

Ministerul Educației al Republicii Moldova

Universitatea Tehnică a Moldovei
Catedra Tehnologii Informaționale

RAPORT

Lucrarea de laborator#2

la Medii Interactive de Dezvoltare a Produselor Soft

A efectuat:
st.gr. TI – 143

Profir Andrei

A verificat:
lect.asist.

Cojanu Irina

Chișinău 2016

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 condiției problemei (sarcina de lucru):

- *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* (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 HelloWorldPrograms projects.

Implimentare task-uri:

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

```
Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB/Lab-2 (master)
$ git clone git@github.com:TUM-FAF/IDE.git clone
Cloning into 'clone'...
remote: Counting objects: 297, done.
remote: Total 297 (delta 0), reused 0 (delta 0)g objects: 83% (247, pack-
k-/297)reused 297
Receiving objects: 100% (297/297), 68.13 KiB | 0 bytes/s, done.
Resolving deltas: 100% (163/163), done.
Checking connectivity... done.
```

- o compileaza cel putin 2 sample programs din setul HelloWorldPrograms folosind CLI

```
Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB/Lab-2 (master)
$ cd ./clone/HelloWorldPrograms/java
Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB/Lab-2/clone/HelloWorldProg
rams/java (master)
$ javac hello.java
Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB/Lab-2/clone/HelloWorldProg
rams/java (master)
$ java HelloWorld
Hello World
```


```
Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB/Lab-2/clone/HelloWorldProg
rams/c (master)
$ gcc -o hello.exe hello.c
Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB/Lab-2/clone/HelloWorldProg
rams/c (master)
$ ./hello.exe
Hello world
Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB/Lab-2/clone/HelloWorldProg
rams/c (master)
$ cd ../cpp
Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB/Lab-2/clone/HelloWorldProg
rams/cpp (master)
$ g++ -o hello.exe hello.cpp
Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB/Lab-2/clone/HelloWorldProg
rams/cpp (master)
$ ./hello.exe
Hello World!
```

- o executa primul commit folosind VCS

```
Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB (master)
$ cd ./Lab-2
Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB/Lab-2 (master)
$ echo "# Lab#2 - MIDPS" >> README.md
Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB/Lab-2 (master)
$ git add README.md
```

```
Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB/Lab-2 (master)
$ git commit -m "lab-2 first commit"
[master cdf03ef] lab-2 first commit
1 file changed, 2 insertions(+)
create mode 100644 Lab-2/README.md
```

- *Normal Level* (nota 7 || 8):
 - initializeaza un nou repository
 - configureaza-ti VCS



```

MINGW64:/d/Andrei/UTM-Anul II/MIDPS_LAB

Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB (master)
$ git init
Reinitialized existing Git repository in D:/Andrei/UTM-Anul II/MIDPS_LAB/.git/

Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB (master)
$ git config --global user.name "Profir Andrei"

Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB (master)
$ git config --global user.email trigun1994@inbox.ru

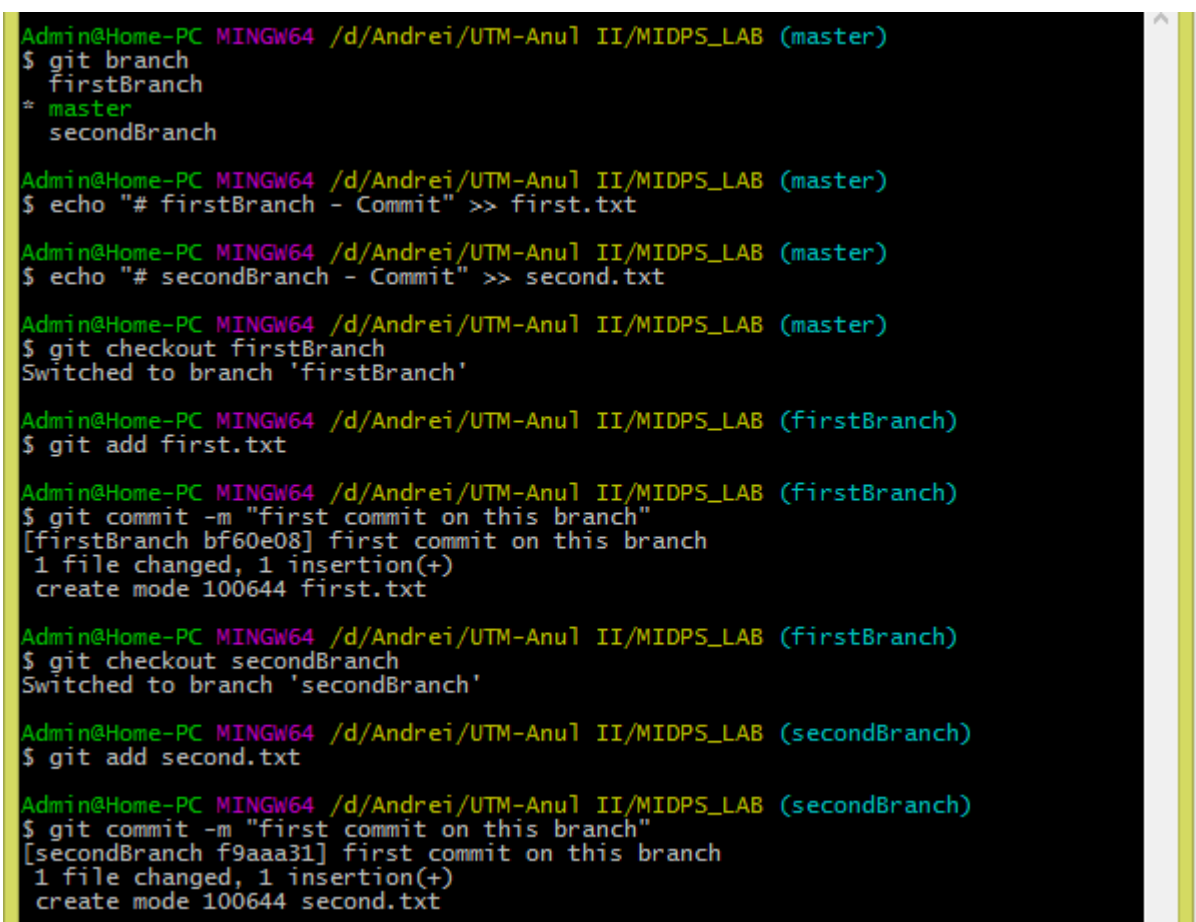
Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB (master)
$ git config user.name
Profir Andrei

Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB (master)
$ git config user.email
trigun1994@inbox.ru

Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB (master)
$ |

```

- crearea branch-urilor (creeaza cel putin 2 branches)
- commit pe ambele branch-uri (cel putin 1 commit per branch)



```

Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB (master)
$ git branch
  firstBranch
* master
  secondBranch

Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB (master)
$ echo "# firstBranch - Commit" >> first.txt

Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB (master)
$ echo "# secondBranch - Commit" >> second.txt

Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB (master)
$ git checkout firstBranch
Switched to branch 'firstBranch'

Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB (firstBranch)
$ git add first.txt

Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB (firstBranch)
$ git commit -m "first commit on this branch"
[firstBranch bf60e08] first commit on this branch
1 file changed, 1 insertion(+)
create mode 100644 first.txt

Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB (firstBranch)
$ git checkout secondBranch
Switched to branch 'secondBranch'

Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB (secondBranch)
$ git add second.txt

Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB (secondBranch)
$ git commit -m "first commit on this branch"
[secondBranch f9aaa31] first commit on this branch
1 file changed, 1 insertion(+)
create mode 100644 second.txt

```

- *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)

```
Admin@Home-PC MINGW64 /d/Andrei/UTM-Anu1 II/MIDPS_LAB (secondBranch)
$ git checkout -b track origin/master
Branch track set up to track remote branch master from origin.
Switched to a new branch 'track'

Admin@Home-PC MINGW64 /d/Andrei/UTM-Anu1 II/MIDPS_LAB (track)
$ echo "# track Branch - commit" >> track.txt

Admin@Home-PC MINGW64 /d/Andrei/UTM-Anu1 II/MIDPS_LAB (track)
$ git add .track.txt
fatal: pathspec '.track.txt' did not match any files

Admin@Home-PC MINGW64 /d/Andrei/UTM-Anu1 II/MIDPS_LAB (track)
$ git add track.txt

Admin@Home-PC MINGW64 /d/Andrei/UTM-Anu1 II/MIDPS_LAB (track)
$ git commit -m "commit"
[track 97c4511] commit
1 file changed, 1 insertion(+)
create mode 100644 track.txt

Admin@Home-PC MINGW64 /d/Andrei/UTM-Anu1 II/MIDPS_LAB (track)
$ git push origin track
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 294 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/ProfirAndrei/MIDPS.git
 * [new branch]      track -> track
```

- reseteaza un branch la commit-ul anterior

```
Admin@Home-PC MINGW64 /d/Andrei/UTM-Anu1 II/MIDPS_LAB (track)
$ git log -2
commit 97c4511bc41f95cb8cb60423a77513b6ad58d236
Author: Profir Andrei <trigun1994@inbox.ru>
Date:   Wed Mar 16 21:58:42 2016 +0200

    commit

commit ce88dbfd695342c6b2b12aafed236bd48d7fb211
Author: Profir Andrei <trigun1994@inbox.ru>
Date:   Wed Mar 16 20:45:45 2016 +0200

    the file readme.md was deleted

Admin@Home-PC MINGW64 /d/Andrei/UTM-Anu1 II/MIDPS_LAB (track)
$ git reset --hard HEAD^
HEAD is now at ce88dbf the file readme.md was deleted
```

- merge 2 branches

```
Admin@Home-PC MINGW64 /d/Andrei/UTM-Anu1 II/MIDPS_LAB (track)
$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)

Admin@Home-PC MINGW64 /d/Andrei/UTM-Anu1 II/MIDPS_LAB (master)
$ git merge firstBranch
Updating cdf03ef..bf60e08
Fast-forward
 first.txt | 1 +
1 file changed, 1 insertion(+)
create mode 100644 first.txt
```

```
Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB (master)
$ git merge secondBranch
Merge made by the 'recursive' strategy.
second.txt | 1 +
1 file changed, 1 insertion(+)
create mode 100644 second.txt
```

- conflict solving between 2 branches

```
Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB (master)
$ echo "# hello, master Branch" >> hi.txt

Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB (master)
$ git add hi.txt

Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB (master)
$ git commit -m "added hi.txt"
[master 37759e4] added hi.txt
1 file changed, 1 insertion(+)
create mode 100644 hi.txt

Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB (master)
$ git checkout firstBranch
Switched to branch 'firstBranch'

Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB (firstBranch)
$ echo "# KyKy :), firstBranch" >> hi.txt

Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB (firstBranch)
$ git add hi.txt

Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB (firstBranch)
$ git commit -m "added hi.txt on the firstBranch"
[firstBranch 202e923] added hi.txt on the firstBranch
1 file changed, 1 insertion(+)
create mode 100644 hi.txt

Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB (firstBranch)
$ git merge master
Auto-merging hi.txt
CONFLICT (add/add): Merge conflict in hi.txt
Automatic merge failed; fix conflicts and then commit the result.

Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB (firstBranch|MERGING)
$ git mergetool

This message is displayed because 'merge.tool' is not configured.
See 'git mergetool --tool-help' or 'git help config' for more details.
'git mergetool' will now attempt to use one of the following tools:
opendiff kdiff3 tkdiff xxdiff meld tortoisemerge gvimdiff diffuse diffmerge ecme
rge p4merge araxis bc codecompare emerge vimdiff
Merging:
hi.txt

Normal merge conflict for 'hi.txt':
{local}: created file
{remote}: created file
Hit return to start merge resolution tool (vimdiff):
3 files to edit

Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB (firstBranch|MERGING)
$ git commit -m "solved merge conflict"
[firstBranch f59f443] solved merge conflict
```

```

# KyKy :), firstBranch
<<<<<<< HEAD
# KyKy :), firstBranch
=====
# hello, master Branch
>>>>>> master

```

- *Bonus Point:*
 - Scrie un script care va compila HelloWorldPrograms projects.

```
#!/bin/bash

cd ./clone/HelloWorldPrograms

echo "<-----C program...----->"
gcc -o ./c/hello.exe ./c/hello.c
./c/hello.exe

echo "<-----Cpp program...----->"
g++ -o ./cpp/hello.exe ./cpp/hello.cpp
./cpp/hello.exe

echo "<-----Java program...----->"
javac ./java/hello.java
java -cp ./java HelloWorld

echo "<-----Python program...----->"
python ./python/hello.py

echo "<-----Ruby program...----->"
ruby ./ruby/hello.rb

~
<Andrei/UTM-Anul II/MIDPS_LAB/Lab-2/script.sh [unix] (08:05 17/03/2016)21,20 All
"./script.sh" [unix] 21L, 421C
```

```
Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB/Lab-2 (master)
$ vim ./script.sh

Admin@Home-PC MINGW64 /d/Andrei/UTM-Anul II/MIDPS_LAB/Lab-2 (master)
$ ./script.sh
<-----C program...----->
Hello world
<-----Cpp program...----->
Hello World!
<-----Java program...----->
Hello World
<-----Python program...----->
Hello World !
<-----Ruby program...----->
Hello World !
```

Link-ul catre repozitoriu online:

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

Concluzie:

În urma efectuării acestei lucrări de laborator au fost capătate primele deprinderi de bază în lucrul cu un VCS. În această lucrare de laborator a fost folosit GIT ca fiind unul dintre cele mai populare DVCS și unul dintre cele mai comode în lucru.

Am capatat deprinderi în lucrul cu repozitoriul atât online cât și local. M-am învățat să initializez un repozitoriu să-l clonez ș.a. Pe lângă aceste două manipulări cu repozitoriul am folosit și alte posibilități așa cum: crearea unui nou branch, am făcut unirea a două branch-uri, am învățat cum să hotărîm unele conflicte care pot apărea cînd facem “merge” la branch-uri.

De asemenea am învățat cum să compilez cod sursă în diferite limbaje atât direct din CLI cât și prin intermediul unui script creat anterior. Evident că prin intermediul scriptului este mult mai ușor din cauza că scriptul o dată creat nu necesită o multime de comenzi pentru a face ceea ce de fapt a fost făcut o dată.