Ministerul Educației Culturii și Cercetarii al Republicii Moldova

Universitatea Tehnică a Moldovei

Facultatea Calculatoare Informatică şi Microelectronică

Catedra Automate și Tehnologii Informaționale

**RAPORT**

Lucrarea de laborator nr. 1

laPR

Tema: Versionarea codului sursa utilizand GIT.

A efectuat:

St.gr. TI-163 Chiriac Ion

A verificat:

Asistent univ.

Chișinău -2019

Versionarea codului sursa utilizand GIT. Lucrarea de laborator are ca scop

studiul și înțelegerea principiilor de funcționare și utilizare a sistemului distribuit

de control al versiunilor numit GIT.

**Terminologie:**

* repository - pe server, conține ierarhia de fișiere și informațiile de versiune;
* working copy - varianta locală, obținuta de la server, pe care se fac

modificările;

* revision  - o versiune a unui document. (v1, v2, v3...).
* checkout - aducerea pe masina locala a versiunii de pe server, sub forma

unei working copy

* update/pull: actualizarea repozitoriului local în funcție de modificările

survenite, intre timp, pe server. Se aduc doar fișierele modificate;

* commit - înregistrează o nouă versiune a fișierului (fișierelor) modificat în

repozitoriu.

* commit message -  un mesaj asociat unei acțiuni commit care descrie

schimbările făcute în noua versiune.

* changelog - o listă a versiunilor (commit-urilor) unui fișier/proiect de obicei

însoțită de mesajele asociate fiecărui commit.

* diff: Afișează diferențele dintre două versiuni a unui fișier sau dintre fișierul

modificat local (pe working copy) și o versiune de pe repository.

* revert - renunțarea la ultimele modificări (locale) făcute într-un fișier

din working copy, și revenirea la ultima versiune aflată în repozitoriu sau la o

versiune la alegere.

* branch - creează o “copie” a unui fișier/proiect pentru modificări „în paralel”

fără a afecta starea actuală a unui proiect.

* merge - aplică ultimele modificări dintr-o versiune a unui fișier peste alt fișier;
* conflict - situația în care un merge nu se poate executa automat și modificările

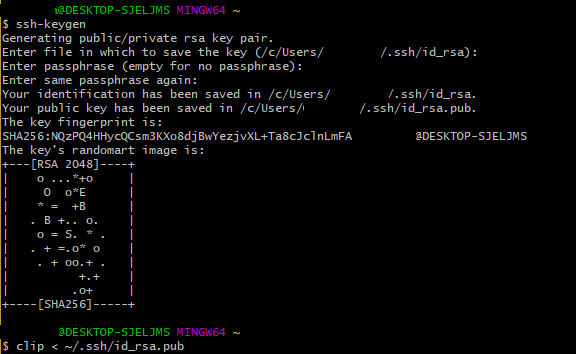
locale sunt în conflict cu modificările din repozitoriu.

* resolve: rezolvarea (de obicei manuală) a conflictelor apărute într-un fișier după

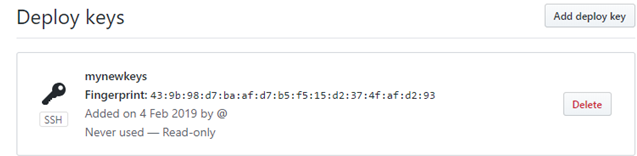
un merge.

#### Setting up our system with our GitHub data and Setting up your GitHub access key

În imaginea de mai ajos am setat în GitHub cheia de acces cu ajutorul comandei „ssh-keygen”, iar „clip < ~/.ssh/id\_rsa.pub” sa copiat cheia pentru a pune adresa pe GitHub.

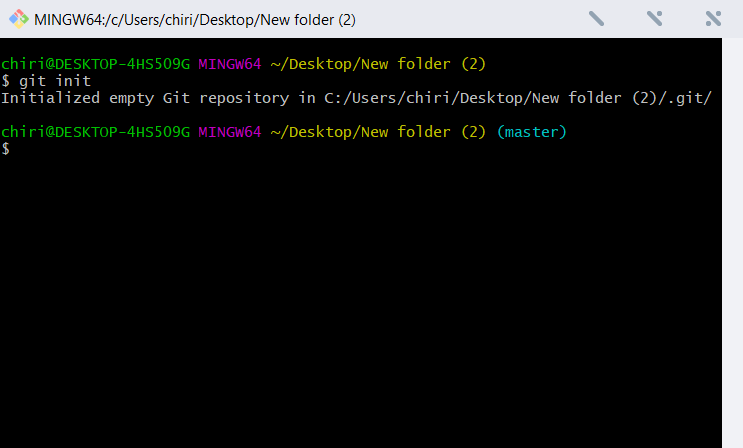


#### Am creat cheie după pașii propuși. În GitHub observam cheia creată.



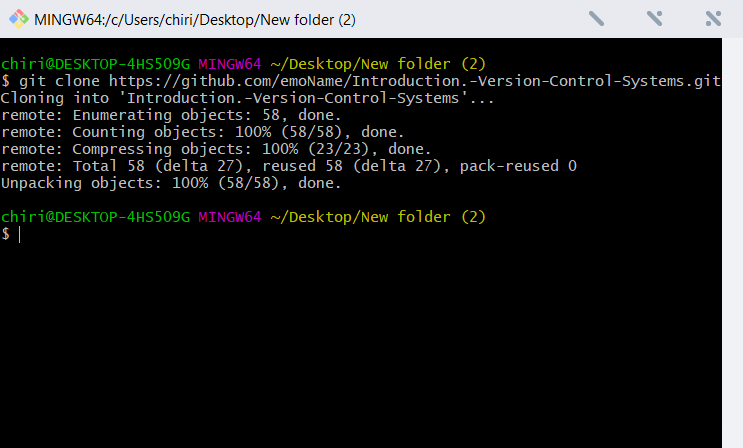
**Create your workspace**

Cu ajutorul comandei git init am creat un nou workspace



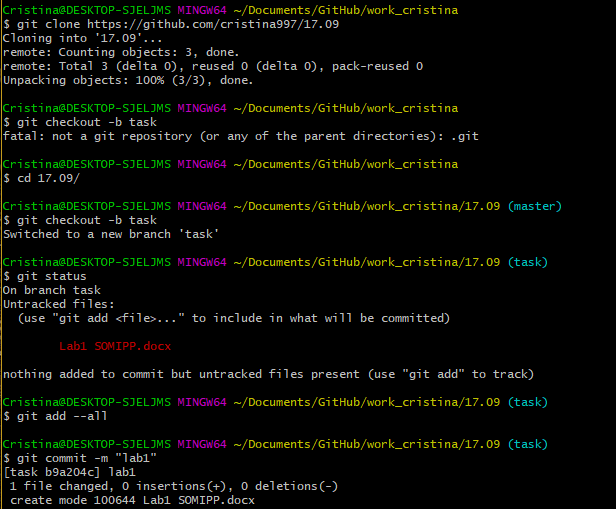
#### ****Clone your repositories****

Pasul urmator am efectuat clonarea unui repository, am copiat toate fișierele și directoarele de pe serverul GitHub. Mai jos observăm clonarea repositoriu pe care l-am creat pe GitHub .



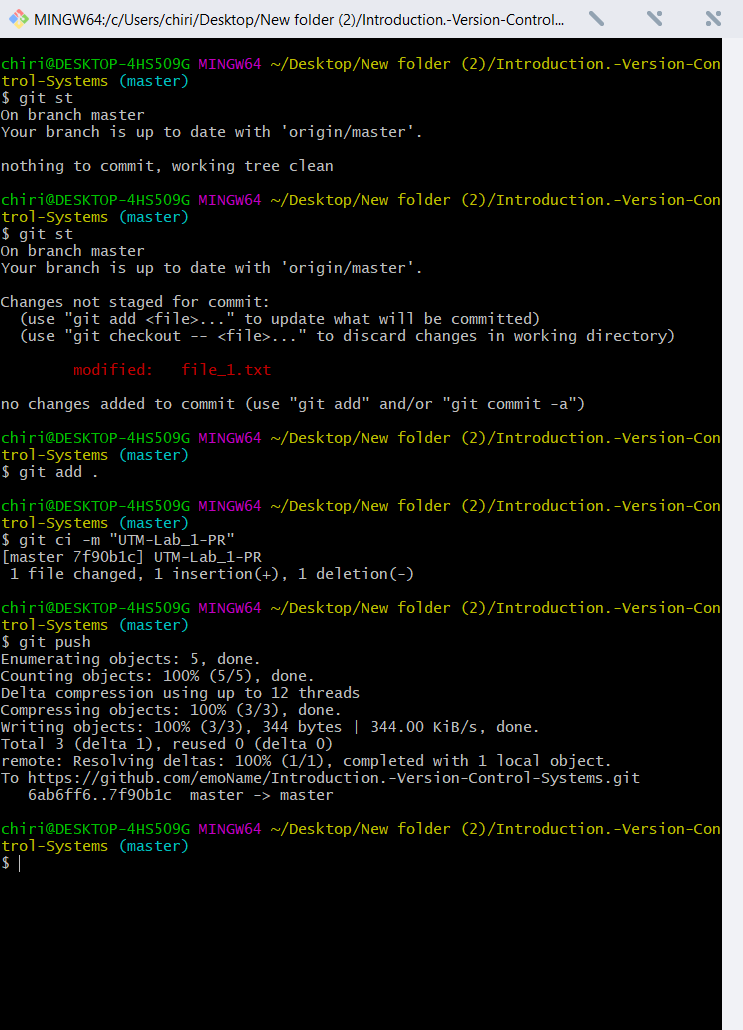
#### ****Create a branch****

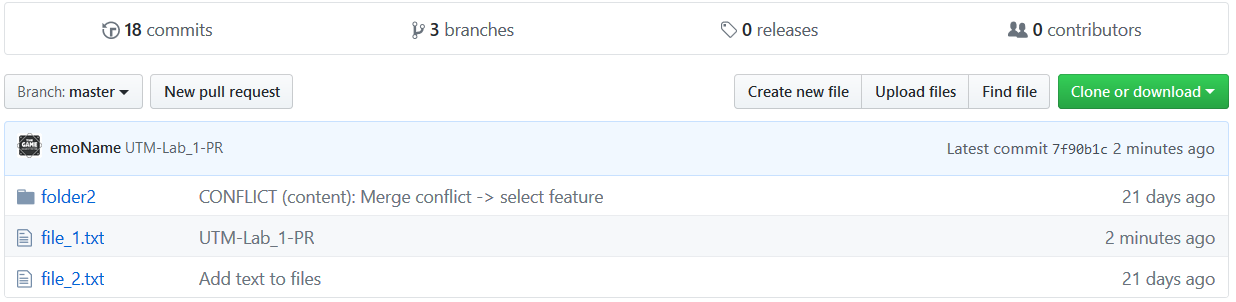
Am creat o ramura task.



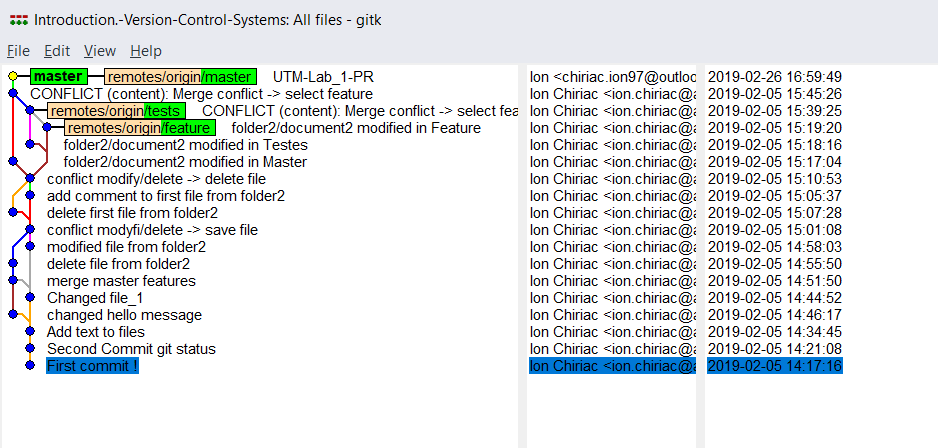
#### ****Commit the changes****

Cu ajutorul comenzii git commit înregistrăm modificările în istoria proiectului. La fel stocăm conținutul indexului în istorie împreună cu un mesaj cu ajutorul comenzii „git commit –m” ca de exemplu mai jos. Totodată cu ajutorul comenzii „git add - -all” am adaugat toate fisierele în index ,la fel am acest lucru observăm in imaginea de mai jos:





#### ****Merging the changes****



#### ****Sending it to GitHub****