

Versioonihaldustarkvara GIT kasutusjuhend (käsurea kasutamine)

Sisukord

Versioonihaldustarkvara GIT kasutusjuhend (käsurea kasutamine).....	1
Algus.....	1
Git salve (repositooriumi) kloonimine ehk kohaliku koopia tegemine.....	1
Failide lisamine Git salve versioonihaldusse.....	2
Muudatuste registreerimine (commit).....	2
Registreerimiste üleslaadimine.....	3
Projekti kloonimine teise arvutisse või teise kataloogi (kontrollimine, mis Giti salves siis ikkagi on).....	3

Algus

Alustuseks oled loonud kataloogi ja sellesse failid, mida kodutöös nõuti. Lisaks oled õppejõult saanud juhised, milline on sinu privaatse GITi salve (repositooriumi) aadress serveris git.ttu.ee.

Eeldame, et meil on nüüd vaja esitada projekt1, milles on fail src/projekt1/Vinni.java:

Soovime selle projekti üles laadida salve aadressiga

`https://controlina.alt-delete@git.ttu.ee/kursused/iti0011/controlina.alt-delete.git`

Selles Git salve aadressis on enne ampersandi '@' Uni-ID kasutajanimi. Serveris asuvas ainele mõeldud kataloogis on juba valmis tehtud vastav Git salv, mis on minu Uni-ID kasutajaminega.

Git salve (repositooriumi) kloonimine ehk kohaliku koopia tegemine

Eeldame, et arvutisse on installeeritud GIT tarkvara aadressilt <http://git-scm.com/>. Käivitame programmi Git Bash (Linuxis/Mac OSX-s lihtsalt terminali). Seal saab käivitada käsku git.

Kloonime antud salve kohalikule kettale, et saaksime hakata sinna oma projekte lisama. Selleks liigume terminalis käsuga cd sobivasse kataloogi näiteks [c:\salved](#):

```
cd /c/salved
```

Seejärel kloonime salve serverist kohalikule kettale:

```
git clone https://controlina.alt-delete@git.ttu.ee/kursused/iti0011/controlina.alt-delete
```

Selle tagajärjel tekib tühi salv kataloogi `c:\salved\controlina.alt-delete`

Enne, kui olete repositooriumi midagi üles laadinud, on antud repositoorium tühi.

Failide lisamine Git salve versioonihaldusse

Nüüd, kus oleme oma salve värskelt klooninud kataloogi /c/salved/controlina-alt-delete, saame hakata sinna faile lisama. Selleks kopeerime failid kõigepealt salve kataloogi

(Git Bashist saab kopeerida käsuga cp -a lähtekataloog sihtkataloog)

Eeldame, et algselt sisaldub meie fail src/projekt1/Vinni.java kataloogis /c/projektid/kodutoo1/

```
cp /c/projektid/kodutoo1 /c/salved/controlina.alt-delete
```

Seejärel liigume juba sinna repositooriumi kataloogi:

```
cd /c/salved/controlina.alt-delete
```

Nüüd käivitame käsu

```
git add src/projekt1/Vinni.java
```

NB! Lisage vaid faile! Kataloogid lisatakse automaatselt. Kataloogide git add käsuga lisamisel võivad tekkida probleemid.

Versioonihalduse toimingud on

- failide lisamine (add)
- lisatud failide registreerimine repositooriumis (commit)
- registreerimiste üleslaadimine (push)
- registreerimiste allalaadimine (pull)

Muudatuste registreerimine (commit)

Failide lisamine ütles Gitile, et nüüd siin on uus fail / uued failid, mille üle tuleks hakata arvet pidama. Commit on käsk, millega registreeritakse käsu käivitamise hetke seis versioonihalduse kohalikus koopias.

Kui käivitate käsu

```
git commit
```

Siis käivitub vaikimisi määratud tekstiredaktor. Windowsis, Git Bashis on selleks tihti vim nimeline tekstiredaktor, millega on alguses suhteliselt harjumatu ümber käia.

Muudatuse registreerimist tuleb kirjeldada lühikese, aga täpse tekstida. Tekstiredaktoris git commit käsu peale avanenud failis tuleb selline kirjeldus lisada faili algusesse. Vim-s saab seda teha järgmiselt:

- 1) vajutage klahvi "i". Seejärel saab hakata teksti sisestama.
- 2) Kui tekst on sisestatud, siis "ESC" siis ':' siis 'x' ja siis "ENTER".

Et vältida maadlust vim-ga võib teate lisada käsureale võtme '-m' järele. Näiteks nii:

```
git commit src/Projekt1/Vinni.java -m "Lisatud lahendus 1 kodutööle"
```

Kui korraga registreeritakse muutused paljudes failides, siis võib pruukida ka käsku:

```
git commit -a -m "Kodutöö 1 valmis"
```

Kui lisate Git salve Eclipse projekte, siis on mõistlik lisada ka .project fail.

Registreerimiste üleslaadimine

Esimene kord muudatusi üles laadides tuleb Gitile kindlasti öelda, millisesse harusse serveris paiknevas salves tuleb muudatused saata.

Eeldame jällegi, et oleme Git salve kataloogis /c/salved/controlina.alt-delete

Muudatuste üleslaadimine käib käsuga "push":

```
git push origin master
```

Selle käsuga öeldakse, et muudatused tuleb laadida üles Git salve algasukohta (origin) peamisse harusse (master).

Projekti kloonimine teise arvutisse või teise kataloogi (kontrollimine, mis Giti salves siis ikkagi on)

Kui muudatuste registreerimised on laetud git.ttu.ee serveris paiknevasse salve, siis saab neid kerge vaevaga kloonida ka teistes arvutites ja sama arvuti teistesse kataloogidesse.

Oletame, et liikusime kataloogi [c:\kontrollsalved](#)

```
cd /c/kontrollsalved
```

nüüd kloonime salve uuesti kohalikku arvutisse uude kohta (või siis teise arvutisse):

```
git clone https://controlina.alt-delete@git.ttu.ee/kursused/iti0011/controlina.alt-delete
```

Sedasi näeme, mis failid serveris antud salves olemas on.

Kui salv on kord juba kloonitud, siis on võimalik käsuga "pull" tõmmata serverist vaid viimased muudatused ilme kogu salve kloonimata.

Serverist muudatuste sikutamine käib käsuga

```
git pull
```

Kui kohalikus arvutis on juhtumisi muudetud sama faili, mis teises arvutis (tavaline olukord meeskonnas arendades), siis käsk git pull püüab muudatused intelligentselt kokku sobitada. Kui see ei õnnestu, siis läheb salv konfliktiolekusse. Selliste olukordade lahendamiseks lisandub dokumentatsioon hiljem.