Facultatea Calculatoare, Informatica si Microelectronica

Universitatea Tehnica a Moldovei

Medii Interactive de Dezvoltare a Produselor Soft

Lucrarea de laborator#2

Version Control Systems si modul de setare a unui server

Autor:
Croitoru Ionel

lector asistent:
Irina Cojanu
lector superior:
Svetlana Cojocaru

Lucrarea de laborator #2

1 Scopul lucrarii de laborator

Version Control Systems si modul de setare a unui server

2 Obiective

– 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

3 Laboratory work implementation

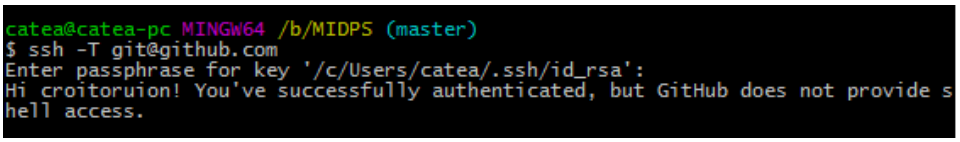
3.1 Tasks and Points

- 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 (nota 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

3.2 Analiza lucrarii de laborator

GitHub este un serviciu de găzduire web pentru proiecte de dezvoltare a software-ului care utilizează sistemul de control al versiunilor Git. GitHub oferă planuri tarifare pentru depozite private, și conturi gratuite pentru proiecte open source. Site-ul a fost lansat în 2008 de către Tom Preston-Werner, Chris Wanstrath, și PJ Hyett.

Pentru elaborarea acestui laborator m-am inregistrat pe github,am creat un repozitoriu si am instalat git bash-ul in calculator . Am reusit sa ma conectez la repozitoriul creat folosind SSH prin git remote add origin email. Am creeat o cheia ce ma autorizeaza automat prin ssh-keygen care creeaza o parola autentificarea de la acest device.



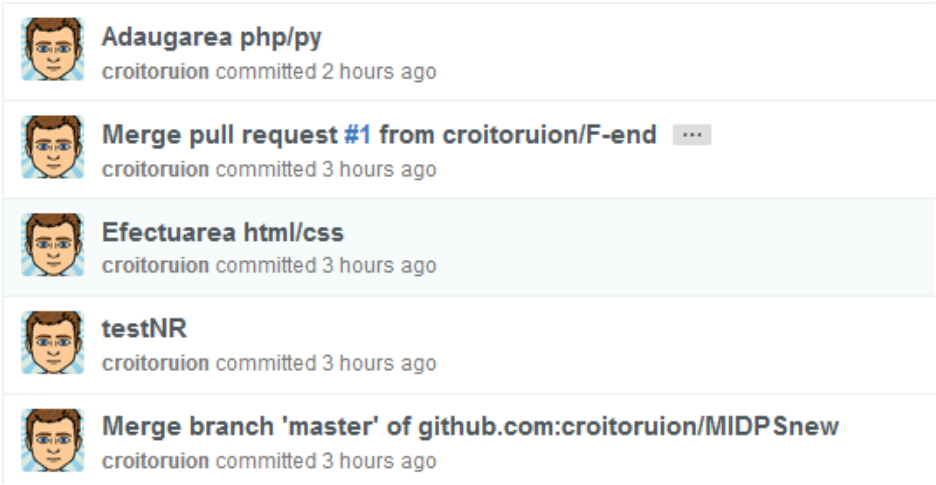
Pentru compilarea si rularea unor programe hello world de exemplu scrise in limbajul java si C++ avem nevoie in primul rand de jdk shi gcc.In setari de la windows introducem calea "path" spre jdk si gcc,pentru compilarea acestor programe in continuare. Am creat un file java si cpp ,am scris acele hello world programe. Dupa ce ne-am autentificat in github cu shh-key putem cu ajutorul comenzii javac sa compilam programul jhello iar apoi cu ajutorul comenzii java sa rulam acest program

```
catea@catea-pc MINGW64 /b/MIDPS/lab#2 (master)
$ javac jhello.java

catea@catea-pc MINGW64 /b/MIDPS/lab#2 (master)
$ java jhello
Hello GITHUB
```

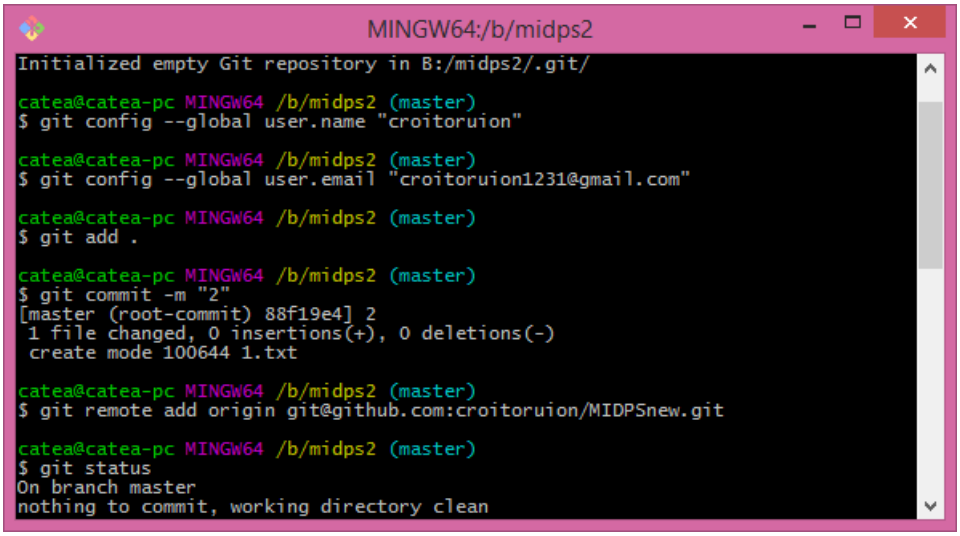
Pentru a compila si rula un program C++ este comanda g++ "den fisierului" -o "denum file-ului exe" iar apoi sal rulam cu ajutorul comenzii ./"denumirea fisierului exe"

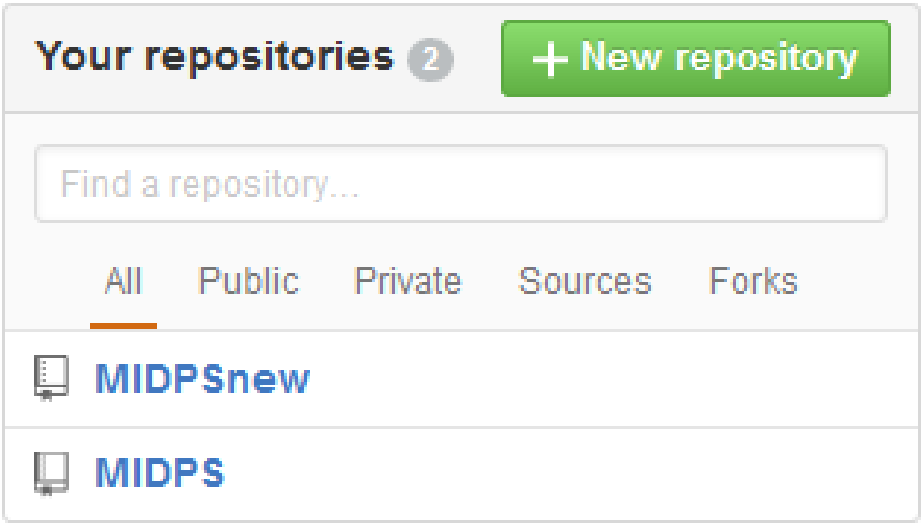
```
catea@catea-pc MINGW64 /b/MIDPS/lab#2 (master)
$ g++ hello.cpp -o C++hello

catea@catea-pc MINGW64 /b/MIDPS/lab#2 (master)
$ ./C++hello
Hello github!

catea@catea-pc MINGW64 /b/MIDPS/lab#2 (master)
$ F
```

La fiecare push se face si un commit care sistematizeaza orice schimbare pentru o usurare. Commiturile se fac prin comanda "git commit -m "COMENTARIU" ,iar apoi se face push pentru intrarea in vigoare a schimbarilor noastre



Am creat un repozitoriu nou MIDPSnew prin git init,apoi am configurat acest repozitoriu (name/email din github)





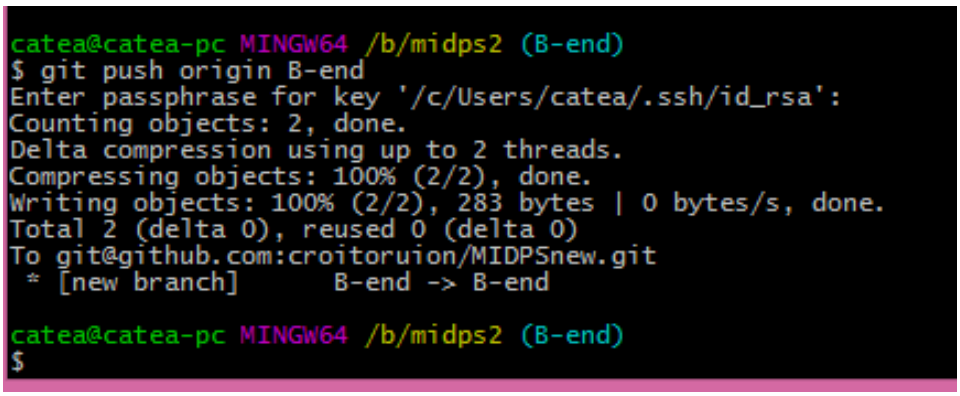
Am creat 2 branch-uri prin ssh cu denumirile F-end si B-end

```
catea@catea-pc MINGW64 /b/midps2 (master)
$ git branch
* master

catea@catea-pc MINGW64 /b/midps2 (master)
$ git branch F-end

catea@catea-pc MINGW64 /b/midps2 (master)
$ git branch B-end

catea@catea-pc MINGW64 /b/midps2 (master)
$ git branch
  B-end
  F-end
* master

catea@catea-pc MINGW64 /b/midps2 (master)
$
```

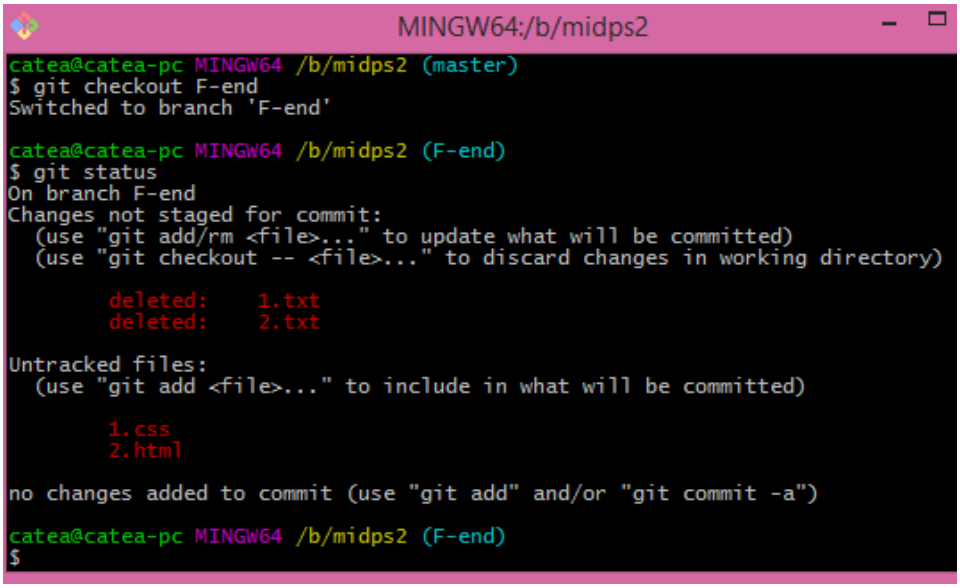
Am creat 2 fisiere din branch-ul B-end si le-am adaugat in repozitoriu

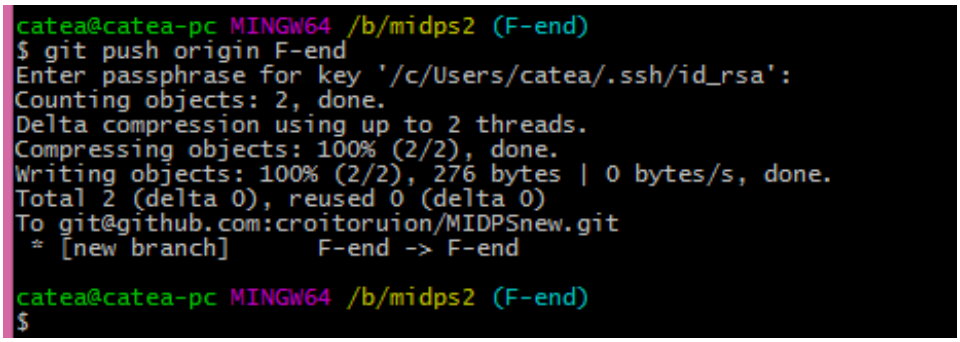
```
catea@catea-pc MINGW64 /b/midps2 (B-end)
$ git status
On branch B-end
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        3.php
        4.py

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

catea@catea-pc MINGW64 /b/midps2 (B-end)
$ git add .

catea@catea-pc MINGW64 /b/midps2 (B-end)
$ git commit -m "Adaugarea php/py"
[B-end 8dcf858] Adaugarea php/py
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 3.php
 create mode 100644 4.py
```

Observam ca push-ul se face cu ajutorul ssh-key

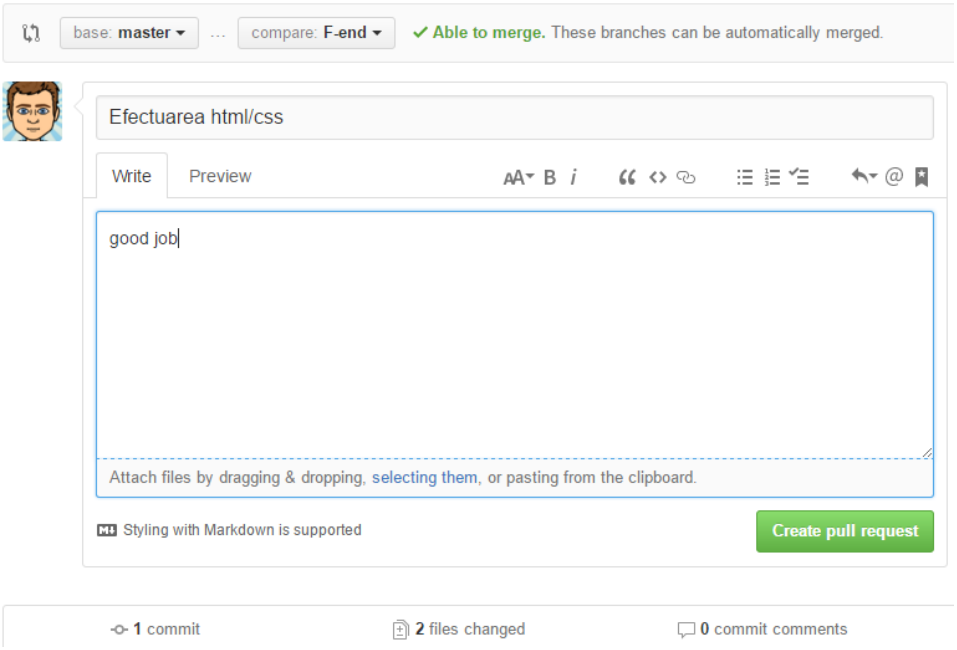


La fel si pentru branch-ul F-end am creat alte 2 fisiere si le-am adaugat in repozitoriu



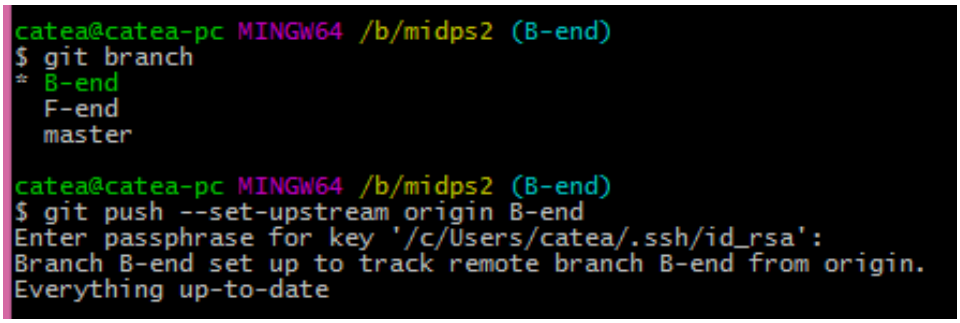


Cand intram pe github in calitate de master observam ca branch-urile noastre creeate au facut push,putem accepta acest push putem sa nu-l acceptam astfel in repozitoriu nu se va schimba nimic.Putem adauga comentarii la acel care a facut acest push/commit



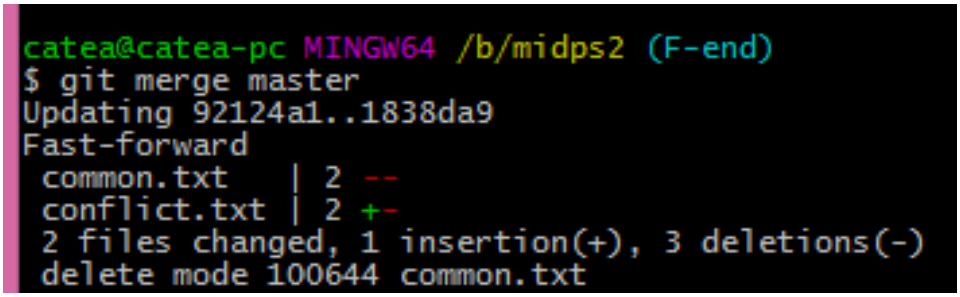
Astfel in urma acceptarii ambelor comituri din partea ambelor branch-uri observam ca in repozitoriu a aparut 4 fisiere cate 2 din partea fiecarui branch

Setam un branch to track remote origin pe care vom putea face push Am selectat branch-ul B-end Am facut o schimbare,astfel am facut un commit



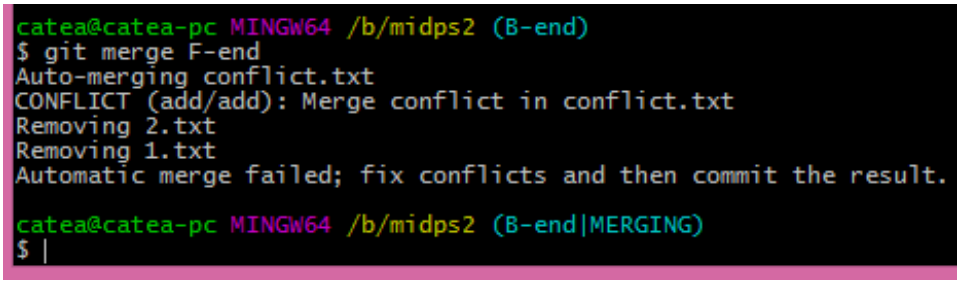
Am intrat in log-urile unde se afla toate commiturile si ne-am reintors la starea anterioara adica la penultimul commit

Merge 2 branches. Pentru a face merge la 2 branch-uri utilizam comanda merge.

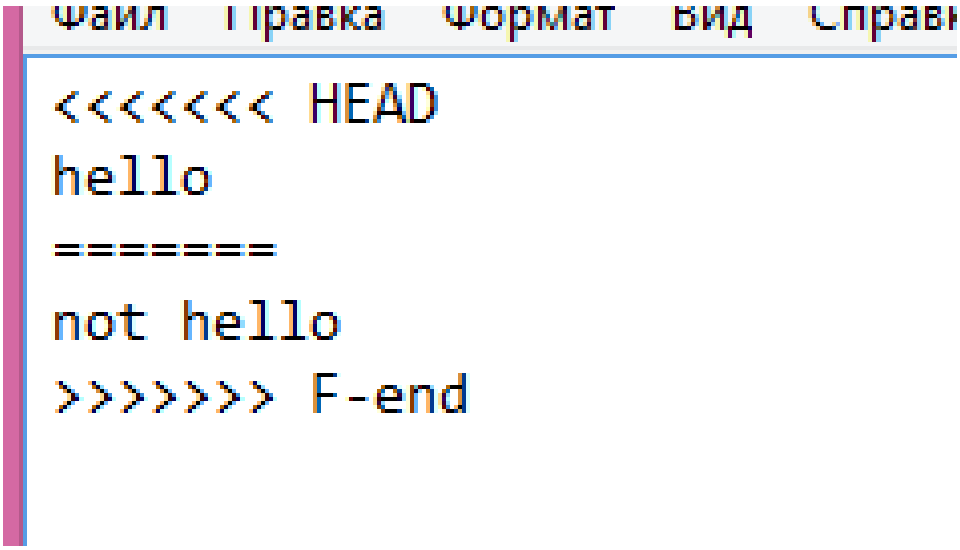


Rezolvarea conflictelor a 2 branch-uri

Am creat 2 fisiere identice dar cu continut diferit in fiecare din branch astfel cand dam merge observam ca apare conflictul



Github-ul ne indica si ne ajuta sa rezolvam acest conflict



Stergem ce nu ne trebuie si lasam ceea de ce avem nevoie apoi ii dam commit

Concluzie

In urma efectuarii acestei lucrari de laborator am studiat serviciile de baza a retelei sociale"IT" numita github.com . Githubul ofera gazduire web pentru proiecte de dezvoltare a software-ului care utilizeaza sistemul de control al Git-ului. Am efectuat pe rand toate task-urile care au fost propuse in conditiile laboratorului precum initializarea si setarea repozitoriului,autentificarea ssh,compilarea si rularea programelor Java si C++,adaugarea commiturilor ,crearea a mai multor branch-uri ,lucrul cu aceste branch-uri precum ,commiturile de pe branch si acceptarea lor de master ,efectuarea a merge la branch-uri ,diminuarea conflictelor de fisiere. Totul se efectueaza prin git Bash care este un terminal destul de comod cu comenzi destul de usoare si eficiente Githubul reprezinta un instrument ce vine in ajutorul echipelor/firmelor/companiilor de dezvoltare a tuturor serviciilor it.

References

1 stackoverflow.com/repository init

2 google.com/git set previous commit

3 google.com/ls equivalent cmd

4 https://www.youtube.com/watch?v=ZFYhW3kBjnE

5 helpgithub.com