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

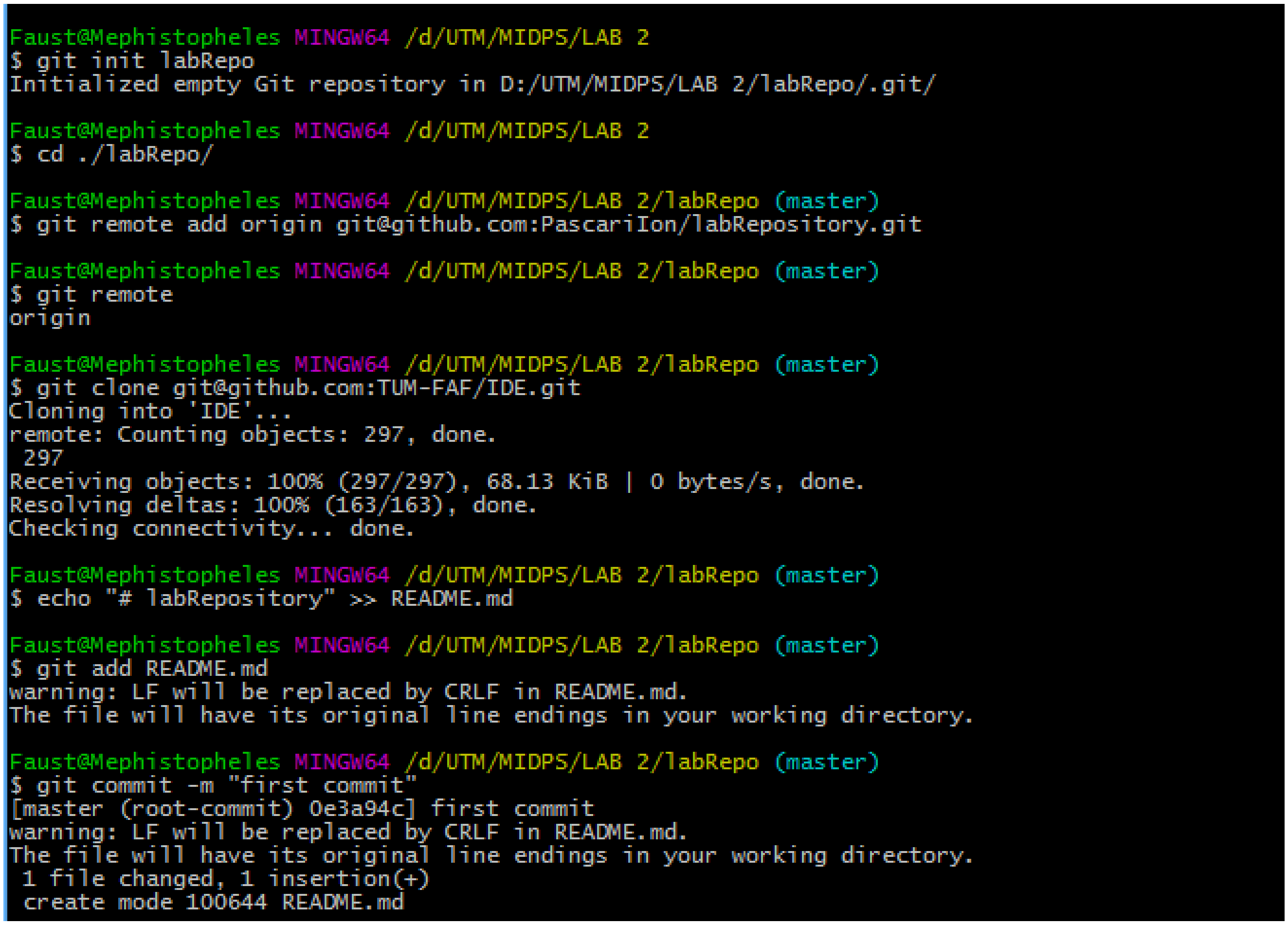
Formularea condiţiei problemei (sarcina de lucru):

- *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

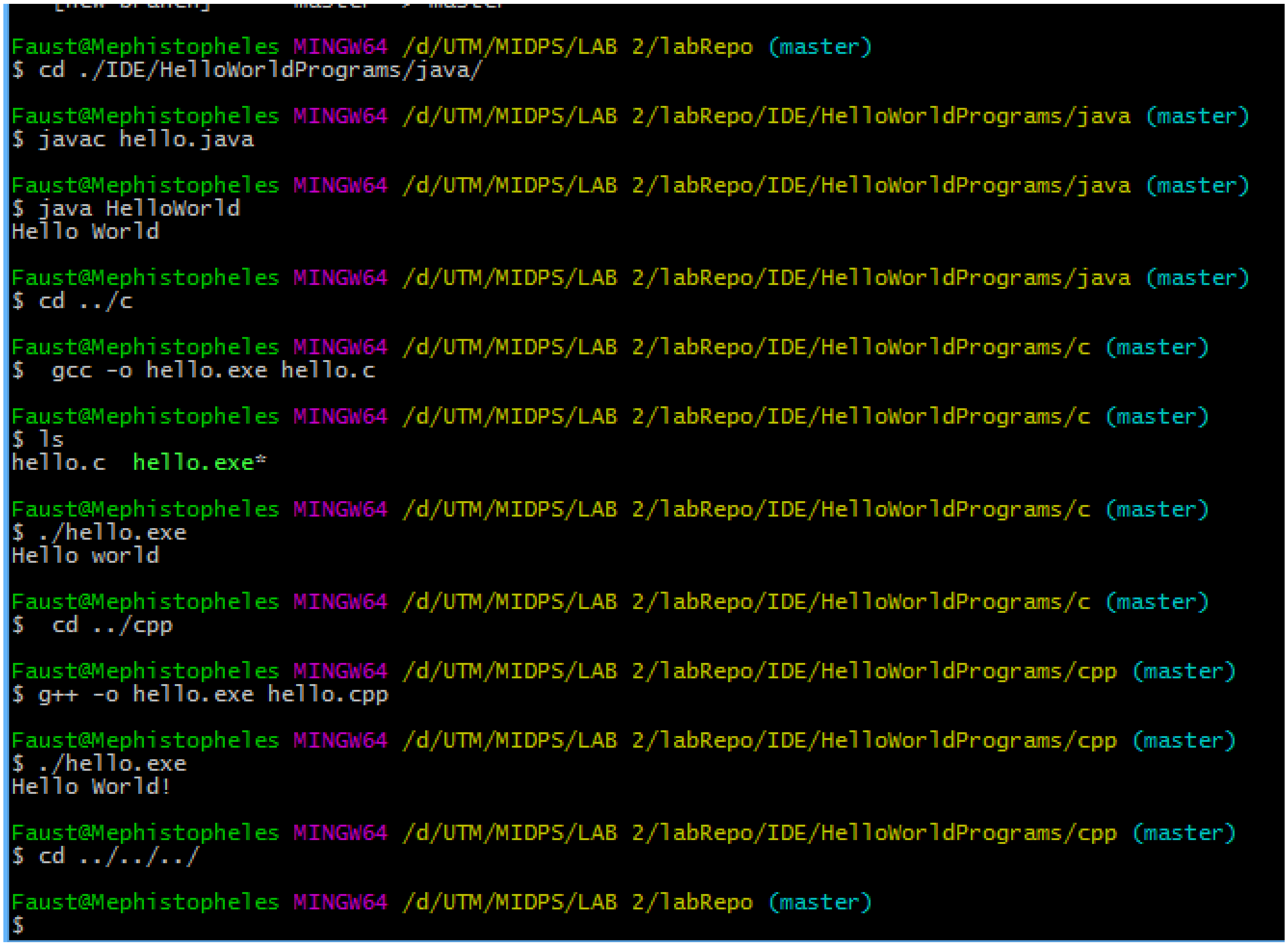
```
git init labRepo
cd ./labRepo/
git remote add origin git@github.com:PascariIon/labRepository.git
```



```
Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2
$ git init labRepo
Initialized empty Git repository in D:/UTM/MIDPS/LAB 2/labRepo/.git/

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2
$ cd ./labRepo/

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ git remote add origin git@github.com:PascariIon/labRepository.git

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ git remote
origin

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ git clone git@github.com:TUM-FAF/IDE.git
Cloning into 'IDE'...
remote: Counting objects: 297, done.
 297
Receiving objects: 100% (297/297), 68.13 KiB | 0 bytes/s, done.
Resolving deltas: 100% (163/163), done.
Checking connectivity... done.

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ echo "# labRepository" >> README.md

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (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.

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ git commit -m "first commit"
[master (root-commit) 0e3a94c] first commit
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
```

- compileaza cel putin 2 sample programs din setul HelloWolrdPrograms folosind CLI

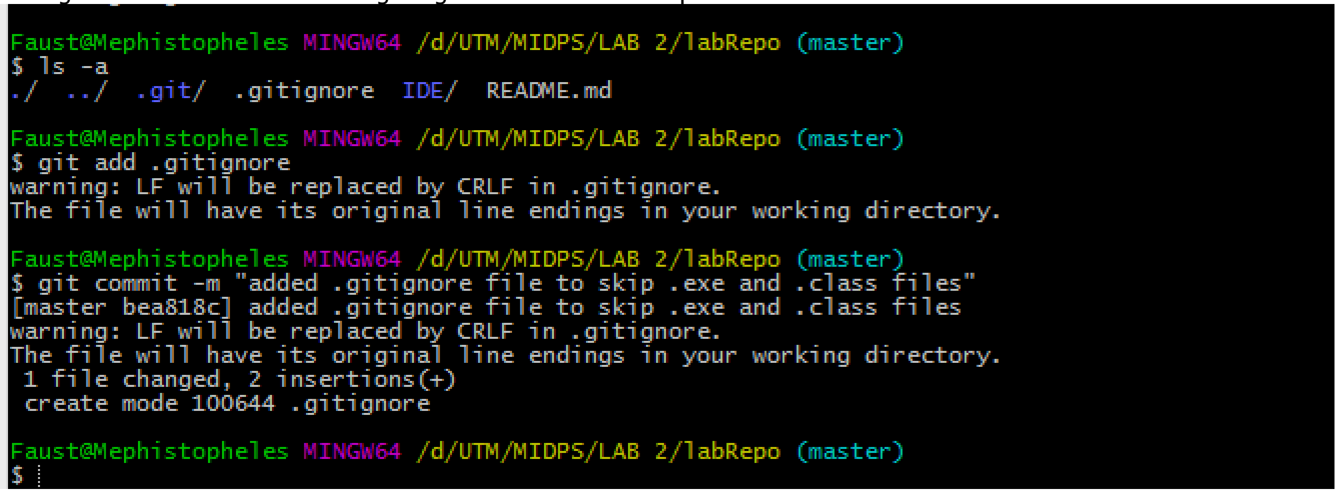
```
javac hello.java
gcc -o hello.exe hello.c
```



```
 [new branch]      master -> master

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ cd ./IDE/HelloWorldPrograms/java/

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo/IDE/HelloWorldPrograms/java (master)
$ javac hello.java

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo/IDE/HelloWorldPrograms/java (master)
$ java HelloWorld
Hello World

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo/IDE/HelloWorldPrograms/java (master)
$ cd ../c

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo/IDE/HelloWorldPrograms/c (master)
$  gcc -o hello.exe hello.c

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo/IDE/HelloWorldPrograms/c (master)
$ ls
hello.c  hello.exe*

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo/IDE/HelloWorldPrograms/c (master)
$ ./hello.exe
Hello world

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo/IDE/HelloWorldPrograms/c (master)
$  cd ../cpp

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo/IDE/HelloWorldPrograms/cpp (master)
$ g++ -o hello.exe hello.cpp

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo/IDE/HelloWorldPrograms/cpp (master)
$ ./hello.exe
Hello World!

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo/IDE/HelloWorldPrograms/cpp (master)
$ cd ../../../

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$
```

- executa primul commit folosind VCS

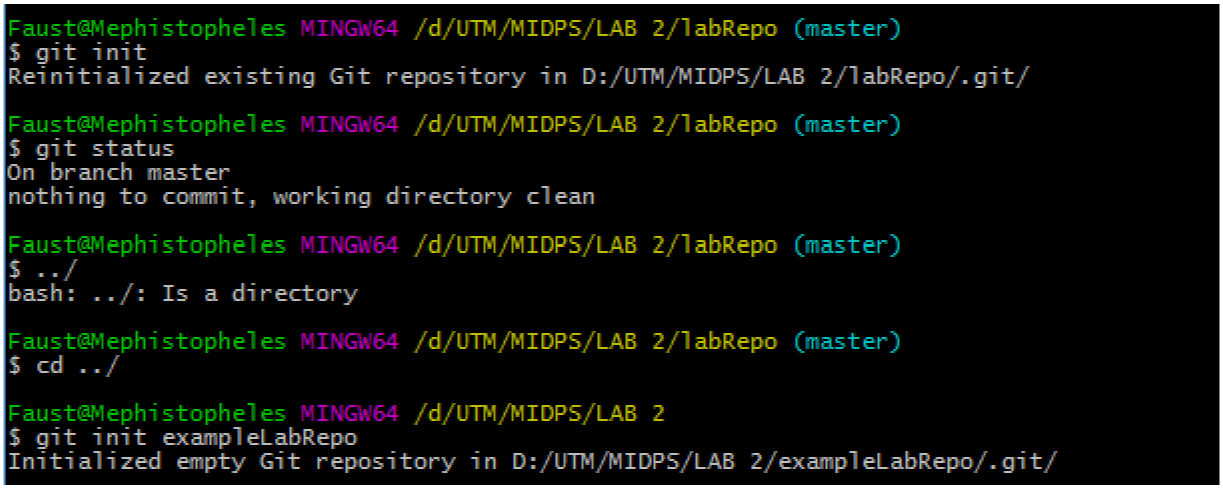
```
vim .gitignore
ls -a
git add .gitignore
git commit -m "added .gitignore file to skip .exe and .class files"
```



```
Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ ls -a
./  ../  .git/  .gitignore  IDE/  README.md

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ git add .gitignore
warning: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory.

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ git commit -m "added .gitignore file to skip .exe and .class files"
[master bea818c] added .gitignore file to skip .exe and .class files
warning: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory.
 1 file changed, 2 insertions(+)
 create mode 100644 .gitignore

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$
```

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

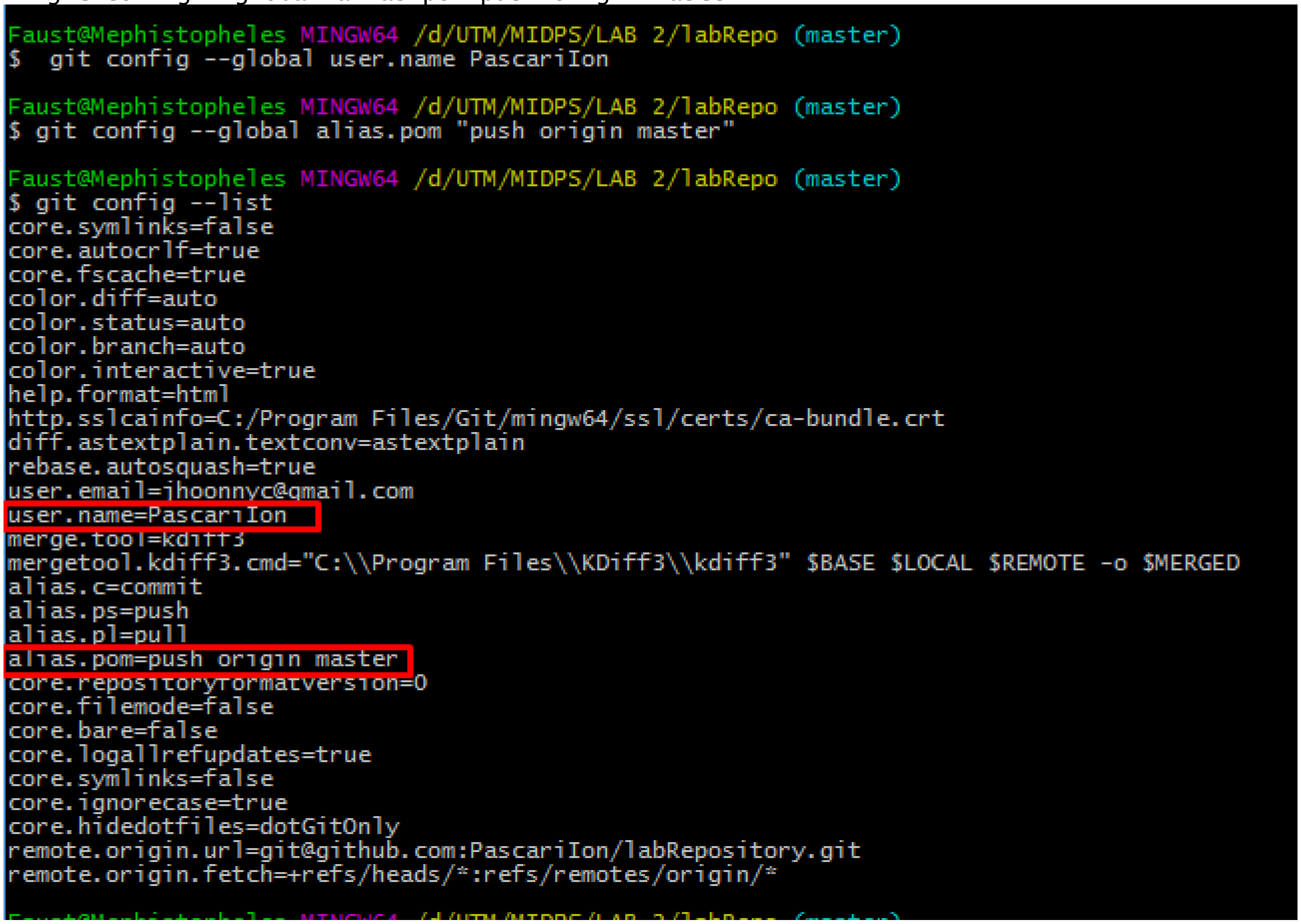
```
git init
git init exampleLabRepo
```



```
Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ git init
Reinitialized existing Git repository in D:/UTM/MIDPS/LAB 2/labRepo/.git/

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ git status
On branch master
nothing to commit, working directory clean

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ ../
bash: ../: Is a directory

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ cd ../

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2
$ git init exampleLabRepo
Initialized empty Git repository in D:/UTM/MIDPS/LAB 2/exampleLabRepo/.git/
```

- configureaza-ti VCS

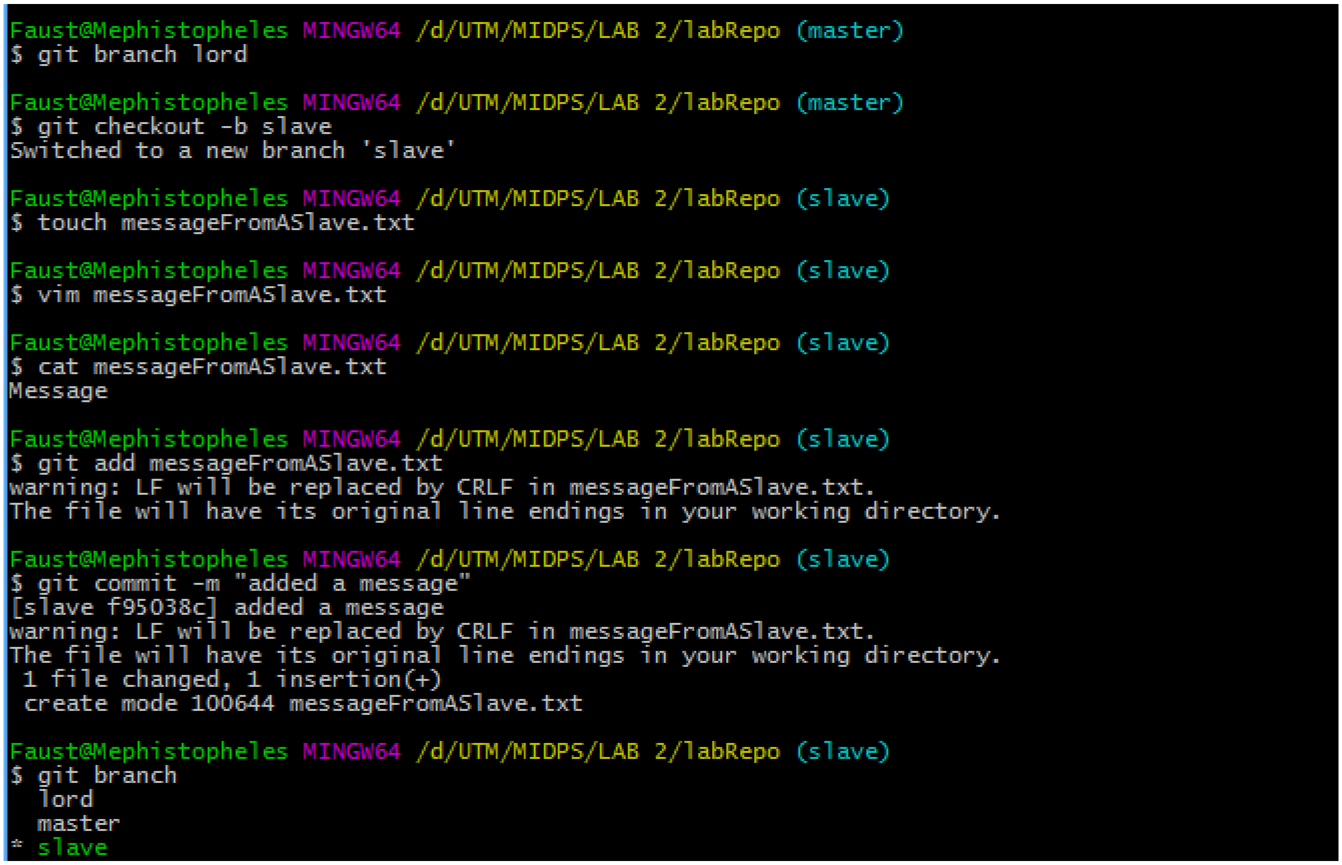
```
git config --global user.name PascariIon
git config --global alias.c commit
git config --global alias.pom push origin master
```



```
Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$  git config --global user.name PascariIon

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ git config --global alias.pom "push origin master"

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ git config --list
core.symlinks=false
core.autocrlf=true
core.fscache=true
color.diff=auto
color.status=auto
color.branch=auto
color.interactive=true
help.format=html
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
diff.astextplain.textconv=astextplain
rebase.autosquash=true
user.email=jhoonnyc@gmail.com
user.name=PascariIon
merge.tool=kdiff3
mergetool.kdiff3.cmd="C:\\Program Files\\KDiff3\\kdiff3" $BASE $LOCAL $REMOTE -o $MERGED
alias.c=commit
alias.ps=push
alias.pl=pull
alias.pom=push origin master
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true
core.hidedotfiles=dotGitOnly
remote.origin.url=git@github.com:PascariIon/labRepository.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
```

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

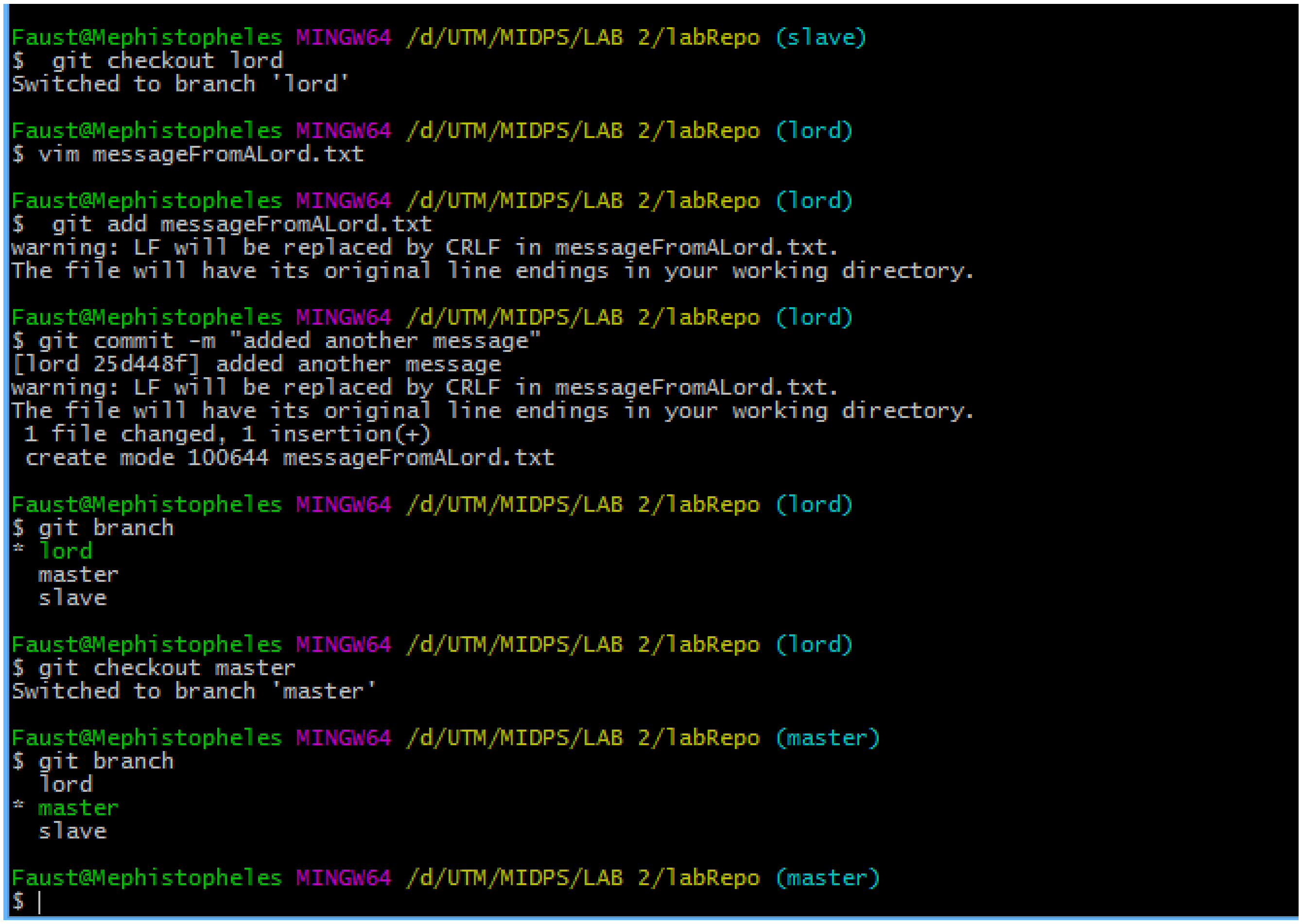
```
git branch lord
git checkout -b slave
```



```
Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ git branch lord

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ git checkout -b slave
Switched to a new branch 'slave'

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (slave)
$ touch messageFromASlave.txt

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (slave)
$ vim messageFromASlave.txt

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (slave)
$ cat messageFromASlave.txt
Message

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (slave)
$ git add messageFromASlave.txt
warning: LF will be replaced by CRLF in messageFromASlave.txt.
The file will have its original line endings in your working directory.

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (slave)
$ git commit -m "added a message"
[slave f95038c] added a message
warning: LF will be replaced by CRLF in messageFromASlave.txt.
The file will have its original line endings in your working directory.
 1 file changed, 1 insertion(+)
 create mode 100644 messageFromASlave.txt

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (slave)
$ git branch
  lord
  master
* slave
```

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

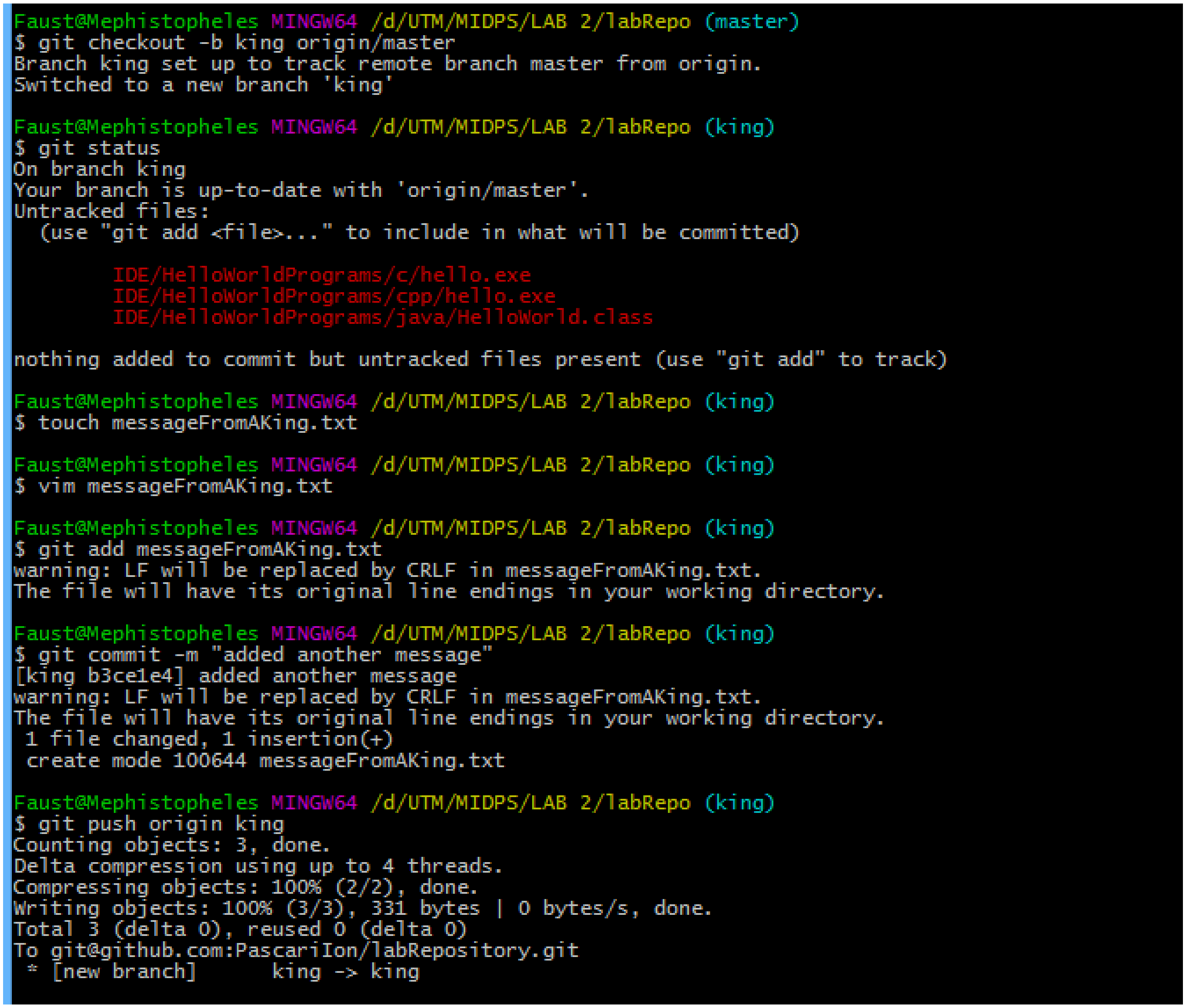
```
git checkout lord
vim messageFromALord.txt
git add messageFromALord.txt
git commit -m "added another message"
```



```
Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (slave)
$  git checkout lord
Switched to branch 'lord'

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (lord)
$ vim messageFromALord.txt

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (lord)
$  git add messageFromALord.txt
warning: LF will be replaced by CRLF in messageFromALord.txt.
The file will have its original line endings in your working directory.

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (lord)
$ git commit -m "added another message"
[lord 25d448f] added another message
warning: LF will be replaced by CRLF in messageFromALord.txt.
The file will have its original line endings in your working directory.
 1 file changed, 1 insertion(+)
 create mode 100644 messageFromALord.txt

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (lord)
$ git branch
* lord
  master
  slave

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (lord)
$ git checkout master
Switched to branch 'master'

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ git branch
  lord
* master
  slave

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ |
```

- *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)

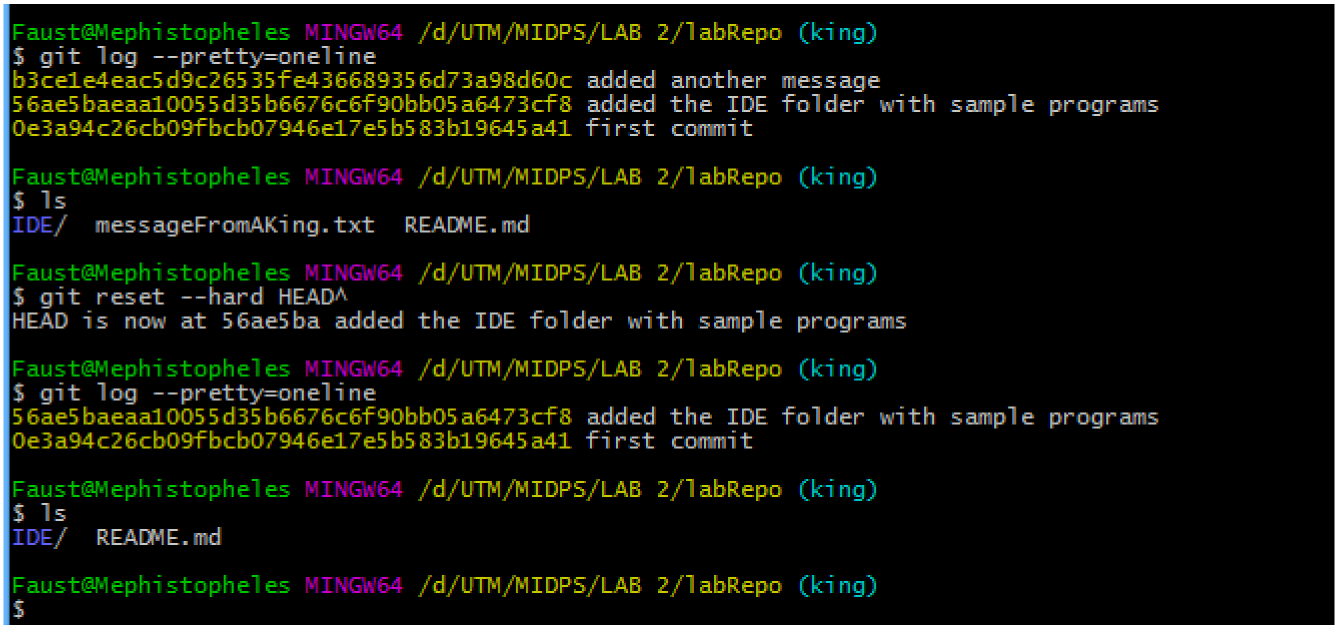
```
git checkout -b king origin/master
git status
touch messageFromAKing.txt
vim messageFromAKing.txt
git add messageFromAKing.txt
git commit -m "added another message"
git push origin king
```



```
Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ git checkout -b king origin/master
Branch king set up to track remote branch master from origin.
Switched to a new branch 'king'

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (king)
$ git status
On branch king
Your branch is up-to-date with 'origin/master'.
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        IDE/HelloWorldPrograms/c/hello.exe
        IDE/HelloWorldPrograms/cpp/hello.exe
        IDE/HelloWorldPrograms/java/HelloWorld.class

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

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (king)
$ touch messageFromAKing.txt

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (king)
$ vim messageFromAKing.txt

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (king)
$ git add messageFromAKing.txt
warning: LF will be replaced by CRLF in messageFromAKing.txt.
The file will have its original line endings in your working directory.

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (king)
$ git commit -m "added another message"
[king b3ce1e4] added another message
warning: LF will be replaced by CRLF in messageFromAKing.txt.
The file will have its original line endings in your working directory.
 1 file changed, 1 insertion(+)
 create mode 100644 messageFromAKing.txt

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (king)
$ git push origin king
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 331 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To git@github.com:PascariIon/labRepository.git
 * [new branch]      king -> king
```

- reseteaza un branch la commit-ul anterior

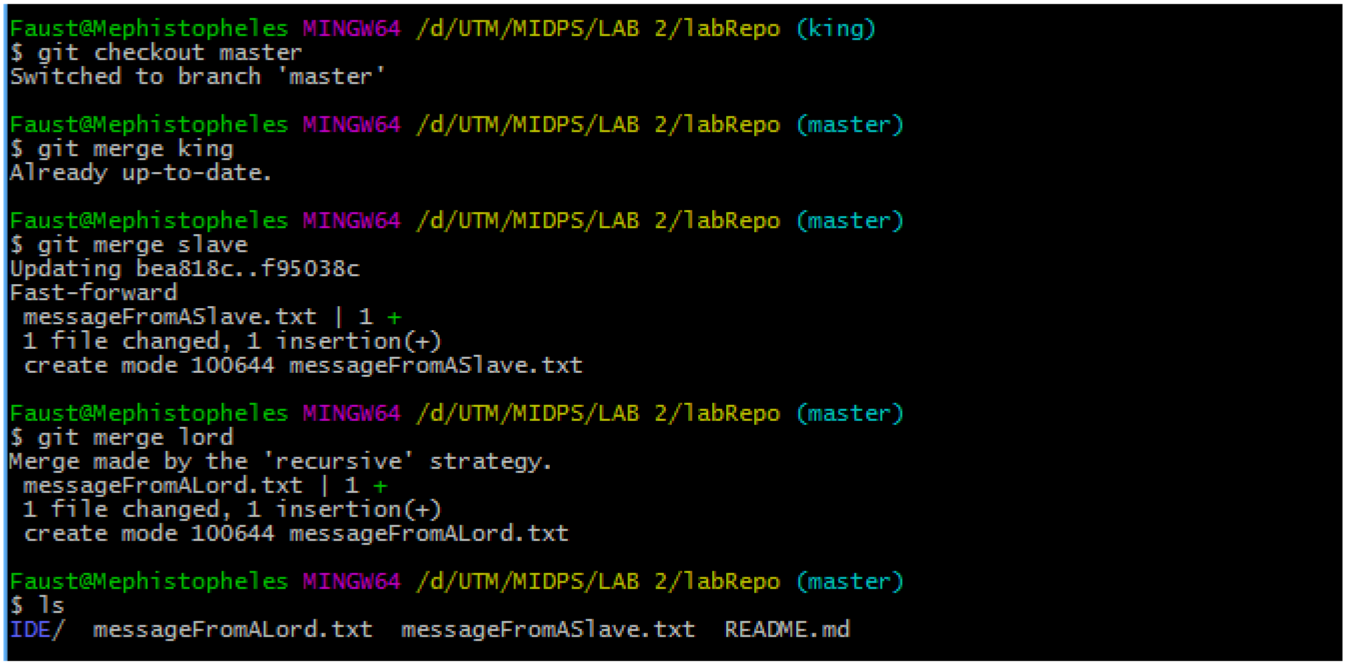
```
git log --pretty=oneline
ls
git reset --hard HEAD^
git log --pretty=oneline
ls
```



```
Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (king)
$ git log --pretty=oneline
b3ce1e4eac5d9c26535fe436689356d73a98d60c added another message
56ae5baeaa10055d35b6676c6f90bb05a6473cf8 added the IDE folder with sample programs
0e3a94c26cb09fbcb07946e17e5b583b19645a41 first commit

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (king)
$ ls
IDE/  messageFromAKing.txt  README.md

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (king)
$ git reset --hard HEAD^
HEAD is now at 56ae5ba added the IDE folder with sample programs

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (king)
$ git log --pretty=oneline
56ae5baeaa10055d35b6676c6f90bb05a6473cf8 added the IDE folder with sample programs
0e3a94c26cb09fbcb07946e17e5b583b19645a41 first commit

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (king)
$ ls
IDE/  README.md

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (king)
$
```

- merge 2 branches

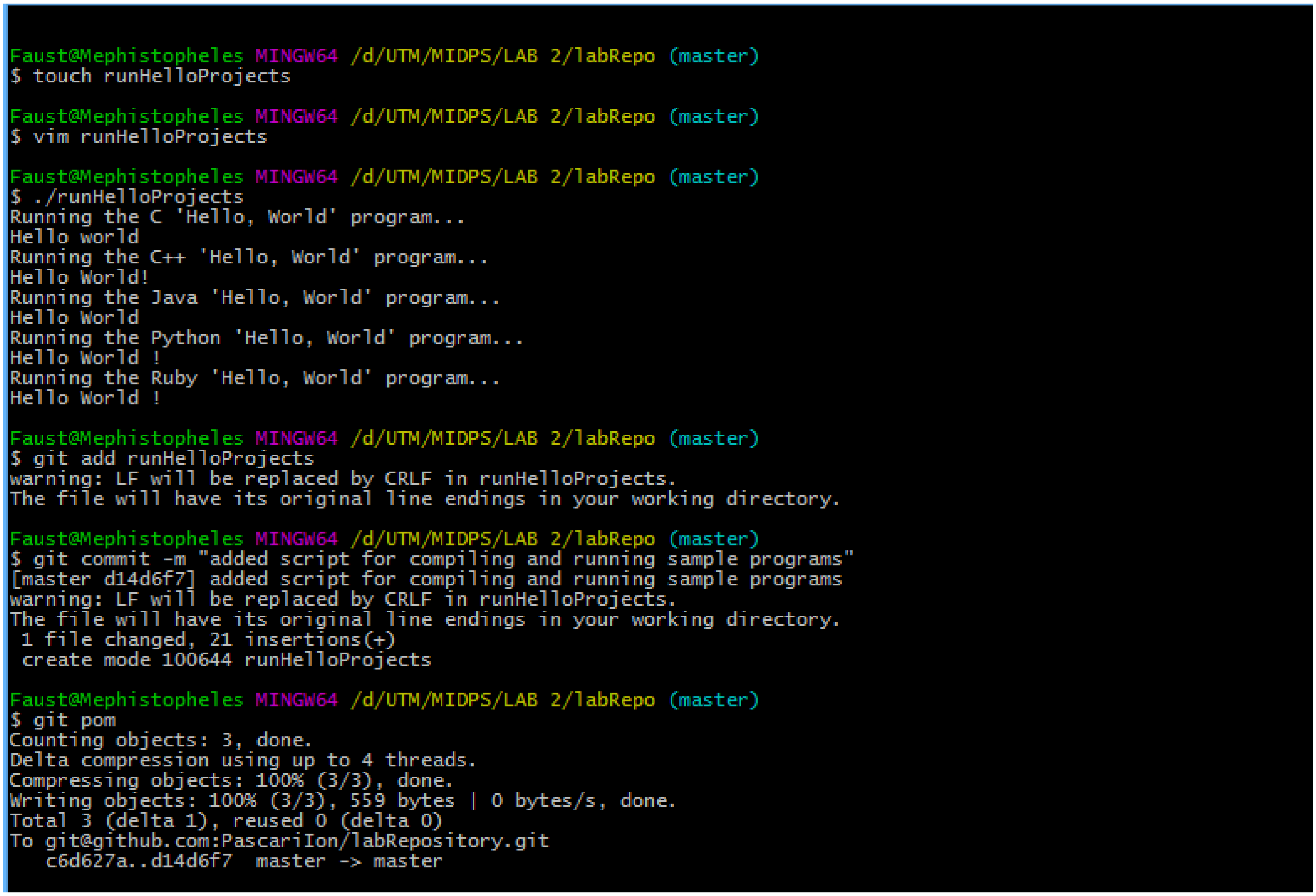
```
git checkout master
git merge king
git merge slave
git merge lord
```



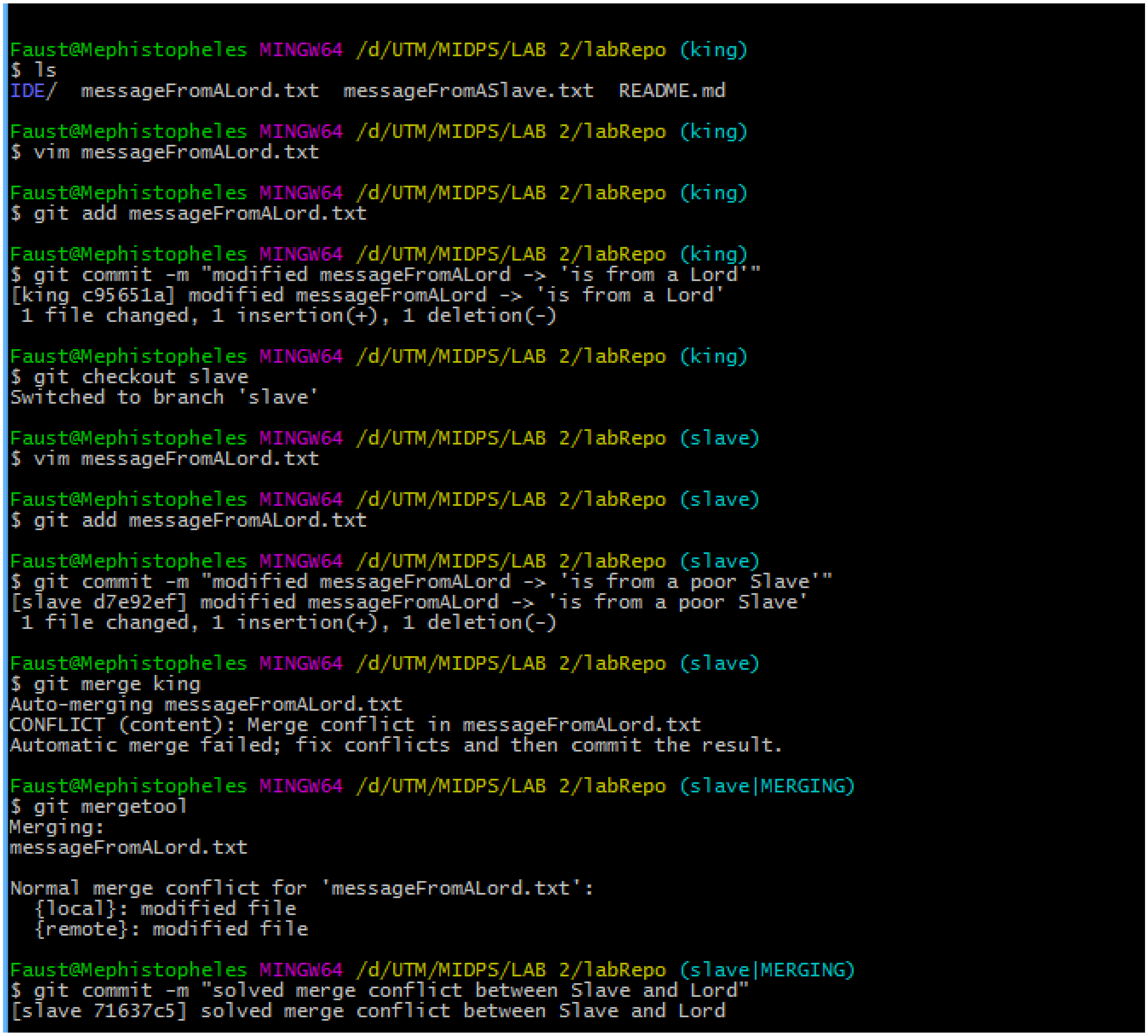
```
Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (king)
$ git checkout master
Switched to branch 'master'

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ git merge king
Already up-to-date.

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ git merge slave
Updating bea818c..f95038c
Fast-forward
 messageFromASlave.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 messageFromASlave.txt

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ git merge lord
Merge made by the 'recursive' strategy.
 messageFromALord.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 messageFromALord.txt

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ ls
IDE/  messageFromALord.txt  messageFromASlave.txt  README.md
```

- conflict solving between 2 branches

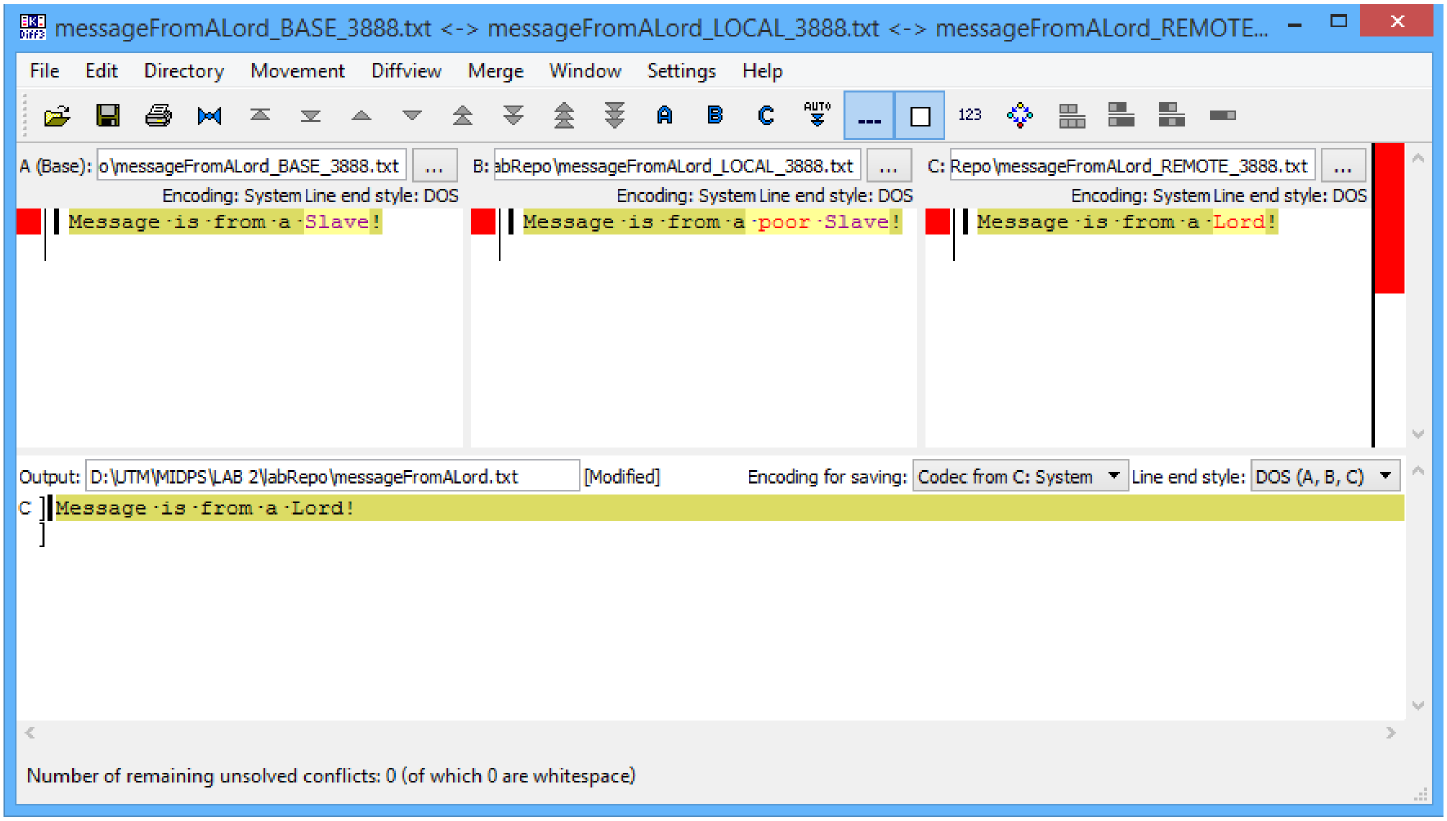
```
vim messageFromALord.txt
git add messageFromALord.txt
git commit -m "modified messageFromALord -> 'is from a Lord'"
git checkout slave
vim messageFromALord.txt
git add messageFromALord.txt
git commit -m "modified messageFromALord -> 'is from a poor Slave'"
git merge king
git mergetool
git commit -m "solved merge conflict between Slave and Lord"
git checkout master
```



```
Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ touch runHelloProjects

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ vim runHelloProjects

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ ./runHelloProjects
Running the C 'Hello, World' program...
Hello world
Running the C++ 'Hello, World' program...
Hello World!
Running the Java 'Hello, World' program...
Hello World
Running the Python 'Hello, World' program...
Hello World !
Running the Ruby 'Hello, World' program...
Hello World !

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ git add runHelloProjects
warning: LF will be replaced by CRLF in runHelloProjects.
The file will have its original line endings in your working directory.

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ git commit -m "added script for compiling and running sample programs"
[master d14d6f7] added script for compiling and running sample programs
warning: LF will be replaced by CRLF in runHelloProjects.
The file will have its original line endings in your working directory.
 1 file changed, 21 insertions(+)
 create mode 100644 runHelloProjects

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ git pom
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 559 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To git@github.com:PascariIon/labRepository.git
   c6d627a..d14d6f7  master -> master
```



```
Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (king)
$ ls
IDE/  messageFromALord.txt  messageFromASlave.txt  README.md

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (king)
$ vim messageFromALord.txt

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (king)
$ git add messageFromALord.txt

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (king)
$ git commit -m "modified messageFromALord -> 'is from a Lord'"
[king c95651a] modified messageFromALord -> 'is from a Lord'
 1 file changed, 1 insertion(+), 1 deletion(-)

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (king)
$ git checkout slave
Switched to branch 'slave'

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (slave)
$ vim messageFromALord.txt

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (slave)
$ git add messageFromALord.txt

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (slave)
$ git commit -m "modified messageFromALord -> 'is from a poor Slave'"
[slave d7e92ef] modified messageFromALord -> 'is from a poor Slave'
 1 file changed, 1 insertion(+), 1 deletion(-)

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (slave)
$ git merge king
Auto-merging messageFromALord.txt
CONFLICT (content): Merge conflict in messageFromALord.txt
Automatic merge failed; fix conflicts and then commit the result.

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (slave|MERGING)
$ git mergetool
Merging:
messageFromALord.txt

Normal merge conflict for 'messageFromALord.txt':
  {local}: modified file
  {remote}: modified file

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (slave|MERGING)
$ git commit -m "solved merge conflict between Slave and Lord"
[slave 71637c5] solved merge conflict between Slave and Lord
```



messageFromALord_BASE_3888.txt <-> messageFromALord_LOCAL_3888.txt <-> messageFromALord_REMOTE...

A (Base): o\messageFromALord_BASE_3888.txt ...
Encoding: System Line end style: DOS
Message is from a Slave!

B: abRepo\messageFromALord_LOCAL_3888.txt ...
Encoding: System Line end style: DOS
Message is from a poor Slave!

C: Repo\messageFromALord_REMOTE_3888.txt ...
Encoding: System Line end style: DOS
Message is from a Lord!

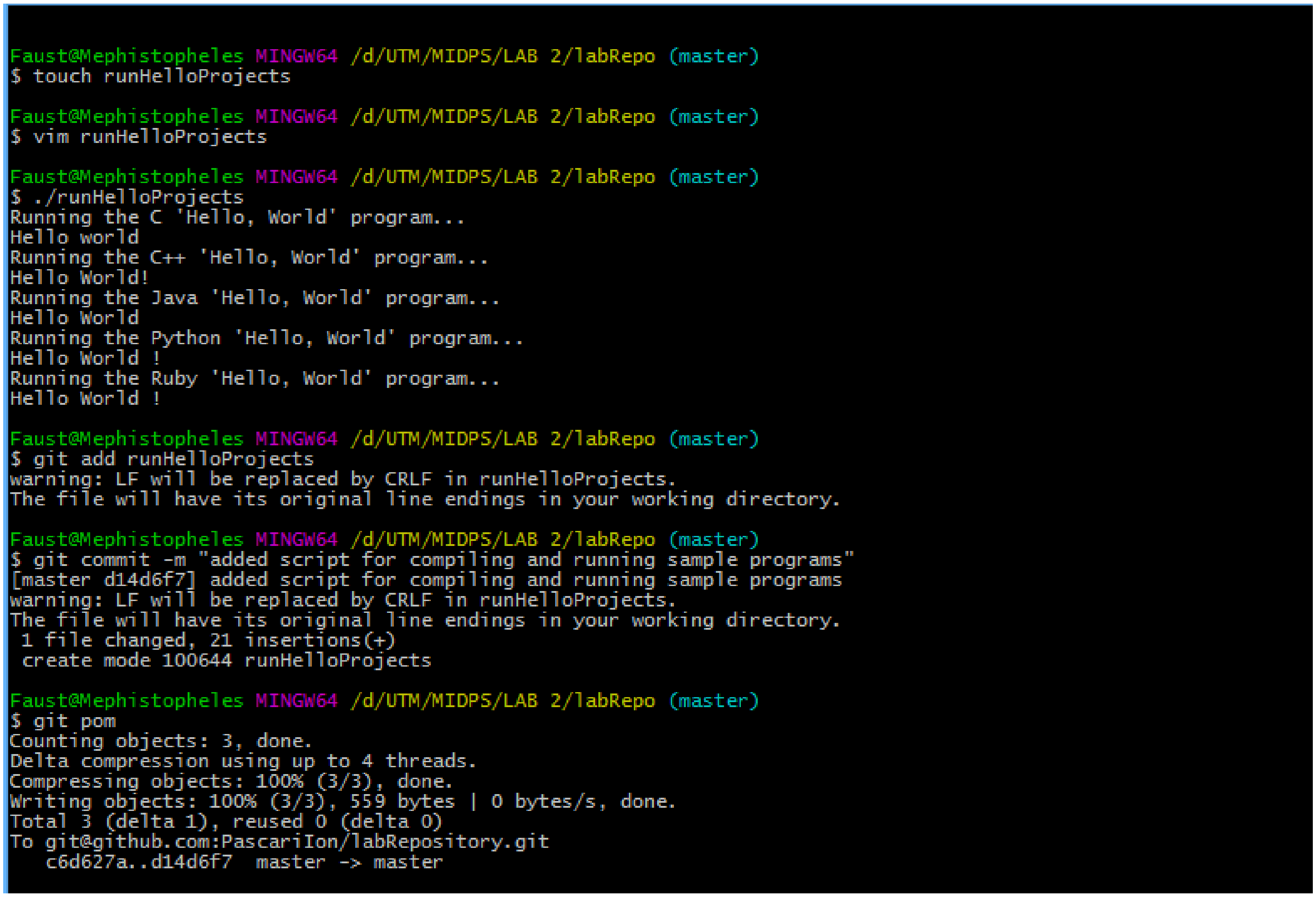
Output: D:\UTM\MIDPS\LAB 2\labRepo\messageFromALord.txt [Modified] Encoding for saving: Codec from C: System Line end style: DOS (A, B, C)

C] Message is from a Lord!

Number of remaining unsolved conflicts: 0 (of which 0 are whitespace)

- *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
```



```
Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ touch runHelloProjects

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ vim runHelloProjects

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ ./runHelloProjects
Running the C 'Hello, World' program...
Hello world
Running the C++ 'Hello, World' program...
Hello World!
Running the Java 'Hello, World' program...
Hello World
Running the Python 'Hello, World' program...
Hello World !
Running the Ruby 'Hello, World' program...
Hello World !

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ git add runHelloProjects
warning: LF will be replaced by CRLF in runHelloProjects.
The file will have its original line endings in your working directory.

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ git commit -m "added script for compiling and running sample programs"
[master d14d6f7] added script for compiling and running sample programs
warning: LF will be replaced by CRLF in runHelloProjects.
The file will have its original line endings in your working directory.
 1 file changed, 21 insertions(+)
 create mode 100644 runHelloProjects

Faust@Mephistopheles MINGW64 /d/UTM/MIDPS/LAB 2/labRepo (master)
$ git pom
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 559 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To git@github.com:PascariIon/labRepository.git
   c6d627a..d14d6f7  master -> master
```

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 , https://guides.github.com/activities/hello-world/
- Îndrumar metodic pentru lucrările de laborator la MIDPS