Ministerul Educaţiei al Republicii Moldova

Universitatea Tehnică a Moldovei

Catedra: Tehnologii Informaționale

**RAPORT**

Lucrare de laborator Nr.2

*la Medii Interactive de Dezvoltare a Produselor Soft*

A efectuat: Pascari Ion

st.gr.TI-144

A verificat: Cojocaru Svetlana

dr.conf.univ.,

Chişinău 2016

**Lucrarea de laborator nr.2**

**Tema:**  **Version Control Systems si modul de setare a unui server**

**Scopul lucrării:**

* 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

**Formularea condiţiei problemei (sarcina de lucru):**

* *Basic Level* (nota 5 || 6) :
  + conecteaza-te la server folosind SSH
  + compileaza cel putin 2 sample programs din setul HelloWolrdPrograms folosind CLI
  + executa primul commit folosind VCS
* *Normal Level* (nota 7 || 8):
  + 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)
* *Advanced Level* (grade 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
* *Bonus Point*:
  + Scrie un script care va compila HelloWolrdPrograms projects.

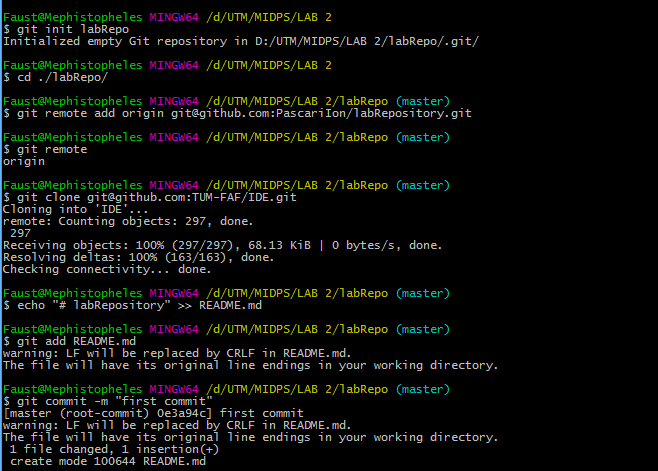
**Implementarea task-urilor :**

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

git init labRepo

cd ./labRepo/

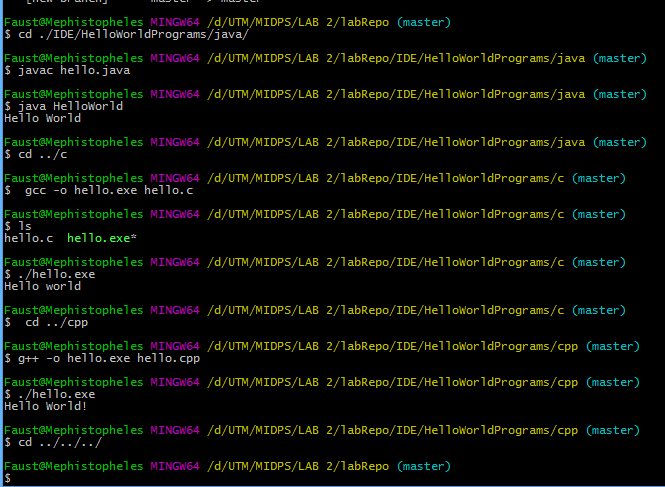
git remote add origin [git@github.com:PascariIon/labRepository.git](mailto:git@github.com:PascariIon/labRepository.git)



* + compileaza cel putin 2 sample programs din setul HelloWolrdPrograms folosind CLI

javac hello.java

gcc -o hello.exe hello.c



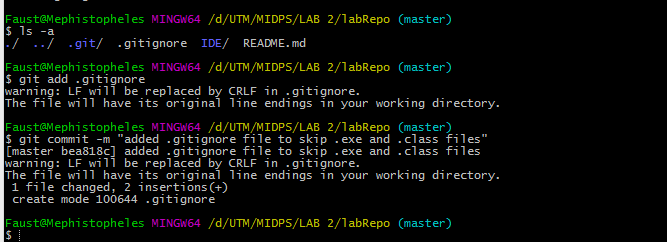
* + executa primul commit folosind VCS

vim .gitignore

ls -a

git add .gitignore

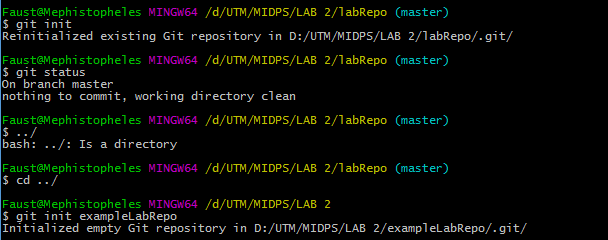
git commit -m "added .gitignore file to skip .exe and .class files"



* *Normal Level* (nota 7 || 8):
  + initializeaza un nou repositoriu

git init

git init exampleLabRepo

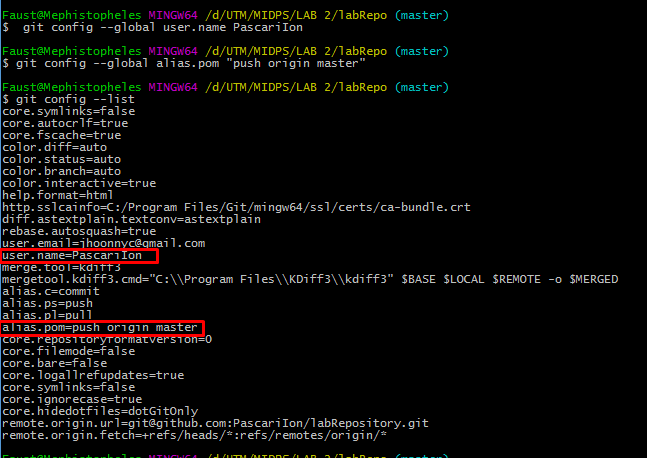


* + configureaza-ti VCS

git config --global user.name PascariIon

git config --global alias.c commit

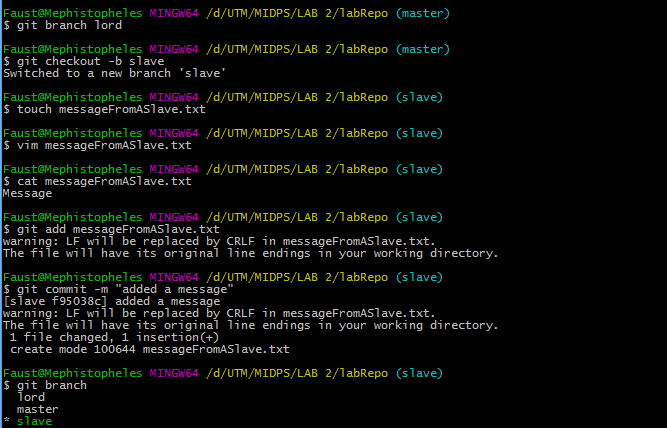
git config --global alias.pom push origin master



* + crearea branch-urilor (creeaza cel putin 2 branches)

git branch lord

git checkout -b slave



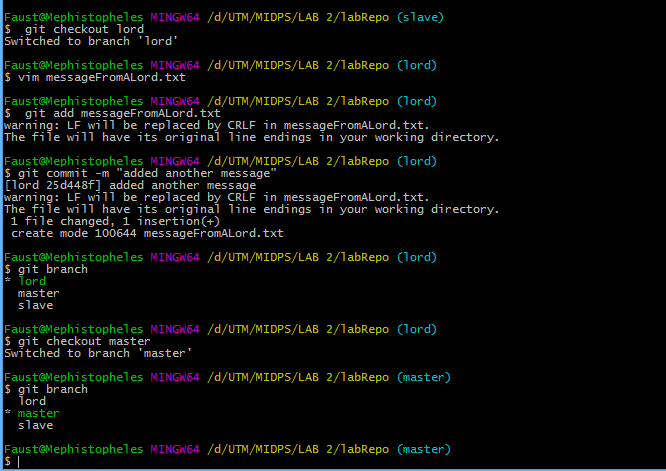
* + commit pe ambele branch-uri (cel putin 1 commit per branch)

git checkout lord

vim messageFromALord.txt

git add messageFromALord.txt

git commit -m "added another message"



* *Advanced Level* (grade 9 || 10):
  + seteaza un branch to track a remote origin pe care vei putea sa faci push (ex. Github, Bitbucket or custom server)

git checkout -b king origin/master

git status

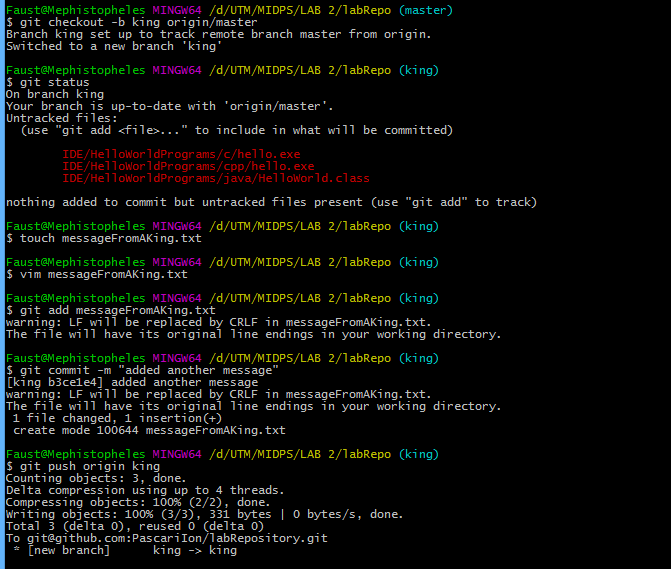
touch messageFromAKing.txt

vim messageFromAKing.txt

git add messageFromAKing.txt

git commit -m "added another message"

git push origin king



* + reseteaza un branch la commit-ul anterior

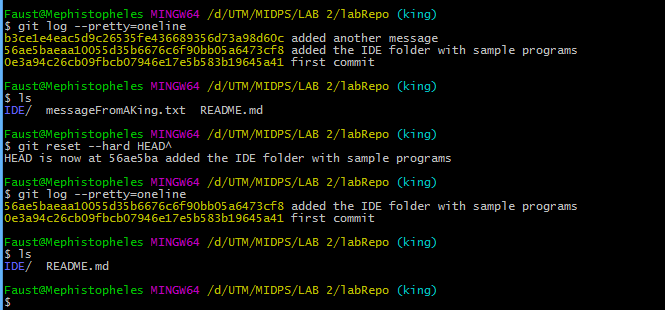
git log --pretty=oneline

ls

git reset --hard HEAD^

git log --pretty=oneline

ls



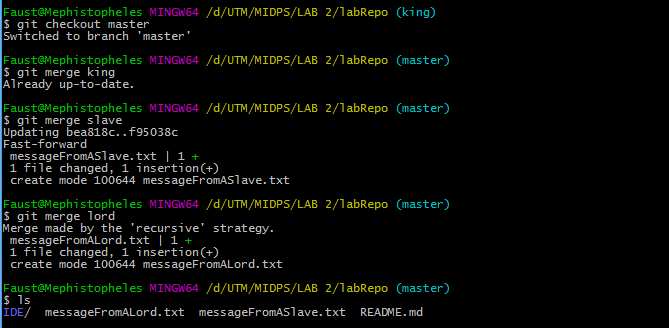
* + merge 2 branches

git checkout master

git merge king

git merge slave

git merge lord



* + conflict solving between 2 branches

vim messageFromALord.txt

git add messageFromALord.txt

git commit -m "modified messageFromALord -> 'is from a Lord'"

git checkout slave

vim messageFromALord.txt

git add messageFromALord.txt

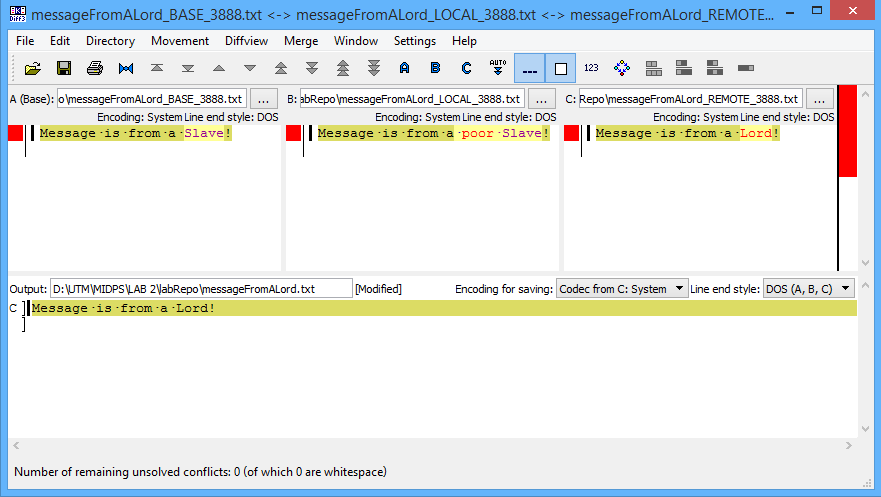
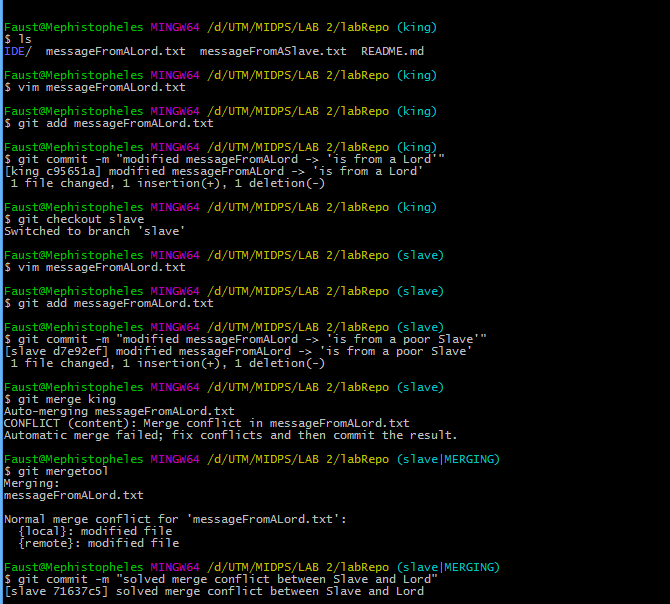
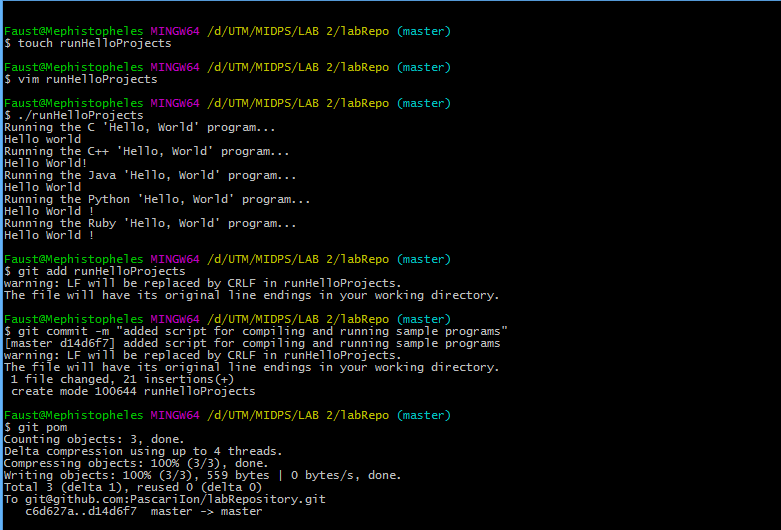
git commit -m "modified messageFromALord -> 'is from a poor Slave'"

git merge king

git mergetool

git commit -m "solved merge conflict between Slave and Lord"

git checkout master



* *Bonus Point*:
  + Scrie un script care va compila HelloWolrdPrograms projects.

#! /bin/bash

echo "Running the C 'Hello, World' program..."

gcc -Wall -std=c99 -g -o ./IDE/HelloWorldPrograms/c/hello.exe ./IDE/HelloWorldPrograms/c/hello.c

./IDE/HelloWorldPrograms/c/hello.exe

echo "Running the C++ 'Hello, World' program..."

g++ -Wall -std=c++11 -g -o ./IDE/HelloWorldPrograms/cpp/hello.exe ./IDE/HelloWorldPrograms/cpp/hello.cpp

./IDE/HelloWorldPrograms/cpp/hello.exe

echo "Running the Java 'Hello, World' program..."

javac ./IDE/HelloWorldPrograms/java/hello.java

java -cp ./IDE/HelloWorldPrograms/java/ HelloWorld

export PATH=/C/Python27:{$PATH}

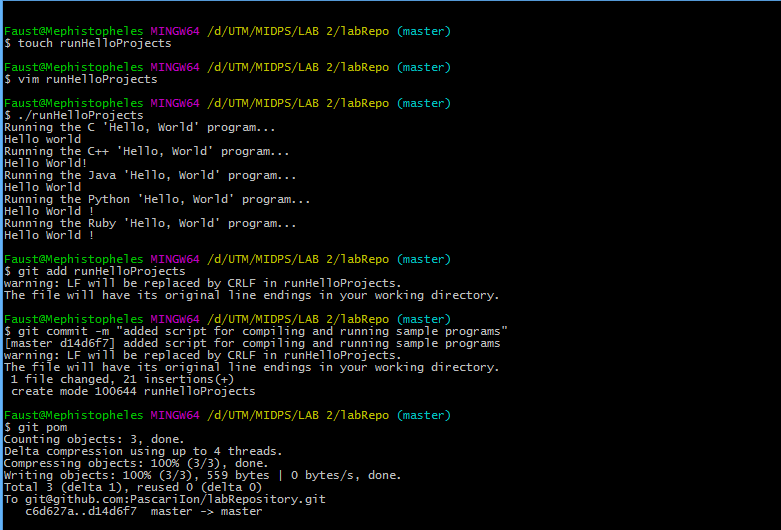
echo "Running the Python 'Hello, World' program..."

python ./IDE/HelloWorldPrograms/python/hello.py

export PATH=/C/Ruby22-x64/bin:{$PATH}

echo "Running the Ruby 'Hello, World' program..."

ruby ./IDE/HelloWorldPrograms/ruby/hello.rb



**Concluzii**

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

Am înțeles cum se crează un cont pe github.com și cum poate fi genereată o cheie SSH, pentru a putea efectua diverse comenzi (git push origin master) în GIT Bash fără introducerea username-ului și password-ului.

Am învățat cum să compilez și să rulez cod ce aparție diferitor limbaje de programare (C, C++, Java, Ruby, Python) prin intermediul CLI (command-line-interface) și prin intermediul scriptului scris în Shell.

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.

Și m-am învățat să setez un branch cu access la un repositoriu de pe github.

**Bibliografie**

* Git Reference , <http://gitref.org/branching/#branch>
* [GitHub Guides logo](https://guides.github.com/) , <https://guides.github.com/activities/hello-world/>
* Îndrumar metodic pentru lucrările de laborator la MIDPS