

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 repositoryu.
- * 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 repositoryului **<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"
```

```
MINGW64:/c/Users/Dan
--get-colorbool    find the color setting: slot [stdout-is-tty]

Type
--bool             value is "true" or "false"
--int              value is decimal number
--bool-or-int      value is --bool or --int
--path             value is a path (file or directory name)

Other
-z, --null         terminate values with NUL byte
--name-only        show variable names only
--includes         respect include directives on lookup
--show-origin      show origin of config (file, standard input, blob, command line)

Dan@Dannyk MINGW64 ~
$ git config --global user.name "GolovaticDan"

Dan@Dannyk MINGW64 ~
$ git config --global user.email golovaticd@mail.ru

Dan@Dannyk MINGW64 ~
$ |
```

Exista mai multe metode de a crea un repository.

```
MINGW64:/d/midps-labs
vcredist.bmp
'Viber 2016 for PC'/
Visual_Studio_Enterprise_2015_x86_x64/
Voroniny.19.2017.SATRip.Files-x/
WebSite1/
'Windows 8.1 Pro'/
'Wolfram Mathematica 10.3.0.0'/
Безымянный.jpg
'Доктор Стрэндж (2016) WEB-DLRip'/
'Шерлок Холмс.1x01-03.Дубляж'/

Dan@Dannyk MINGW64 /d
$ cd /d/midps-labs

Dan@Dannyk MINGW64 /d/midps-labs
$ git clone https://github.com/GolovaticDan/MIDPS.git
Cloning into 'MIDPS'...
remote: Counting objects: 7, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (7/7), done.

Dan@Dannyk MINGW64 /d/midps-labs
$
```

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 repositoryul nostru cu un fisier **README.md** si un **.gitignore**. In fisierul README.md vom adauga informatii pentru cei care se vor folosi de repository iar in fisierul .gitignore vom adauga toate fisierele ce trebuiesc ignorate (adica sa nu fie incarcate la moment).

```
MINGW64:/c/Users/Dan
Dan@Dannyk MINGW64 ~
$ git config --global user.email golovaticd@mail.ru

Dan@Dannyk MINGW64 ~
$ ssh-keygen -t rsa -b 4096 -C "golovaticd@mail.ru"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Dan/.ssh/id_rsa): midps
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in midps.
Your public key has been saved in midps.pub.
The key fingerprint is:
SHA256:c0+m+zTFY0SzGpsL68/HPzt11tT2MAGtw40cngQ5gVs golovaticd@mail.ru
The key's randomart image is:
+---[RSA 4096]-----+
|
|  =O  O
| ..E O. O
| O =..O .
| . O.B O +
| S + * ==O
| O * + . *
| O =. . O
| . + .OO
| ..+. O=
+---[SHA256]-----+
Dan@Dannyk MINGW64 ~
$ |
```

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

```
MINGW64:/d/midps-labs/midps
nothing added to commit but untracked files present (use "git add" to track)
Dan@Dannyk MINGW64 /d/midps-labs/midps (master)
$ git add .

Dan@Dannyk MINGW64 /d/midps-labs/midps (master)
$ git commit -m "RAport file"
[master 96eefee] RAport file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 LAB-1/Raport-1.pdf

Dan@Dannyk MINGW64 /d/midps-labs/midps (master)
$ git push origin master
Username for 'https://github.com': GolovaticDan
Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 141.42 KiB | 0 bytes/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/GolovaticDan/MIDPS.git
b9a4713..96eefee master -> master

Dan@Dannyk MINGW64 /d/midps-labs/midps (master)
$ |
```

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

***git status**

***git show**

```
MINGW64:/d/midps-labs/midps
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/GolovaticDan/MIDPS.git
b9a4713..96eefee master -> master

Dan@Dannyk MINGW64 /d/midps-labs/midps (master)
$ git show
commit 96eefeed805c76a64d45593041d0acc07efba076
Author: GolovaticDan <golovaticd@mail.ru>
Date: Mon Mar 20 12:40:00 2017 +0200

    Raport file

diff --git a/LAB-1/Raport-1.pdf b/LAB-1/Raport-1.pdf
new file mode 100644
index 0000000..cb7641c
--- /dev/null
+++ b/LAB-1/Raport-1.pdf
@@ -0,0 +1,2 @@
+Dfghjkl;
+
\ No newline at end of file

Dan@Dannyk MINGW64 /d/midps-labs/midps (master)
$
```

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.

```
MINGW64:/d/midps-labs/midps
Your branch is up-to-date with 'origin/master'.

Dan@Dannyk MINGW64 /d/midps-labs/midps (master)
$ git chekout Junior
git: 'chekout' is not a git command. See 'git --help'.

Did you mean this?
    checkout

Dan@Dannyk MINGW64 /d/midps-labs/midps (master)
$ git checkout -b Junior
Switched to a new branch 'Junior'

Dan@Dannyk MINGW64 /d/midps-labs/midps (Junior)
$ git branch
* Junior
  master

Dan@Dannyk MINGW64 /d/midps-labs/midps (Junior)
$ |
```

In caz ca dorim sa schimbam 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.

```
MINGW64:/d/midps-labs/midps/lab-1
Dan@Dannyk MINGW64 /d/midps-labs/midps/lab-1 (Junior)
$ git checkout Creanga
Switched to branch 'Creanga'

Dan@Dannyk MINGW64 /d/midps-labs/midps/lab-1 (Creanga)
$ git add .

Dan@Dannyk MINGW64 /d/midps-labs/midps/lab-1 (Creanga)
$ git commit -m -a "3 branch"
error: pathspec '3 branch' did not match any file(s) known to git.

Dan@Dannyk MINGW64 /d/midps-labs/midps/lab-1 (Creanga)
$ git push origin Creanga
Username for 'https://github.com': GolovaticDan
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/GolovaticDan/MIDPS.git
 * [new branch]      Creanga -> Creanga

Dan@Dannyk MINGW64 /d/midps-labs/midps/lab-1 (Creanga)
$
```

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**.

```
MINGW64:/d/midps-labs/midps/lab-1
Already up-to-date.

Dan@Dannyk MINGW64 /d/midps-labs/midps/lab-1 (master)
$ git merge Junior
Auto-merging LAB-1/conflict.txt
CONFLICT (add/add): Merge conflict in LAB-1/conflict.txt
Automatic merge failed; fix conflicts and then commit the result.

Dan@Dannyk MINGW64 /d/midps-labs/midps/lab-1 (master|MERGING)
$ git mergetool
Merging:
LAB-1/conflict.txt

Normal merge conflict for 'LAB-1/conflict.txt':
  {local}: created file
  {remote}: created file
The merge tool kdiff3 is not available as 'kdiff3'

Dan@Dannyk MINGW64 /d/midps-labs/midps/lab-1 (master|MERGING)
$
```

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 usura 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 repository si a fișierelor 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>