



Mittuniversitetet

MID SWEDEN UNIVERSITY

Metoder och verktyg i mjukvaruprojekt

Laboration 1 – Versionshantering med Git

Laborationen syftar till att ge praktisk erfarenhet av versionshantering med Git!

Innehållsförteckning

Laboration 1.....3

 Uppgift 1.....3

 Uppgift 2.....4

 Uppgift 3.....5

Redovisning.....6

Laboration 1

*Innan du börjar med laborationen bör du ha läst igenom lektion 1-2 samt anvisningarna i kursboken **Pro Git**. Läs igenom hela laborationsbeskrivningen med alla moment innan du påbörjar arbetet med den första uppgiften!*

Laborationen består av tre deluppgifter men innan du påbörjar första uppgiften så kan du gärna göra **Git-quizen** i *Moodle* som handlar om hur man använder **Git** i terminalen / kommandotolken. Denna quizz ger dig en fingervisning kring den förståelse du behöver för att kunna gå vidare. Innan du gör quizen bör du ha studerat de moment i kursboken som behandlar de grundläggande kommandon som tas upp i **kapitel 1** till **3**.

Uppgift 1

Nu när du kommit igång med Git och *Bitbucket* är det dags att skapa ditt första **repository** i ditt privata Bitbucket konto!

1. Skapa ett nytt **privat repository** på ditt personliga *Bitbucket*-konto namnge det med ditt student id!
2. Hämta det repo som du nyss skapat till din lokala dator!
3. Skapa en **markdown README**-fil (lokalt) i en texteditor och se till att lägga denna under repots rotkatalog. OBS! Filen skall heta **README.md** och innehålla lite info om dig själv samt den utrustning du använder i kursen, *operativsystem*, *git version*, *C++ kompilator* osv!
4. Spara ändringarna lokalt med lämpligt meddelande.
5. Synkronisera de nya ändringarna med repots **remote origin** på *Bitbucket*.

Uppgift 2

Eftersom lösningar enbart skall redovisas genom de repos som finns under kursens gemensamma **Bitbucket Team** så skall du nu ändra repots **remote origin** till ditt tilldelade repo i detta team:

```
# show existing remote
$ git remote -v

# remove existing remote origin
$ git remote rm origin

# set new remote origin (replace studid with your student id)
$ git remote add origin git@bitbucket.org:miun_dt042g/studid_laboration_1_vt21.git
```

När du är färdig med detta så kan du verifiera att **remote origin** verkligen har ändrats och därefter med gott samvete ta bort det gamla repot från ditt personliga *Bitbucket*-konto eftersom det ej längre används som **remote origin**.

För att uppdatera den nya fjärran källan behöver vi utföra en **push** tillsammans med en flagga för den **upstream** som skall användas:

```
# set an upstream to the new origin
$ git push --set-upstream origin master
```

Ajdå, **push rejected**... Baserat på den information Git ger oss verkar det som att innehållet av källorna inte överrenstämmer och mycket riktigt så kan vi, genom exempelvis Bitbuckets webbgränssnitt, se att den fjärran källan har en **README.md** i en underkatalog istället för sin rot.

Genom att lägga till flaggan **--force** skulle denna **push** kunna genomföras, men inte på ett önskvärt sätt eftersom origins aktuella innehåll då kommer att gå förlorat till förmån åt det lokala innehållet.

För att bevara allt innehåll måste källorna synkroniseras med varandra på ett korrekt sätt. Utifrån de ledtrådar vi får via konsolmeddelanden från Git bör vi uppdatera den lokala källan innan vi utför en push, men skriver vi nu kommandot...

```
# attempt to update local source with content of remote
$ git pull origin master
```

... så får vi ett nytt felmeddelande som anger att sammanföring inte kan genomföras på grund av **icke relaterad historik**. Något behöver därför läggas till i detta kommando för att sammanföringen skall fungera och din uppgift blir att finna en lösning på detta problem. Lektionsmaterialet kan hjälpa dig förstå problemets innebörd, men för att finna en lösning behöver du göra vidare efterforskningar på Internet / kurslitteratur och diskutera gärna problemet tillsammans med dina medstudenter.

```
project_root/
Laboration_1/
  README.md
  README.md
```

När sammanföringen till den lokala källan väl är genomförd framgångsrikt synkroniserar du **remote origin** med det uppdaterade innehållet, som nu alltså skall inkludera två separata **README.md**.

Uppgift 3

I denna uppgift skall du arbeta med förgreningar och skriftligen redovisa tillvägagångssätten för de föregående två uppgifterna.

1. I ditt lokala repo skapar du en ny **branch** från **master** och namnger denna **solution**.
2. Gör så att den nya lokala förgreningen (**solution**) blir aktiv **branch**.
3. Redigera **Laboration_1/README.md** och lägg till en underrubrik för **uppgift 1** samt en formaterad kodruta som innehåller alla kommandon du skrev för att åstadkomma punkterna för den uppgiften (du skall inte ha med lösenord). Tänk på att använda **Markdown** formattering!
4. I samma **README.md** lägger du till en underrubrik för **uppgift 2** tillsammans med en text där du i egna ord beskriver de problem som uppstod under arbetet med uppgiften och på vilka sätt dessa löstes. Nämn gärna även alternativa lösningar och / eller eventuella tillvägagångssätt för att undvika dessa problem.
5. Spara ändringarna lokalt med lämpligt **commit message** och synkronisera så att både **master** och **solution** finns på Bitbucket (**master** skall inte innehålla de nya förändringarna).

Redovisning

Utför en formell inlämning i avsedd inlämningslåda i kursens Moodle-instans. Projektillstånd och nödvändiga filer hämtas direkt från ditt dedikerade studentrepo, så se till att detta repo är uppdaterat innan inlämningen.

Arbetet skall bedrivas enskilt med individuell redovisning, och pararbete är inte tillåtet!