
- *Normal Level* (nota 7 || 8):
  - inițializează un nou repositoriu



  - configurează-ți VCS

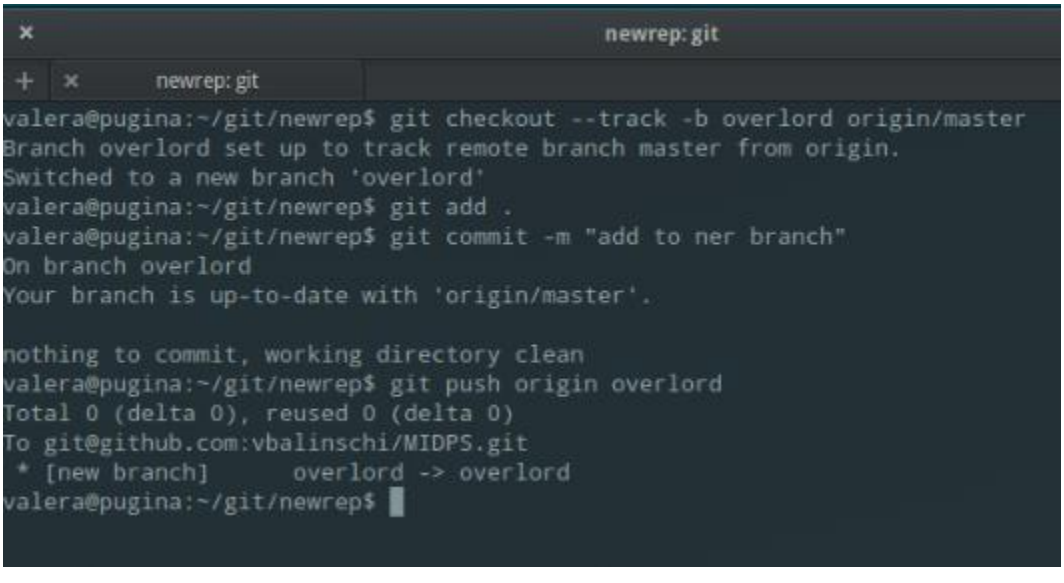


- crearea branch-urilor (creează cel puțin 2 branches)
- commit pe ambele branch-uri (cel puțin 1 commit per branch)

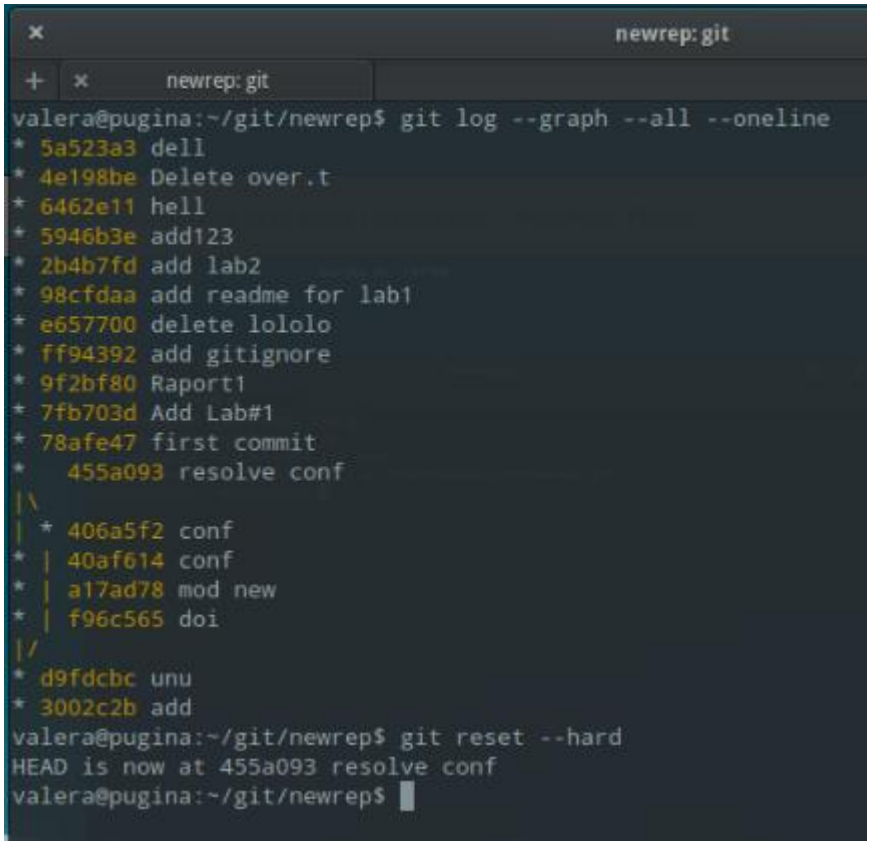


- *Advanced Level* (nota 9 || 10):

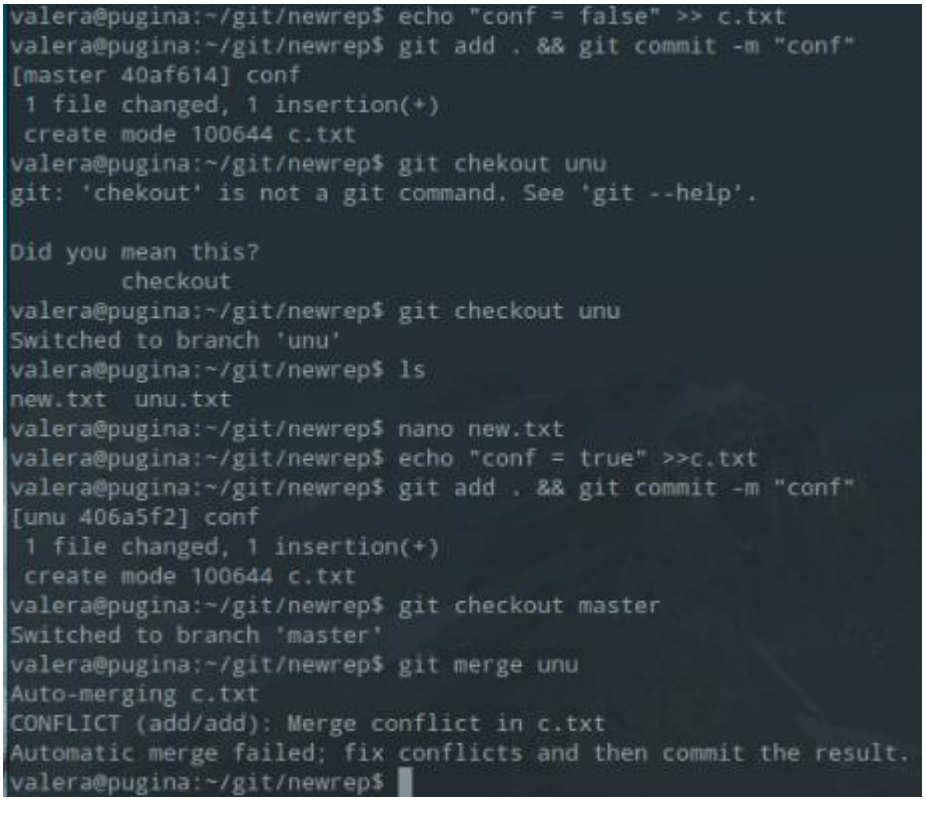
  - setează un branch to track a remote origin pe care vei putea sa faci push (ex. Github, Bitbucket or custom server)



- resetează un branch la commit-ul anterior



- merge 2 branches



- rezolvarea conflictelor a 2 branches

Concluzie

În urma realizării laboratorului nr.2 la tema: *"Version Control Systems si modul de setare a unui server"*, am însușit modul de utilizare a CLI, de administrarea remote a mașinilor linux machine folosind SSH.

Am efectuat conexiunea la un remote server folosind SSH (drept server remote, am folosit o mașină virtuală). În continuare compilând programele din lista dată și efectuând primul commit, folosind VCS.

Am creat 2 branch-uri asupra cărora am efectuat un commit din nou.

La fel am scris un script *"helloscript.sh"* ,prin intermediul căruia am compilat HelloWolrdPrograms projects din lista dată: c, cpp, java, pyton, ruby.

De asemenea am însușit instalarea corectă a Ubuntu pe o mașina virtuală, cît și bazele utilizării comenzilor din Linux.

În timpul efectuării laboratorului am lucrat cu comenzi ca :

`git init` – crearea unui repositoriu dintr-un fișier existent
`git remote add origin` – pentru interconectarea repositoriului local cu cel de pe github.
`git commit -m` – pentru înregistrarea unei schimbări(snapshot), ca `git add.`
`git config --global` – operațiuni cu fișierul de configurație GIT Bash
`git checkout` – pentru selectarea ramurii curente de lucru.
`git status` - informații despre starea fișierelor
`git mergetool` – ustensilă pentru soluționarea conflictelor ce le poate crea `git merge`
`git merge` – actualizarea schimbărilor de pe două sau mai multe ramuri.

În concluzie, am pus în practică VCS-ul GIT Bash creând un repositoriu și inițializându-l, clonând repositoriu și efectuând diverse comenzi în el.

Bibliografie

1. http://www.vogella.com/tutorials/Git/article.html
2. http://www.psychocats.net/ubuntu/virtualbox
3. http://www.manniwood.com/starting_a_project_with_git.html