

Atvirojo kodo įranga

Laboratorinis darbas 2

AKSfm-16 gr. stud. Justin Byčkovič

Data: 2017-09-28

1. Repozitorijos sukūrimas

Laboratoriame darbe naudojama 1 laboratorinio virtuali mašina. Serveris ir klientai sukurimai vienoje sistemoje ir yra logiskai atskiriami. Serveris – sistemos (kompiuterio) vartotojo sukūrimu, o klientai – GIT repozitorijoje konfiguruojant lokalų vartotoją. Pirmojo vartotojo ir serverio repozitorijų kūrimas pateiktas žemiau.

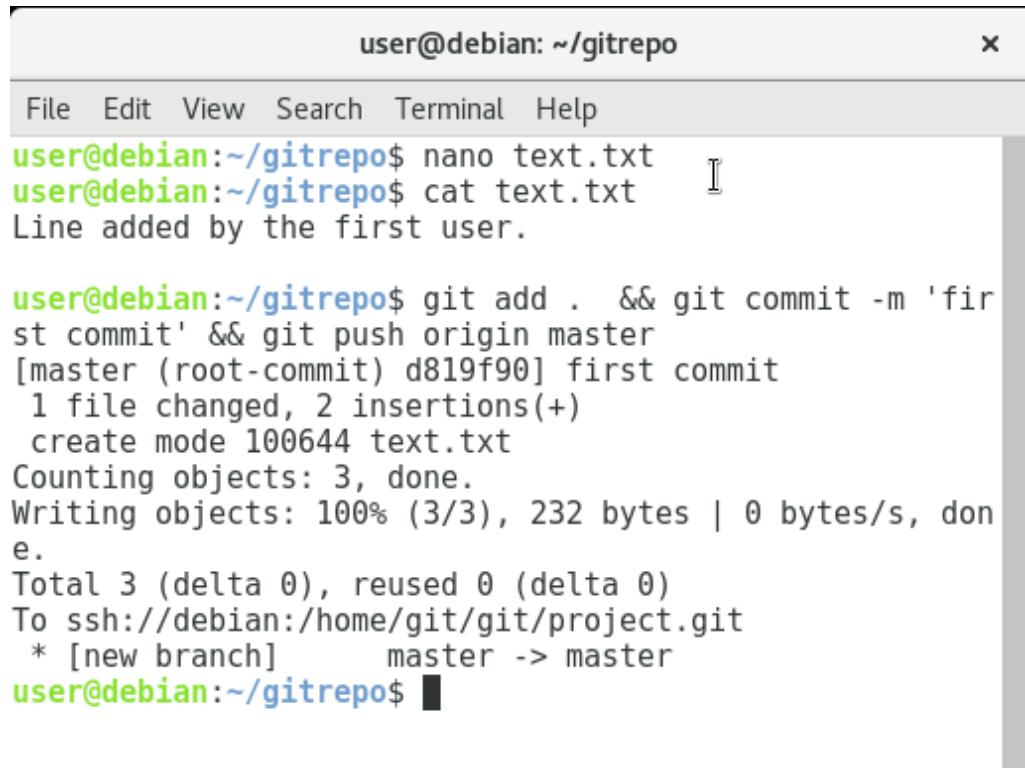
Siekiant dalintis failais tarp „client_side/gitrepo“ ir „client2_side/gitrepo“ repozitorijų procese yra sukuriamas serveris. Siekiant susisiekti su vartotoju yra įdiegiamas ir konfiguruojamas SSH serveris. Nustatymas pateiktas žemiau:

1. Įdiegiamas SSH serverio klientas skirtas GIT serverio darbui su SSH klientu. Komanda „sudo apt-get install openssh-server“. Naudojant SSH Git nuotolinię serverio nuoroda pavadinimu „origin“ yra sudaroma taip:
„ssh://kompiuterio_vartotojas@kompiuteris:kelias_iki_katalogo“.
2. SSH „sshd_config“ faile nukomentuojama ir nustatomas slaptažodžio užklausymas į „no“, nes jungiantis prie GIT serverio naudojamas atviras raktas.
3. Serveryje kuriamas „git“ kompiuterio vartotojas komanda:
„sudo adduser git“. Ir užteinama tuo vartotoju „su git“ komanda.
4. „~/.ssh/id_dsa.pub“ viešasis raktas keliamas į serverio kompiuterį.
Tuo tikslu vartotojo kompiuteryje generuojamas SSH prisijungimo viešasis ir privatus raktai „ssh-keygen“ komanda.
5. Serverio kompiuteryje „~/.ssh“ kataloge kuriamas „authorized_keys“ failas, saugoti viešiems klientų raktams. „authorized_keys“ turi būti prieinamas skaityti „root“. Kliento atviras raktas rašomas į failo galą šia komanda:
„cat /tmp/is_sda.pub >> authorized_keys“.
6. Vykdama komanda „mkdir ~/gitrepo && cd ~/gitrepo && git init --bare“ sukurti „server_side/gitrepo“ GIT repozitoriją. “--bare” konfigūracija naudojama sukurti repozitoriją be failų.

GIT kliento kūrimas ir konfigūravimas yra sekantis žingsnis atliekamas laboratorinio darbo pirmą dalį. Šis žingsnis atliekamas visiems vartotojų kompiuteriams prisijungiantiems prie serverio. Darbo metu SSH kliento įdiegiamas neberekalingas, nes įdiegtas su serveriu, ir konfigūravimas pateiktas aukščiau. Todėl „client_side/gitrepo“ repozitorija diegama šiais žingsniais:

1. Kliento kompiuteryje sukūrimas „client_side/gitrepo“ šiomis komandomis:
„mkdir ~/gitrepo && git init“.
2. Nustatomas lokalus GIT repozitorijos vartotojo vardas ir jo el. paštas.
Naudojamos komandos:

- Vardui – „git config --local user.name vardas“;
 - El. paštui – „git config --local user.email el_pastas“.
- Nurodoma serverio repozitorijos nuoroda vykdyti „push“ ir „pull“: „git remote add origin ssh://git@debian:/home/git/gitrepo“.
 - Pirmajam klientui sukūriamas „text.txt“ failas užrašu „Text file created by the first user.“ ir failas keliamas į serverį komandų rinkiniu (pridėti failą, prisegti pakeitimus ir įkelti) „ git add . && git commit -m ,first commit“ && git push origin master“. Rezultatas pateiktas 1 pav.



```

user@debian: ~/gitrepo
File Edit View Search Terminal Help
user@debian:~/gitrepo$ nano text.txt
user@debian:~/gitrepo$ cat text.txt
Line added by the first user.

user@debian:~/gitrepo$ git add . && git commit -m 'first commit' && git push origin master
[master (root-commit) d819f90] first commit
 1 file changed, 2 insertions(+)
 create mode 100644 text.txt
Counting objects: 3, done.
Writing objects: 100% (3/3), 232 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To ssh://debian:/home/git/git/project.git
 * [new branch]      master -> master
user@debian:~/gitrepo$ 

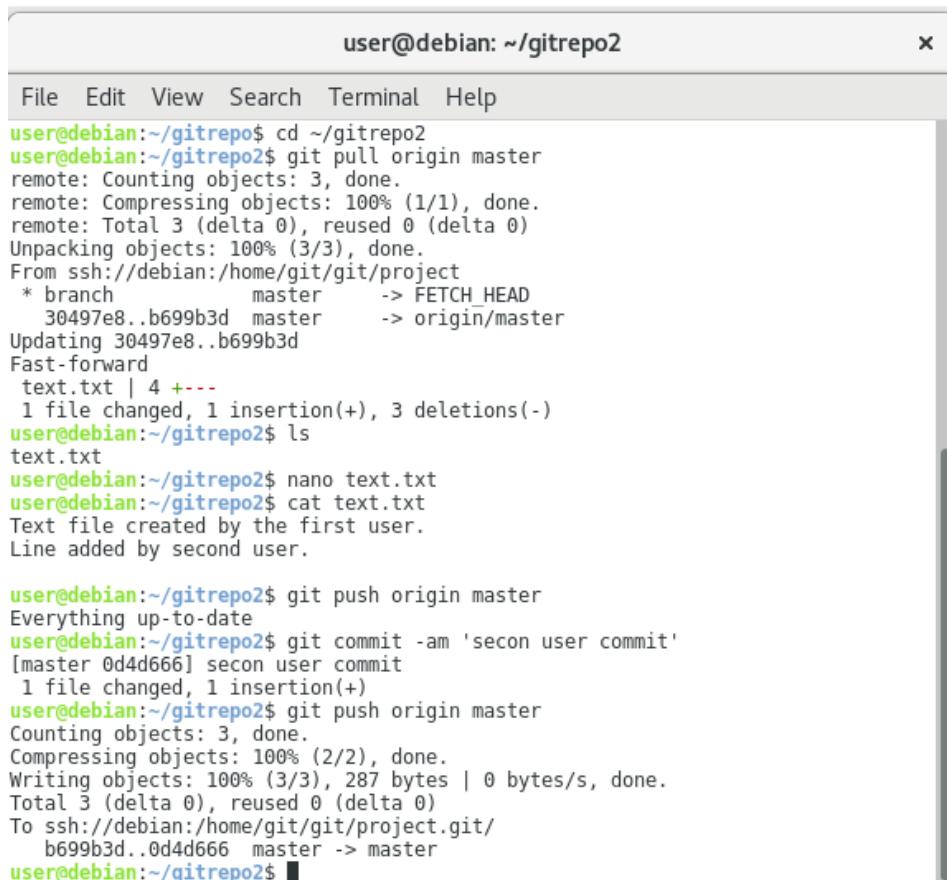
```

1 pav. „text.txt“ failo sukūrimas ir išsaugojimas į serverį

2. Gavimas duomenų iš repozitorijos ir pakeitimų darymas

Laboratorinio darbo antrajam punktui atlikti buvo sukurta antrojo vartotojo „home/user/gitrepo2“ repozitorija, kuri buvo skirta bendradarbiaujant keisti „text.txt“. Kurti antrąjį vartotojo repozitoriją „client_side2/gitrepo“ buvo pakartoti „client_side/gitrepo“ kūrimo 1 – 3 punktai. Vartotojo vardas ir el. paštas yra unikalūs repozitorijai tai yra lokalus.

Kaip reikalautų užduotyje įvykdytas pirmojo vartotojo failo pakeitimai (žiūreti 2 pav.). Pirma atsiunčiami duomenys iš serverio išsaugoti pirmojo vartotojo. „nano“ redaktoriuje pridedama eilutę „Line added by second user.“ Išsaugojus pakeitimus failas keliamas į serverio repozitoriją.



```
user@debian:~/gitrepo2
File Edit View Search Terminal Help
user@debian:~/gitrepo2$ cd ~/gitrepo2
user@debian:~/gitrepo2$ git pull origin master
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (1/1), done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From ssh://debian:/home/git/git/project
 * branch           master      -> FETCH_HEAD
   30497e8..b699b3d  master      -> origin/master
Updating 30497e8..b699b3d
Fast-forward
 text.txt | 4 +---
 1 file changed, 1 insertion(+), 3 deletions(-)
user@debian:~/gitrepo2$ ls
text.txt
user@debian:~/gitrepo2$ nano text.txt
user@debian:~/gitrepo2$ cat text.txt
Text file created by the first user.
Line added by second user.

user@debian:~/gitrepo2$ git push origin master
Everything up-to-date
user@debian:~/gitrepo2$ git commit -am 'secon user commit'
[master 0d4d666] secon user commit
 1 file changed, 1 insertion(+)
user@debian:~/gitrepo2$ git push origin master
Counting objects: 3, done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 287 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To ssh://debian:/home/git/git/project.git/
 b699b3d..0d4d666  master -> master
user@debian:~/gitrepo2$
```

2 pav. „client2_side/gitrepo“ repozitorijoje projekto kėlimas,

Modifikavimas ir išsaugojimas į serverį procesas

3. Konfliktų sprendimas GIT sistemoje

Atliekant laboratorinio darbo trečią punktą susidarė konfliktas, nes „server_side/gitrepo“ repozitorijoje buvo pridėta eilutė „Line added by the first user.“ ankščiau „client2_side/gitrepo“ vartotojo, kurioje pridėta eilutė yra „Another line added by the second user.“

Konfliktas kilo antrajam vartotojui bandant kelti pakeitimus į serverį (žiūrėti 3 pav.). Kaip pateikta 3 pav. konfliktą GIT pateikia tekstinių pavidalų „text.txt“ ir šis išsprendžiamas antrojo vartotojo kompiuteryje modifikuojant failą. Modifikuotas failas yra pateiktas 4 pav. ir šis „client_side/gitrepo“ yra atnaujinamas iš „server_side/gitrepo“, nes konfliktas sprendžiamas dviem vartotojams.

```
user@debian: ~/gitrepo2
File Edit View Search Terminal Help
user@debian:~/gitrepo2$ git push origin master
To ssh://debian:/home/git/git/project.git/
  ! [rejected]          master -> master (fetch first)
error: failed to push some refs to 'ssh://git@debian:/h
ome/git/git/project.git/'
hint: Updates were rejected because the remote contains
work that you do
hint: not have locally. This is usually caused by another
repository pushing
hint: to the same ref. You may want to first integrate
the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push -
-help' for details.
user@debian:~/gitrepo2$ git pull origin master
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From ssh://debian:/home/git/git/project
 * branch            master    -> FETCH_HEAD
   a333733..8da8d4e  master    -> origin/master
Auto-merging text.txt
CONFLICT (content): Merge conflict in text.txt
Automatic merge failed; fix conflicts and then commit t
he result.
user@debian:~/gitrepo2$ cat text.txt
Text file created by the first user.
Line added by second user.
<<<<< HEAD
Another line added by the second user.
=====
Line added by the first user.
>>>>> 8da8d4e560debf21celaed88ald6b980af9e6d9f

user@debian:~/gitrepo2$
```

3 pav. „client2_side/gitrepo“ su „server_side/gitrepo“ konflikto aptikimas

```
user@debian: ~/gitrepo
File Edit View Search Terminal Help
user@debian:~/gitrepo2$ nano text.txt
user@debian:~/gitrepo2$ nano text.txt
user@debian:~/gitrepo2$ cat text.txt
Text file created by the first user.
Line added by second user.
Line added by the first user.
Another line added by the second user.

user@debian:~/gitrepo2$ git add .
user@debian:~/gitrepo2$ git commit -m 'conflict resolved'
[master 94a69f2] conflict resolved
user@debian:~/gitrepo2$ git push origin master
Counting objects: 6, done.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 536 bytes | 0 bytes/s, done.
Total 6 (delta 2), reused 0 (delta 0)
To ssh://debian:/home/git/git/project.git/
  8da8d4e..94a69f2  master -> master
user@debian:~/gitrepo2$ cd ~/gitrepo && git pull origin
master
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 2), reused 0 (delta 0)
Unpacking objects: 100% (6/6), done.
From ssh://debian:/home/git/git/project
 * branch            master      -> FETCH_HEAD
   8da8d4e..94a69f2  master      -> origin/master
Updating 8da8d4e..94a69f2
Fast-forward
 text.txt | 1 +
 1 file changed, 1 insertion(+)
user@debian:~/gitrepo$
```

4 pav. Konflikto sprendimas „client2_side/gitrepo“ ir pateikimas „clien_side/gitrepo“ repozitorijoje

4. Išvados

Laboratorinio darbo metu buvo išmokta kurti GIT serverį ir klientą. Didžiaja darbo dalį užėmė darbas su kliento repozitorijomis ir projekto darbo imitacija. Buvo išmokti pagrindai dirbtų projektuose grupėje naudojant GIT versijavimo sistemą. Prosesu buvo keičiamas ir modifikuojamas „text.txt“ failas. Atlirkas darbas buvo panašus į pakeitimų sąrašą vedamą vartotojų. Dėl šios priežasties kilęs konfliktas „client_side2/gitrepo“ repozitorijoje išspręstas laiko vedimo tvarka. Todėl darbo metu „server_side/gtirepo“ yra pagrindinė visiems klientams.