Facultatea Calculatoare, Informatica si Microelectronica

Universitatea Tehnică a Moldovei

Catedra Automatica şi tehnologii informaţionale

Medii interactiva de dezvoltare a produselor soft

Lucrare de laborator # 2

Version Control System si modul de setare a unui server

A efectuat: st. gr. TI-141

Teodor Teaca

A verificat: lect. asistent

Irina Cojanu

Chişinău 2016

I. Obiective:

1) Initializeaza uni nou repositoriu
2) Configurarea VCS
3) Crearea branch-urilor
3) Commit pe ambele branch-uri (cel putin 1 commit per branch)

II. Realizare:

- *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
    - rezolvarea conflictelor a 2 branches
- *Bonus Point*:
    - Scrie un script care va compila HelloWolrdPrograms projects.

III. Analiza lucrarii de laborator

Link catre repozitoriu: https://github.com/teacateodor/midps

1) Cream un repozitoriu pe github.com
2) Copiem codul de pe site in git bash pentru a initializa repozitoriul local.

Deci si revenim acum la sarcina de baza , si anume crearea branch.

IV. Branches:

1) In primul rind verificam daca sunt prezente branch-urile la noi in repozitoriu.
   Putem vedea asta prin comanda : **git branch**

```
THEO@DESKTOP-UAOT3VK MINGW64 /d/Programe/Doc/Univer/Sim VI/MIDPS/midps (master)
$ git branch
* master

THEO@DESKTOP-UAOT3VK MINGW64 /d/Programe/Doc/Univer/Sim VI/MIDPS/midps (master)
$
```

2) Cream un branch prin comanda: **git branch Nume**
   Si adaug un fisier **doc**

3) Pentru a schimba branchul introducem
   Git checkout Nume.

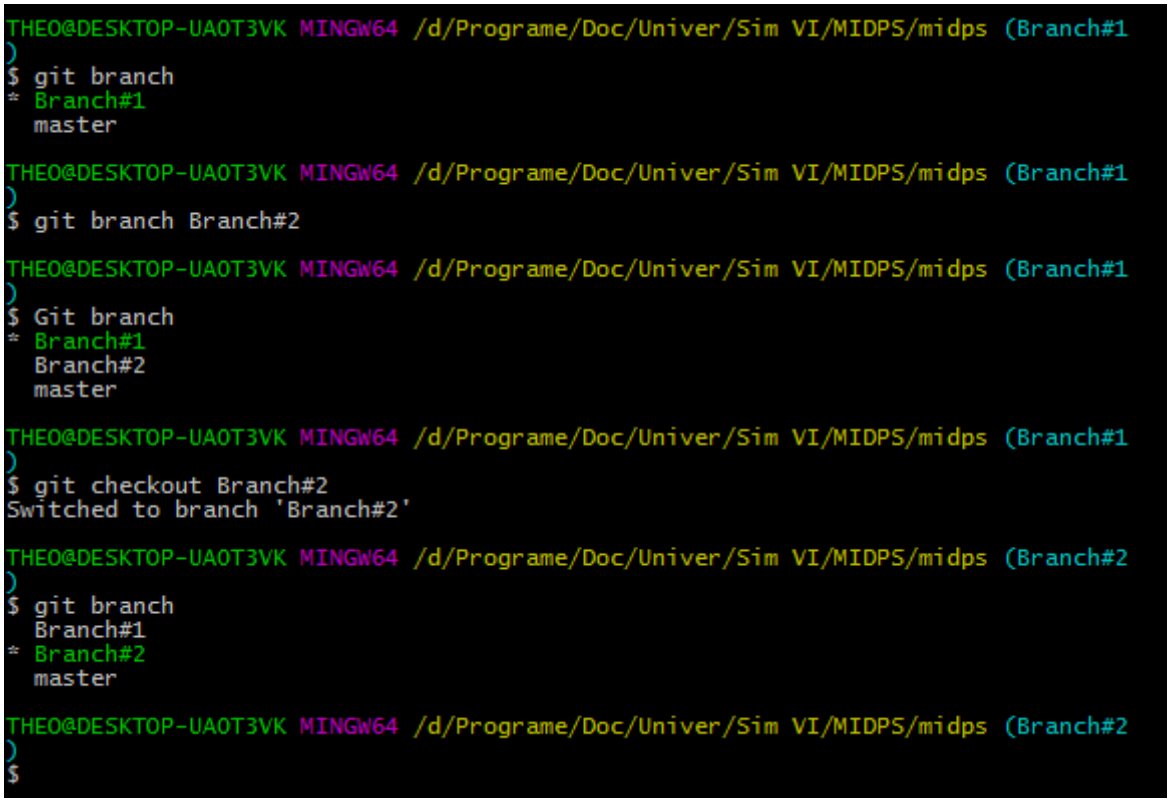
```
$ git branch Branch#1

THEO@DESKTOP-UAOT3VK MINGW64 /d/Programe/Doc/Univer/Sim VI/MIDPS/midpsv3 (master
)
$ git checkout Branch#1
Switched to branch 'Branch#1'

THEO@DESKTOP-UAOT3VK MINGW64 /d/Programe/Doc/Univer/Sim VI/MIDPS/midpsv3 (Branch
#1)
$ git add .

THEO@DESKTOP-UAOT3VK MINGW64 /d/Programe/Doc/Univer/Sim VI/MIDPS/midpsv3 (Branch#1)
$ git status
On branch Branch#1
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   testbtanch1.docx
```

4) Cu ajutorul comenzii
git log
putem afla orice
commit
facut pe acel branch

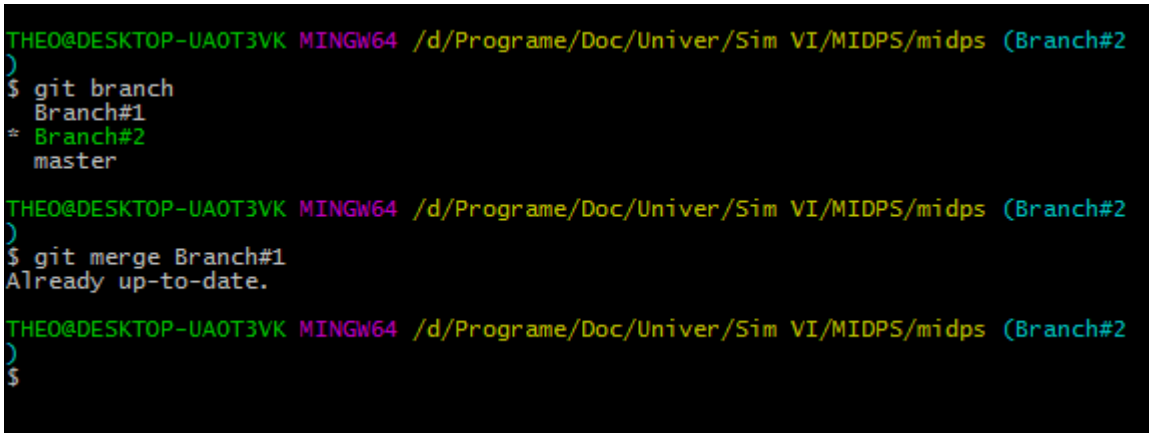
```
THEO@DESKTOP-UAOT3VK MINGW64 /d/Programe/Doc/Univer/Sim VI/MIDPS/midps (Branch#1
)
$ git log
commit bb8d811ed31a6a097368947d08cac335918fed57
Author: teacateodor <teacateodor@mail.ru>
Date:   Wed Apr 6 14:57:54 2016 +0300

    branchtest

commit d5d3c14d053b82fa8e8b3e94c5f95a9ec20cf060
Author: teacateodor <teacateodor@mail.ru>
Date:   Sun Apr 3 21:39:46 2016 +0300

    lab3

commit e69641f6840a9b3d4cc00c0acf7b4070503aa45d
Author: teacateodor <teacateodor@mail.ru>
Date:   Tue Mar 29 00:33:23 2016 +0300

    lab3

commit 42205fe01a12191bf6cf278c788ea224449135d1
Author: teacateodor <teacateodor@mail.ru>
Date:   Sun Mar 27 01:37:52 2016 +0200

    lab1
:...skipping...
commit bb8d811ed31a6a097368947d08cac335918fed57
Author: teacateodor <teacateodor@mail.ru>
Date:   Wed Apr 6 14:57:54 2016 +0300

    branchtest

commit d5d3c14d053b82fa8e8b3e94c5f95a9ec20cf060
Author: teacateodor <teacateodor@mail.ru>
Date:   Sun Apr 3 21:39:46 2016 +0300

    lab3

commit e69641f6840a9b3d4cc00c0acf7b4070503aa45d
Author: teacateodor <teacateodor@mail.ru>
Date:   Tue Mar 29 00:33:23 2016 +0300

    lab3

commit 42205fe01a12191bf6cf278c788ea224449135d1
Author: teacateodor <teacateodor@mail.ru>
Date:   Sun Mar 27 01:37:52 2016 +0200

    lab1

commit 52ad87c51773d8abf35a229a6dc50a1adc8b57d9
Author: teacateodor <teacateodor@mail.ru>
Date:   Sun Mar 27 01:30:32 2016 +0200

    lab1
```

V. Cream al doilea branch

1) Verificam branchurile care sunt existente.
2)Observam ca exista doar master si un branch care lam creat.
3)Putem crea un al branch , prin comanda git branch nume.
4)Dupa ce am creat facem mutare pe el cu ajutorul comenzii git checkout nume , si cream un fisier doc.
5)Il activam cu ajutorul comenzii git push origin Branch#2.



5) **Merge a celor doua branchuri**
   1) Verificam pe ce branch suntem.
   2) Scriem comanda de a face merge a celor doua branchuri.
   3) Ne bucuram de succes ! ;D .



VI. Concluzie

Realizind aceasta lucrare de laborator am facut cunostinta mai buna cu VCS GitHub si modul de setare a unui server.
La un repozitoriu am creat si am setat 2 branchuri , **Branch#1** si **Branch#2** si cite un comnit de fiecare branch apoi am facut merge la acele doua branchuri create de mine.
Sunt sigur ca cunostintele acumulate la VCS GitHub mi vor fi folositoare in viitor.