Facultatea Calculatoare, Informatica si Microelectronica

Universitatea Tehnica a Moldovei

Medii Interactive de Dezvoltare a Produselor Soft

Lucrarea de laborator Nr.1

Version Control Systems si modul de setare a unui server

Efectuat : TI-151 **Golovatic Dan**

Verificat : lector asistent **Gojin Victor**

2017

1. Scopul lucrarii de laborator :

De a se invata utilizarea unui Version Control System si modul de setare a unui server.

2. Obiective

Studierea Version Control Systems (git).

Intelegerea si aplicarea comenzilor GIT.

3. Mersul lucrarii de laborator

3.1 Cerintele :

* Initializare unui nou repositoriu.

* Configurarea VCS.

* Crearea branch-urilor si commit pe ambele branch-uri

* Resetarea branch-urilor la commit-urile anterioare

* Merge la 2 branchuri.

* Folosirea fisierului .gitignore..

* Rezolvarea conflictelor.

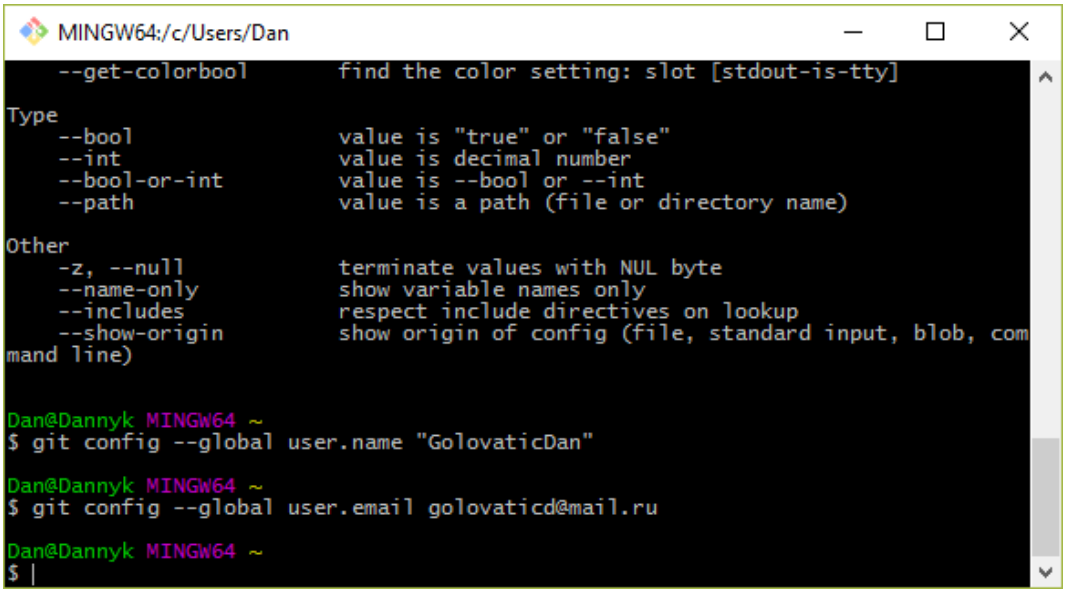
3.2 Analiza lucrarii de laborator :

Linkul repositoriului **https://github.com/GolovaticDan/MIDPS**

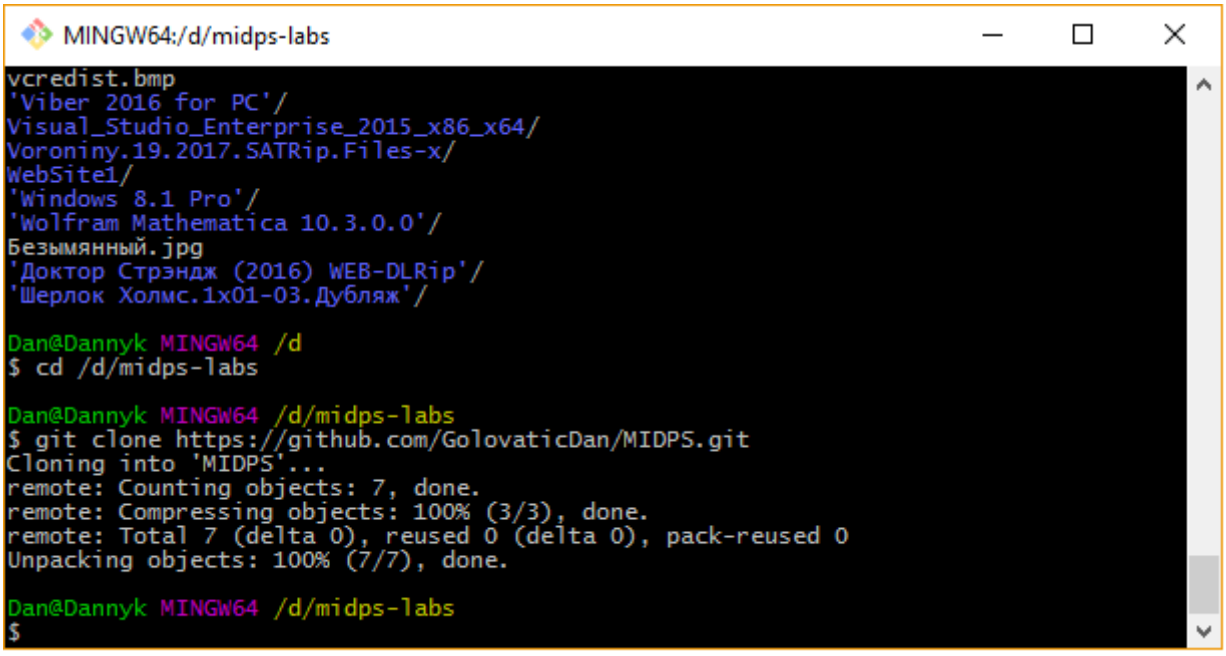
Configurarea gitului consta in mai multe etape. La inceput vom configura numele si emailul prin intermediul urmatoarelor comenzi :

git config --global user.name
“Numele” git

config --global user.email “Email”

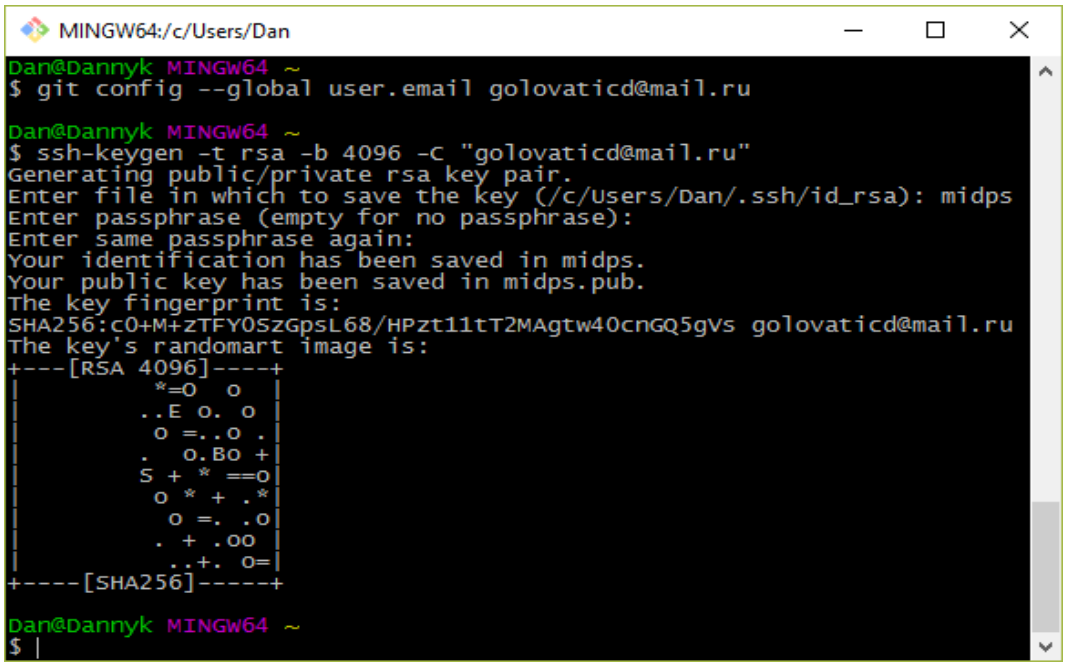


Exista mai multe metode de a crea un repozitoriu.



Urmatorul pas consta in generarea **SSH** key**.** Scriem **ssh-keygen,** iar cheia (publica) obtinuta o copiem in setarile noastre de pe github.com.

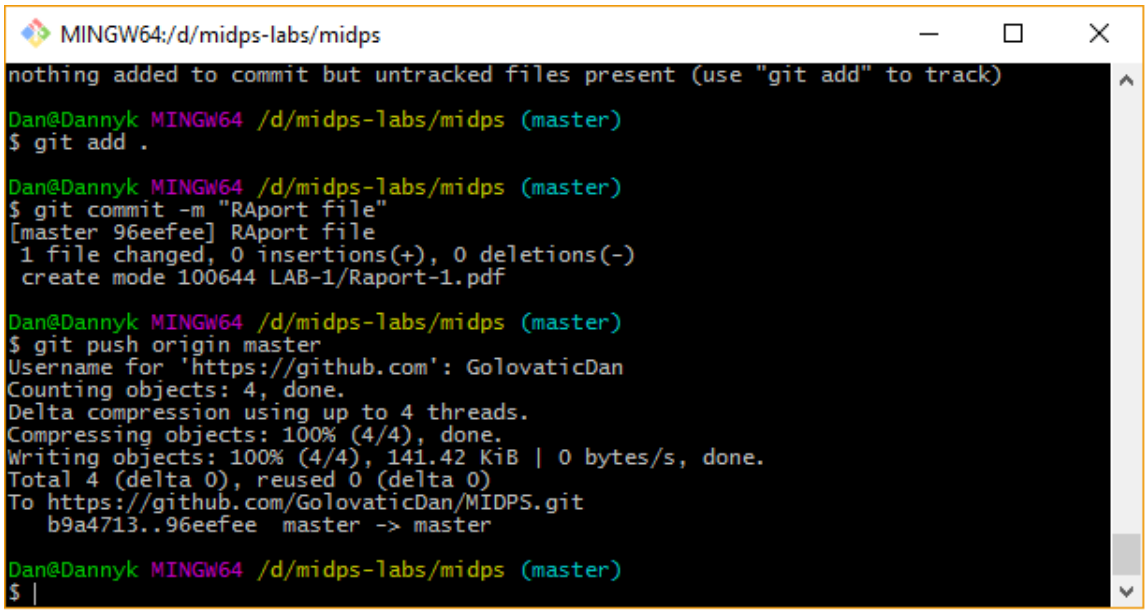
Cum e mentinut si in conditiile laboratorului, este de dorit sa initializam repozitorul nostru cu un fisier **README.md** si un **.gitignore.** In fisierul README.md vom adauga informatii pentru cei care se vor folosi de repozitoriu iar in fisierul .gitignore vom adauga toate fisierele ce trebuiesc ignorate (adica sa nu fie incarcate la moment).



Vom adauga fisierele noi create pe repozitoriul nostru. Pentru aceasta vom avea nevoie de urmatoarele comenzi :

git add * - comanda indexeaza toate fisierele.

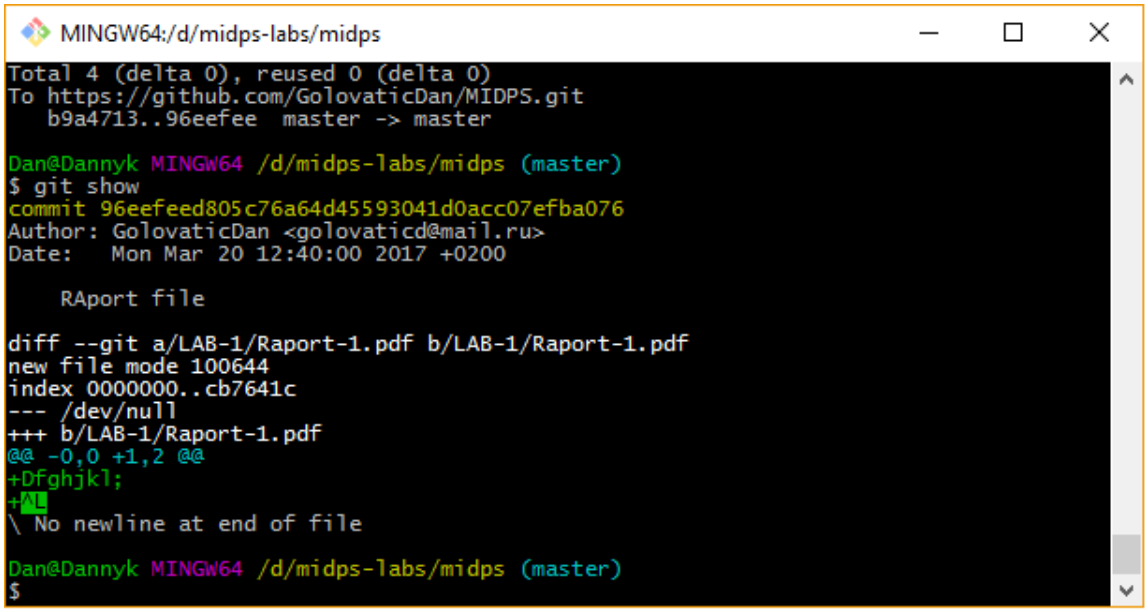
git commit –m “RAport ” – comanda face un snapshot la toate schimbarile noastre. **git push origin master –** comanda incarca toate fisierele indexate pe **github.com**



Pentru a ne asigura ca am facut totul bine si nu avem probleme utilizam urmatoarele comenzi git:

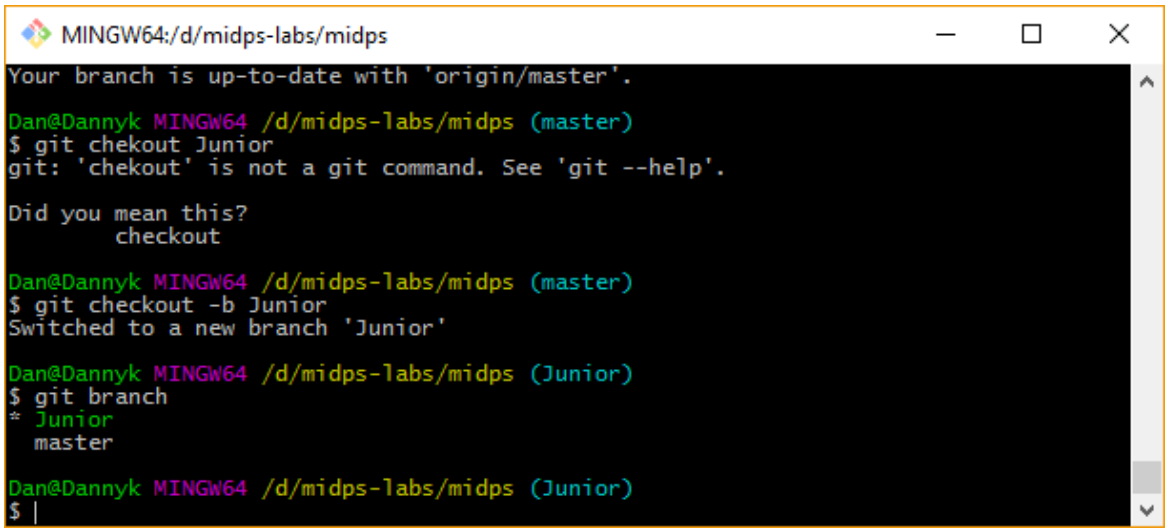
***git status**

***git show**

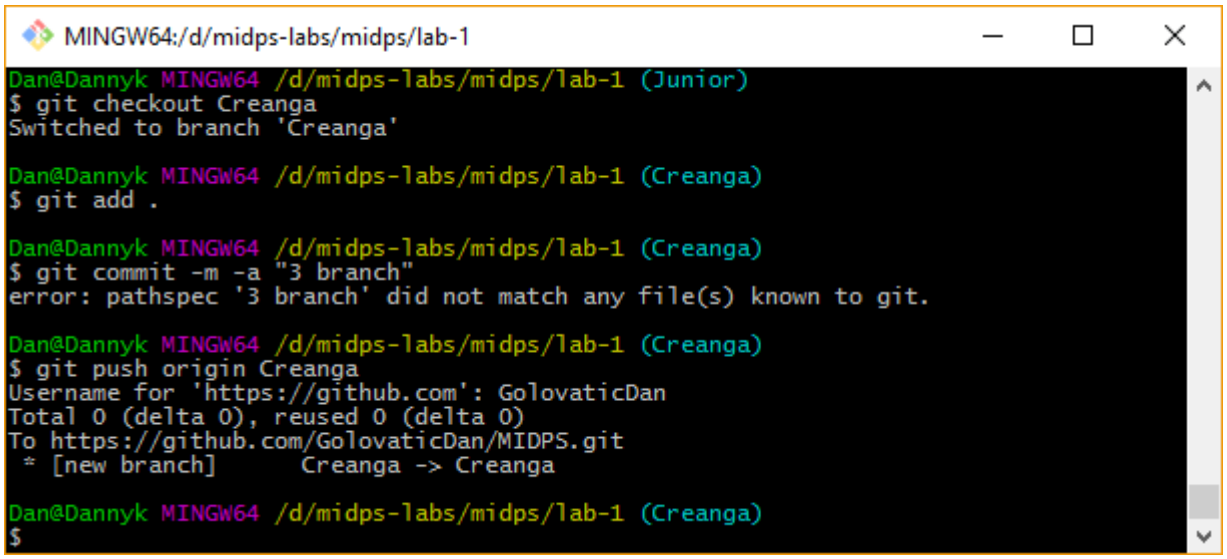


VCS ne permite sa avem mai multe **branchuri.** Din ENG branch semnifica “creanga”. Branch-urile sunt utilizate cind lucram paralel la un proiect si apoi dorim sa combinam toate modificarile.

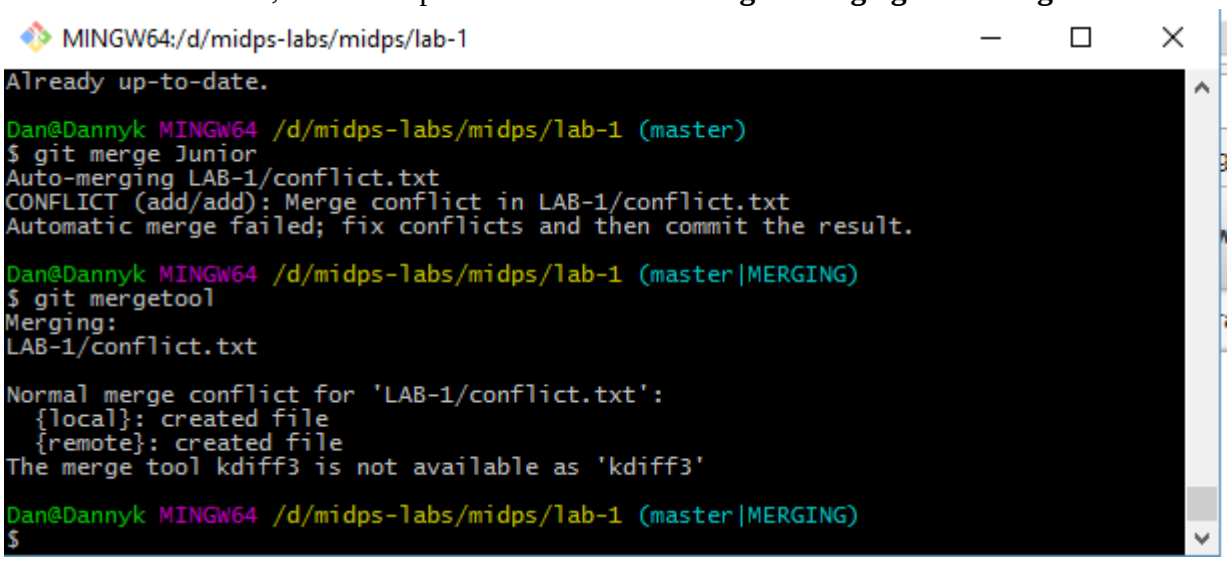
git branch “name” – creeaza un branch nou.
git branch – vizualizarea branchurilor (*indica branchul curent).
git branch –d “name” – sterge branchul “name”.
git checkout –b “name” - creeaza un branch nou cu numele “name” si face switch la el.



In caz ca dorim sa schimbat istoria unui commit, sau sa **resetam un branch la commitul anterior**. Pentru asta putem folosi comanda **git reset commit_index.** Pentru a demonstra asta am ales branchul “Creanga” de pe repo-ul meu si l-am resetat la ultimul commit facut.



Pot aparea conflicte in cazul cind dorim sa facem **merge** la 2 branch-uri si unele rinduri sunt diferite. In asa caz,pentru a elimina conflictele, folosim **mergetool**. Drept mergetool am ales **kdiff3.** Pentru kdiff3, in mod implicit folosim comanda : **git config –global merge.tool kdiff3.**



4. Concluzie

In aceasta lucrare de laborator am pus in practica si am insusit cele mai importante functii ale Git-ului. Am stiut de GitHub pina acum, insa nu am stiut ca el atitea posibilitati. VCS ne face viata de x99 ori mai usora cind lucram asupra unui produs soft. Cel mai mult ma impresionat posibilitatea de a face **branchuri**, asta chiar este ceva extreme de necesar cind asupa unui produs lucreaza mai multi oameni simultan. In urma acestei lucrari am invatat crearea si controlarea unui repozitoriu si a fisierelor din interior. La fel un avantaj este **commitul** si posibilitatea de a vedea cind si ce schimbari au fost facute. GitHub-ul este un must-learn pentru orice developer in devenire!

Referinte :

1. **https://github.com/BestMujik/MIDPS-labs/blob/master/MIDPS_LAB%231.md**

2. **https://www.siteground.com/tutorials/git/commands.htm**

3. **https://www.atlassian.com/git/tutorials/**

4. **https://git-scm.com/book/en/v2/Git-Branching-Basic**

5. **https://learn.sparkfun.com/tutorials/using-github**