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:

Ursachi Daniel

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 gc-c/g++/javac/python

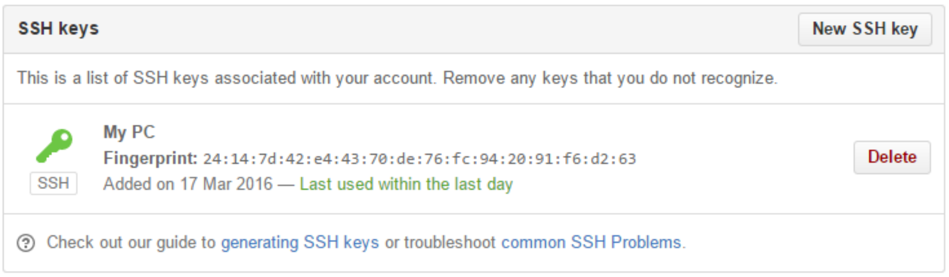
3 Laboratory work implementation

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

Pentru inceput am reusit sa creez un repositoriu pe github , creind un repozitoriu, dupa care am instalat programul git in calculator. Am reusit sa conectez folosind SSH prin git remote add origin email. Am creeat o cheia ce ma autorizeaza automat prin ssh-keygen care creeaza un passphrase for key pentru autentificarea de la acest device.


SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

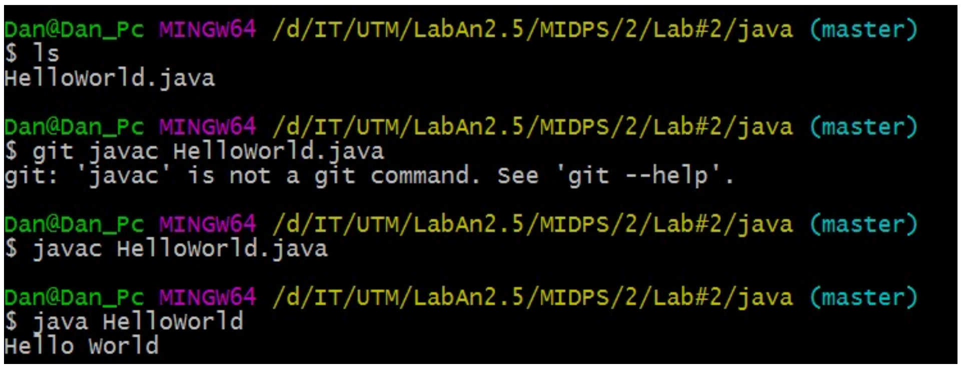
SSH

My PC
Fingerprint: 24:14:7d:42:e4:43:70:de:76:fc:94:20:91:f6:d2:63
Added on 17 Mar 2016 — Last used within the last day

Delete

Check out our guide to generating SSH keys or troubleshoot common SSH Problems.

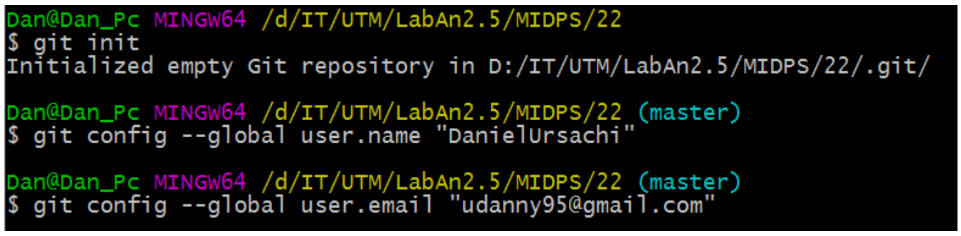
Am creeat un file java si dupa instalara jdk-ului in windowsul local, am reusit sa compilez acest cod prin ssh in github, prin javac - compilarea si accesarea file-ului compilat (Avem necesitatea stricta de a instala virtual masina in windows-ul local pentru posibilitatea rularii programelor, pe motiv ca se acceseaza acest jdk din calculatorul meu).


```
Dan@Dan_Pc MINGW64 /d/IT/UTM/LabAn2.5/MIDPS/2/Lab#2/java (master)
$ ls
HelloWorld.java

Dan@Dan_Pc MINGW64 /d/IT/UTM/LabAn2.5/MIDPS/2/Lab#2/java (master)
$ git javac HelloWorld.java
git: 'javac' is not a git command. See 'git --help'.

Dan@Dan_Pc MINGW64 /d/IT/UTM/LabAn2.5/MIDPS/2/Lab#2/java (master)
$ javac HelloWorld.java

Dan@Dan_Pc MINGW64 /d/IT/UTM/LabAn2.5/MIDPS/2/Lab#2/java (master)
$ java HelloWorld
Hello World
```

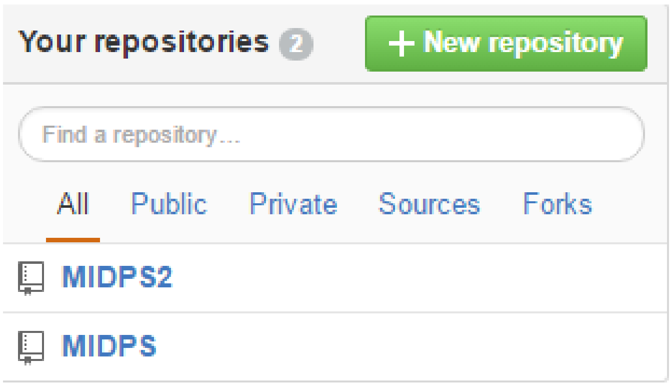
Am reusti sa comentez fiecare schimbare in file-urile locale prin comenzile add dupa care comentem prin commit -m ”comentariu” si git push, care introduce schimbarile pe git. Afisarea comentariilor:

Lab#2/java .java 9 hours ago

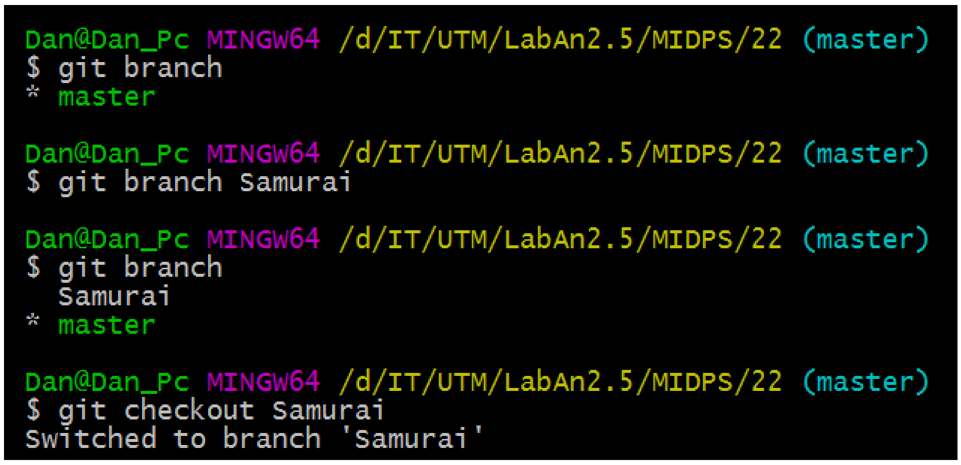
Initializarea unui nou repozitoriu prin comanda init prin ssh si am configurat setarile lui


```
Dan@Dan_Pc MINGW64 /d/IT/UTM/LabAn2.5/MIDPS/22
$ git init
Initialized empty Git repository in D:/IT/UTM/LabAn2.5/MIDPS/22/.git/

Dan@Dan_Pc MINGW64 /d/IT/UTM/LabAn2.5/MIDPS/22 (master)
$ git config --global user.name "DanielUrsachi"

Dan@Dan_Pc MINGW64 /d/IT/UTM/LabAn2.5/MIDPS/22 (master)
$ git config --global user.email "udanny95@gmail.com"
```

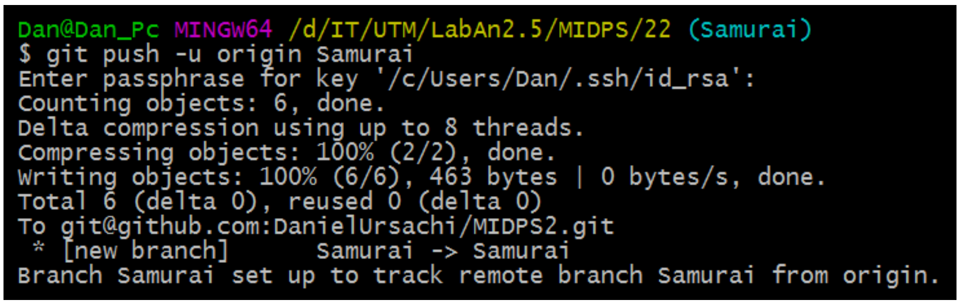
Dupa care in https://github.com/new am introdus acest repozitoriu si l-am adaugat



Am creat un branch nou prin ssh cu denumirea Samurai si l-am ales pentru prelucrarea schimbarilor prin el

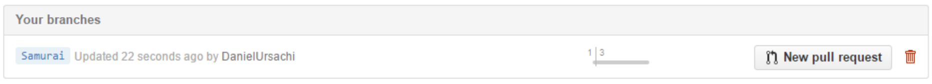

```
Dan@Dan_Pc MINGW64 /d/IT/UTM/LabAn2.5/MIDPS/22 (master)
$ git branch
* master

Dan@Dan_Pc MINGW64 /d/IT/UTM/LabAn2.5/MIDPS/22 (master)
$ git branch Samurai

Dan@Dan_Pc MINGW64 /d/IT/UTM/LabAn2.5/MIDPS/22 (master)
$ git branch
  Samurai
* master

Dan@Dan_Pc MINGW64 /d/IT/UTM/LabAn2.5/MIDPS/22 (master)
$ git checkout Samurai
Switched to branch 'Samurai'
```

Am efectuat efectuat o actiune prin acest branch nou



```
Dan@Dan_Pc MINGW64 /d/IT/UTM/LabAn2.5/MIDPS/22 (Samurai)
$ git push -u origin Samurai
Enter passphrase for key '/c/Users/Dan/.ssh/id_rsa':
Counting objects: 6, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (6/6), 463 bytes | 0 bytes/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To git@github.com:DanielUrsachi/MIDPS2.git
 * [new branch]      Samurai -> Samurai
Branch Samurai set up to track remote branch Samurai from origin.
```

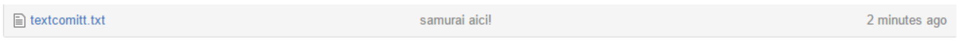
Dupa care am reusit sa fac commit prin branch-ul nou si in calitate de master trebuie sa acceptam sau nu schimbarile



Your branches

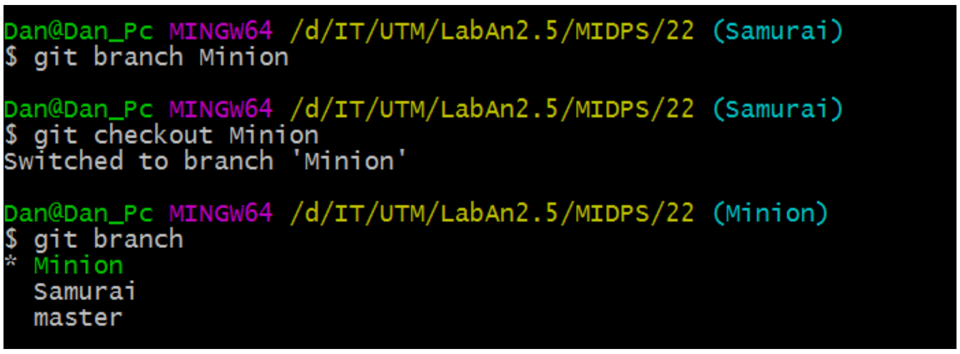
Samurai Updated 22 seconds ago by DanielUrsachi 1 | 3 New pull request

Prin acceptare, obtinem:



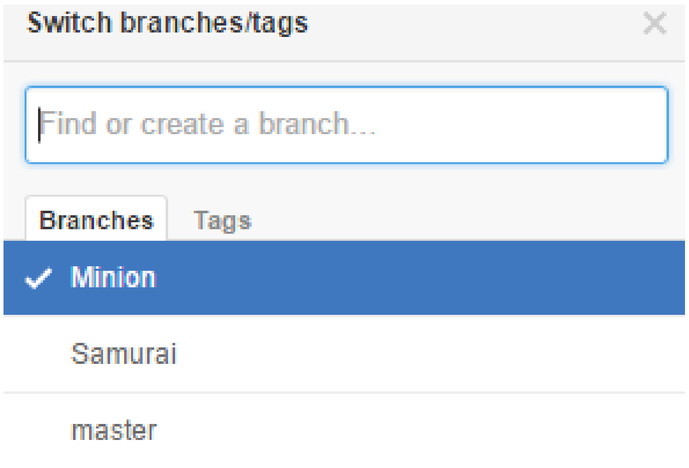
textcomitt.txt samurai aici! 2 minutes ago

Crearea si alegerea unui nou branch

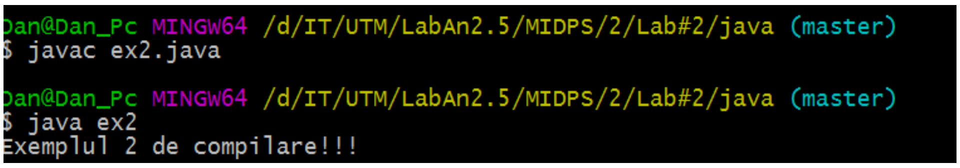


```
Dan@Dan_Pc MINGW64 /d/IT/UTM/LabAn2.5/MIDPS/22 (Samurai)
$ git branch Minion

Dan@Dan_Pc MINGW64 /d/IT/UTM/LabAn2.5/MIDPS/22 (Samurai)
$ git checkout Minion
Switched to branch 'Minion'

Dan@Dan_Pc MINGW64 /d/IT/UTM/LabAn2.5/MIDPS/22 (Minion)
$ git branch
* Minion
  Samurai
  master
```

Obtinem:

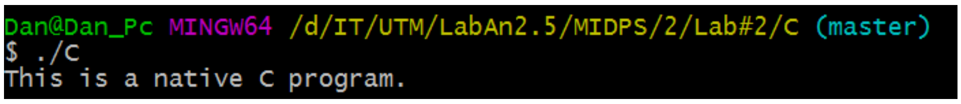


Compilarea file-ului 2



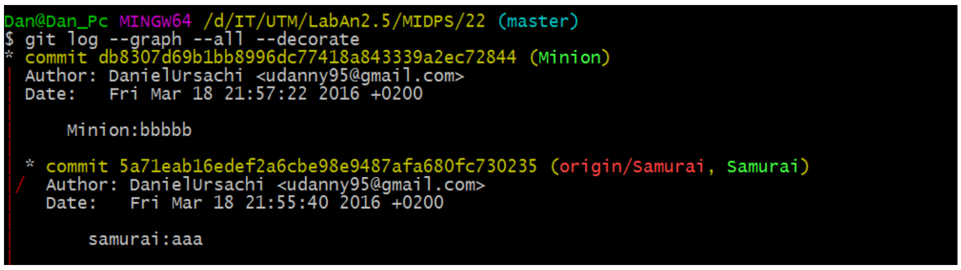
```
Dan@Dan_Pc MINGW64 /d/IT/UTM/LabAn2.5/MIDPS/2/Lab#2/java (master)
$ javac ex2.java

Dan@Dan_Pc MINGW64 /d/IT/UTM/LabAn2.5/MIDPS/2/Lab#2/java (master)
$ java ex2
Exemplul 2 de compilare!!!
```

Rularea programului de C prin metoda similara la Java



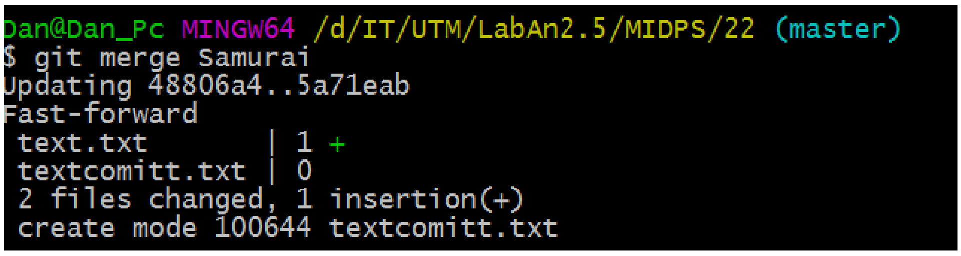
```
Dan@Dan_Pc MINGW64 /d/IT/UTM/LabAn2.5/MIDPS/2/Lab#2/C (master)
$ ./C
This is a native C program.
```

Am creeat un file gol .txt, dupa care prin cele 2 branchuri, prin branch-ul Samurai am scris in file ”Samurai:aaaa” si prin branch-ul Minion am scris in file ”Minion:bbbbb”, dupa care schimbind branch-ul in master, prin apelarea functiei ce ne afiseaza commit-urile, am obtinut CONFLICTUL:



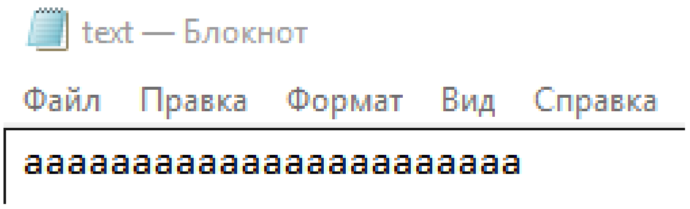
```
Dan@Dan_Pc MINGW64 /d/IT/UTM/LabAn2.5/MIDPS/22 (master)
$ git log --graph --all --decorate
* commit db8307d69b1bb8996dc77418a843339a2ec72844 (Minion)
| Author: DanielUrsachi <udanny95@gmail.com>
| Date:   Fri Mar 18 21:57:22 2016 +0200
|
|     Minion:bbbbb
|
| * commit 5a71eab16edef2a6cbe98e9487afa680fc730235 (origin/Samurai, Samurai)
|/  Author: DanielUrsachi <udanny95@gmail.com>
|   Date:   Fri Mar 18 21:55:40 2016 +0200
|
|       samurai:aaa
|
```

Pentru rezolvarea conflictului am ales schimbarile branch-ului Samurai prin comanda:



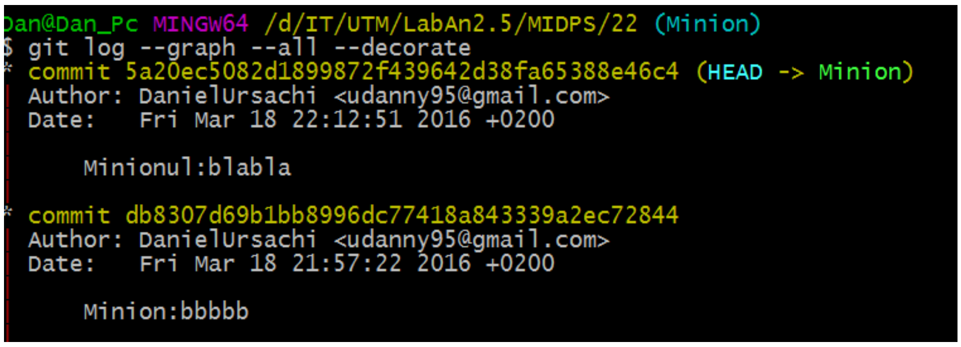
```
Dan@Dan_Pc MINGW64 /d/IT/UTM/LabAn2.5/MIDPS/22 (master)
$ git merge Samurai
Updating 48806a4..5a71eab
Fast-forward
 text.txt        | 1 +
 textcomitt.txt  | 0
 2 files changed, 1 insertion(+)
 create mode 100644 textcomitt.txt
```

Astfel, am obtinut prin branch-ul master continutul file-ului echivalent cu cel al branch-ului Samurai



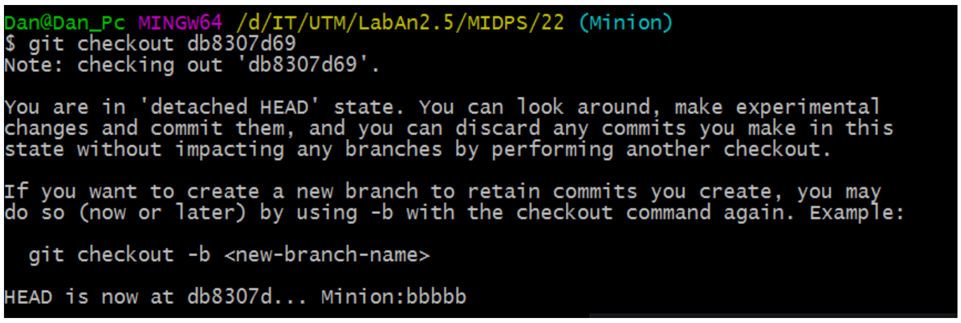
```
aaaaaaaaaaaaaaaaaaaaaaa
```

RESETAREA branch-ului la comitul anterior am efectuat prin schimbarea a 2 schimbari a textului din file din acelasi branch



```
Dan@Dan_Pc MINGW64 /d/IT/UTM/LabAn2.5/MIDPS/22 (Minion)
$ git log --graph --all --decorate
* commit 5a20ec5082d1899872f439642d38fa65388e46c4 (HEAD -> Minion)
| Author: DanielUrsachi <udanny95@gmail.com>
| Date:   Fri Mar 18 22:12:51 2016 +0200
|
|     Minionul:blabla
|
* commit db8307d69b1bb8996dc77418a843339a2ec72844
| Author: DanielUrsachi <udanny95@gmail.com>
| Date:   Fri Mar 18 21:57:22 2016 +0200
|
|     Minion:bbbbb
```

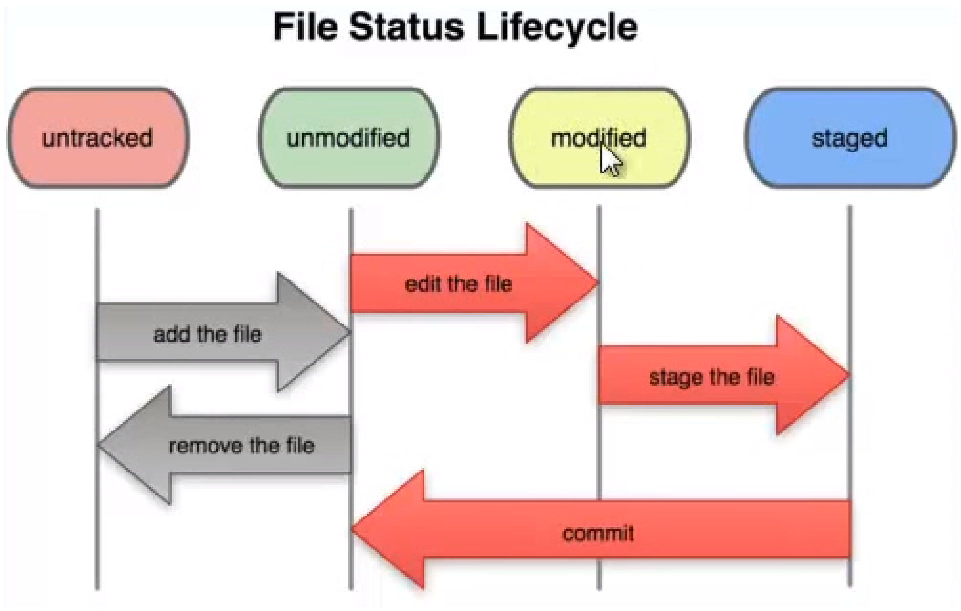
Dupa care, selectam prima parte a adresa codului pentru schimbare si o setam la checkout:



```
Dan@Dan_Pc MINGW64 /d/IT/UTM/LabAn2.5/MIDPS/22 (Minion)
$ git checkout db8307d69
Note: checking out 'db8307d69'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

  git checkout -b <new-branch-name>

HEAD is now at db8307d... Minion:bbbbb
```

Concluzie

In aceasta lucrare de laborator am avut ocazia sa studiez instrumentele de baza pentru lucrul in echipa la un proiect pe platforma github. Am invatat metodele de initializere si setare a acestui program, dupa care am implementat cunostintele din comandLine folosind operatiunile pentru gitBash. Dupa configurare am invatat etapele unui file pentru actualizarea sa in la toate nivelele. Aceasta sitema este foarte clar demonstrata in schema:



Dupa care am studiat metpdele de rulare online folosind resursele calculatorului meu, dar implementind metoda de lucru ca server. Dupa care am invatat metodele de ramificare a branchurilor si gestionarea lor comoda si multi-functionala.

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