

Ministerul Educației al Republicii Moldova  
Universitatea Tehnică a Moldovei

# RAPORT

Lucrarea de laborator nr. 2

Medii Interactive de dezvoltare a produselor software

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

A efectuat:

st. gr. TI-142

Donca Diana

A verificat:

lect. univ.

Cojan Irina

Chișinău 2016

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

**Scopul lucrării:** Studierea bazelor lucrului cu VCS. Obiectivele lucrării:

1. Înțelegerea și folosirea CLI (basic level)
2. Administrarea remote a mașinilor linux machine folosind SSH (remote code editing)
3. Version Control Systems (git || mercurial || svn)
4. Compileaza codul C/C++/Java/Python prin intermediul CLI, folosind compilatoarele gcc/g++/javac/python

### **Modul de lucru:**

#### **Pași lucrării:**

- conecteaza-te la server folosind SSH
- compileaza cel puțin 2 sample programs din setul HelloWorldPrograms folosind CLI
- executa primul commit folosind VCS
- initializeaza un nou repository
- configureaza-ti VCS
- crearea branch-urilor (creeaza cel puțin 2 branches)
- commit pe ambele branch-uri (cel puțin 1 commit per branch)
- 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

#### **Efectuarea Lucrării:**

- ✓ A fost creat un repository pe github.com.
- ✓ A fost stabilită conexiunea cu serverul prin generarea keygen-ului SSH prin instrucțiunea ssh-keygen, ea fiind adăugată în setări.
- ✓ A fost creat un fișier și inițializat git-ul prin instrucțiunea git init.
- ✓ A fost introdus .gitignore și README.md prin instrucțiunea git add, git commit și git push.
- ✓ Au fost create două programe, una în C și alta în C++, cu extensia respectivă .c și .cpp
- ✓ Cu ajutorul la Command Prompt, ele au fost compilate cu instrucțiunea:

gcc hwC.c -o hwC și g++ hwCPP.cpp -o hwCPP .

- ✓ Respectiv, au fost executate fișierele noi cu extensia .exe în Command Prompt, apelând la ele prin: ./hwC și ./hwCPP.
- ✓ A fost creat un branch nou, unde au fost încărcate fișierele compilate în CLI, împreună cu imaginile respective și fișierele originale .c și .cpp.
- ✓ Am făcut merge dintre branch-ul master și BBbranch, unde se aflau fișierele compilate.
- ✓ Am creat o situație de conflict, creînd un fișier Conflict.txt și încărcîndu-l pe branch master.
- ✓ Am creat un branch nou ConflictBranch, în care am modificat conținutul fișierului Conflict.txt, astfel făcînd opțiunea merge imposibilă.
- ✓ Am rezolvat conflictul și am făcut merge între branch-uri.

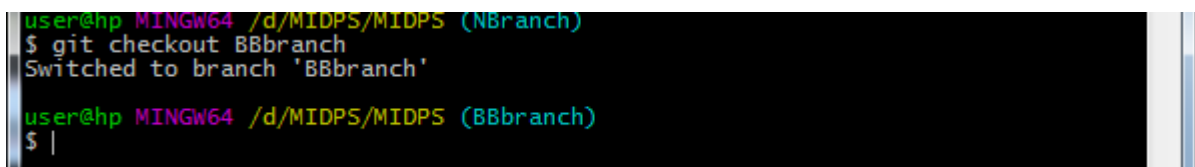
### Screen-uri ale executarii comenzilor:

- Crearea unui nou Branch:



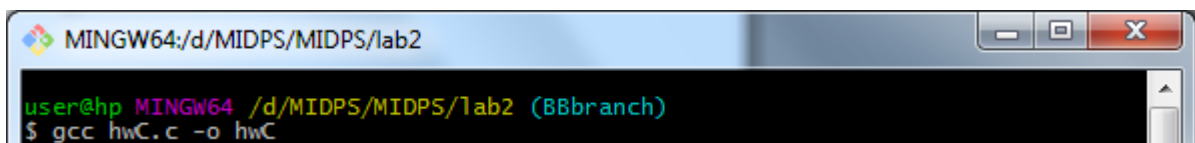
```
MINGW64:/d/MIDPS/MIDPS
user@hp MINGW64 /d/MIDPS/MIDPS (NBranch)
$ git branch BBbranch
```

- Trasnferul la Branch-ul nou:

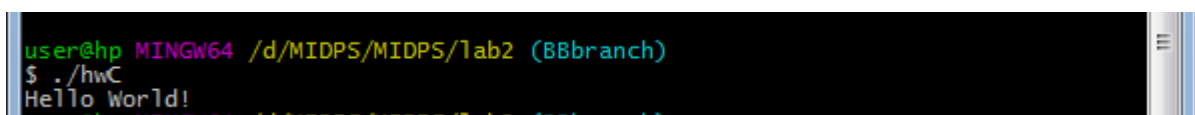


```
user@hp MINGW64 /d/MIDPS/MIDPS (NBranch)
$ git checkout BBbranch
Switched to branch 'BBbranch'
user@hp MINGW64 /d/MIDPS/MIDPS (BBbranch)
$ |
```

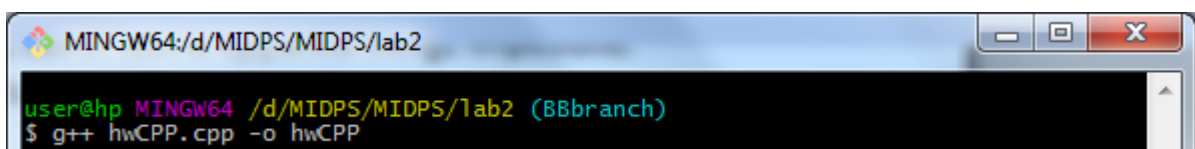
- Crearea si executarea Hello\_C.c și Hello\_C++.cpp:



```
MINGW64:/d/MIDPS/MIDPS/lab2
user@hp MINGW64 /d/MIDPS/MIDPS/lab2 (BBbranch)
$ gcc hwC.c -o hwC
```



```
user@hp MINGW64 /d/MIDPS/MIDPS/lab2 (BBbranch)
$ ./hwC
Hello World!
```



```
MINGW64:/d/MIDPS/MIDPS/lab2
user@hp MINGW64 /d/MIDPS/MIDPS/lab2 (BBbranch)
$ g++ hwCPP.cpp -o hwCPP
```

```
user@hp MINGW64 /d/MIDPS/MIDPS/lab2 (BBbranch)
$ ./hwCPP
Hello world!
```

- Adaugarea unui fisier .txt, care v-a fi resetat:

```
MINGW64:/d/MIDPS/MIDPS
user@hp MINGW64 /d/MIDPS/MIDPS (BBbranch)
$ git status
On branch BBbranch
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        lab2.txt

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

```
MINGW64:/d/MIDPS/MIDPS
user@hp MINGW64 /d/MIDPS/MIDPS (BBbranch)
$ git add .

user@hp MINGW64 /d/MIDPS/MIDPS (BBbranch)
$ git commit -m "new file"
[BBbranch 8d6f970] new file
Committer: Diana <Diana>
```

- Resetarea la commit-ul precedent:

```
user@hp MINGW64 /d/MIDPS/MIDPS (BBbranch)
$ git reset --hard
HEAD is now at 8d6f970 new file

user@hp MINGW64 /d/MIDPS/MIDPS (BBbranch)
$ |
```

- Merge între două branch-uri:

```
$ git pull
Updating 45b603c..3fc8c3b
Fast-forward
 lab 2/branch.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 lab 2/branch.txt
```

- Afișarea situației de conflict:

```
$ git merge MBranch
CONFLICT (rename/delete): Lab#2/Used_Files/Branch1.txt deleted in
MBranch and renamed in HEAD. Version HEAD of Lab#2/Used_Files/Br
anch1.txt left in tree.
Automatic merge failed; fix conflicts and then commit the result.
```

---

## Concluzie

În urma efectuării acestei lucrări de laborator au fost obținute deprinderi practice de lucru cu git – unul din cele mai populare DVCS. Am ajuns la concluzia că un VCS este un instrument indispensabil lucrului în echipă. Acest VCS este un instrument ce verifică versiunea proiectului, permițându-ne să evităm conflicte. Am observat că în git, lucru cu branch-urile se face cu mult mai simplu decât în alte VSC-uri cum ar fi SVN. Un lucru plăcut despre git este viteza și eficiența lui. Repozitoriile git-ului ocupă mai puțin loc decât cele ale lui SVN, și că git este distribuit, deci nu este nevoie de a fi mereu conectat la rețea.