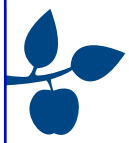

SYDDANSK UNIVERSITET
UNIVERSITY OF SOUTHERN DENMARK

SVN in S_eV_eN Minutes

Notesæt til DM526 af
Arun Vadiveal
arvad04@imada.sdu.dk

IMADA
SDU, Odense

7. november 2007

Indhold

1. Opret din gruppe

Åbn en terminal:

Oprettelse af grupper foregår på maskinen bach, så log først ind på bach.

```
(<imadalogin>@<maskinenavn>) ~> ssh bach
```

Opret en gruppe, erstat <gruppenavn> med dit eget valg af navn til gruppen.

```
(<imadalogin>@bach) ~> addGroup <gruppenavn>
```

Du er ejer af gruppen og kan tilføje medlemmer til gruppen. For at tilføje medlemmer skal man bruge deres brugernavn/login på IMADAs maskiner.

```
(<imadalogin>@bach) ~> addToGroup <gruppenavn> <brugernavn>
```

Log ud fra bach.

```
(<imadalogin>@bach) ~> exit
```

Resultatet bør ligne noget i denne stil:

```
(arvad04@gogol) ~> ssh bach
arvad04@bach's password:
Last login: Tue Nov  6 14:14:07 2007 from gogol.imada.sdu
Sun Microsystems Inc.   SunOS 5.8       Generic February 2000
(arvad04@bach) ~> addGroup testgruppe
Please wait...
the group "testgruppe" has been created, and is owned by you.
Please remeber to delete it when you are finished with it!
(arvad04@bach) ~> addToGroup testgruppe tkn
Please wait...
The user "tkn" has been added to the group "testgruppe"
it now contains the following persons: arvad04,tkn
(arvad04@bach) ~> exit
logout
Connection to bach closed.
```

Sletning af gruppe og udelukkelse af et gruppemedlem foregår analogt til hhv. 'addGroup' og 'addToGroup' med 'removeGroup' og 'removeFromGroup'. Du kan se en liste over eksisterende grupper med kommandoen 'groups'. Husk at det skal køres på bach og husk igen at logge ud fra bach.

1.1. Bemærk:

Som standard er man aktiv med gruppen 'student'. Du skifter til den gruppe du har oprettet vha:

```
(<imadalogin>@<maskinenavn>) ~> newgrp <gruppenavn>
```

Som standard er ens umask (user file creating mode mask) 066, hvilket angiver at filer/mapper oprettes med rettighederne: rwx--x--x, hvor det første er ejerens rettigheder (rwx: read, write, execute), det næste er gruppens rettigheder (--x, dvs. kun execute) og det sidste er alle andres rettigheder (--x, også kun execute). Ønsker man at oprette filer og mapper som ens gruppe kan tilgå, skal man angive sin umask til 022, dvs. én selv og gruppen får fuld adgang til alle filer/mapper som oprettes.

```
(<imadalogin>@<maskinenavn>) ~> umask 007
```

Nu kan der oprettes biblioteker og filer på sædvanlig vis af gruppemedlemmer, som andre medlemmer af gruppen kan tilgå. Husk at 'newgrp' og 'umask' kun gælder for den aktuelle session. Startes en ny terminalvindue, skal de køres igen.

Man kan også ændre rettighederne på allerede oprettede filer og mapper.

Skift gruppen på en allerede eksisterende mappe:

```
(<imadalogin>@<maskinenavn>) ~> chgrp <gruppenavn> -R <sti til mappe>
```

Skift gruppen på en allerede eksisterende fil:

```
(<imadalogin>@<maskinenavn>) ~> chgrp <gruppenavn> <sti_til_fil>
```

Giv rettigheder til gruppen og fjern rettigheder for andre for en mappe:

```
(<imadalogin>@<maskinenavn>) ~> chmod g+rwxs,o-rwx -R <sti til mappe>
```

Giv rettigheder til gruppen og fjern rettigheder for andre for en enkelt fil:

```
(<imadalogin>@<maskinenavn>) ~> chmod g+rw,o-rw <sti til fil>
```

2. Opret dit repository

Skift til dit hjemmebibliotek. Hvis du har fulgt første del af guiden er du der allerede.

```
(<imadalogin>@<maskinenavn>) ~> cd ~/
```

Skift til den gruppe som du har oprettet.

```
(<imadalogin>@<maskinenavn>) ~> newgrp <gruppenavn>
```

Sæt din umask, så både dig selv og gruppen har adgang til de filer/mapper som oprettes.

```
(<imadalogin>@<maskinenavn>) ~> umask 007
```

Opret en mappe til dit SVN-repository.

```
(<imadalogin>@<maskinenavn>) ~> mkdir SVN
```

Opret dit repository. --fs-type fsfs er vigtigt hvis repositoriet ligger på et NFS drev, hvilket er tilfældet på IMADA.

```
(<imadalogin>@<maskinenavn>) ~> svnadmin create ~/SVN --fs-type fsfs
```

Sørg for at dit repository tilhører den gruppe du har oprettet.

```
(<imadalogin>@<maskinenavn>) ~> chgrp <gruppenavn> -R ~/SVN
```

Sørg for at din gruppe får rettigheder til at læse, skrive og eksekvere filer og undermapper i SVN, mens alle andre har ingen rettigheder.

```
(<imadalogin>@<maskinenavn>) ~> chmod g+rwxs,o-rwx -R ~/SVN
```

3. Importer dit projekt

Med projekt menes en mappe indeholdende filer og evt. undermapper. Dette kan eksempelvis være en mappe indeholdende de relevante filer til et programmeringsprojekt. Start med at organisere dit projekt på følgende måde:

1) Opret og navngiv en mappe med et sigende navn for projektet.

2) Opret tre undermapper: "branches", "tags"og "trunk".

3) Placer alle projektrelevante filer i mappen "trunk".

Undermapperne "branches", "tags"og "trunk"kræves egentlig ikke af subversion, men det er en populær konvention som det tilrådes at følge.

I eksemplet herunder ses organiseringen af et projekt, hvor den eneste fil i projektet er en tekstfil med 3 linjer tekst.

```
(arvad04@gogol) ~> mkdir testprojekt
(arvad04@gogol) ~> cd testprojekt
(arvad04@gogol) ~/testprojekt> mkdir branches
(arvad04@gogol) ~/testprojekt> mkdir tags
(arvad04@gogol) ~/testprojekt> mkdir trunk
(arvad04@gogol) ~/testprojekt> cd trunk
(arvad04@gogol) ~/testprojekt/trunk> gedit projektfil.txt
(arvad04@gogol) ~/testprojekt/trunk> cat projektfil.txt
Linje 1
Linje 2
Linje 3
```

Importer nu projektet til dit SVN-repository. For at være konsistent med eksemplet, er det vist at man befinder sig i 'trunk' mappen. Men dette er irrelevant, så længe angivelsen af stierne til

henholdsvis, hvor filerne skal importeres fra, og hvor filerne skal importeres til i repositoriet er korrekte.

```
(<imadalogin>@<maskinenavn>) ~/<projektmappe>/trunk> svn import ~/<sti_til_projektmappe>
file://localhost/home/<imadalogin>/SVN/
<projektnavn> -m "Import af projektet"
```

```
(arvad04@gogol) ~/testprojekt/trunk> svn import ~/testprojekt file://localhost/h
ome/arvad04/SVN/testprojekt -m "Import af projektfiler"
Adding         /home/arvad04/testprojekt/trunk
Adding         /home/arvad04/testprojekt/trunk/projektfil.txt
Adding         /home/arvad04/testprojekt/branches
Adding         /home/arvad04/testprojekt/tags

Committed revision 1.
```

4. Begynd at anvende versionsstyring

4.1. Checkout

Hent den nyeste version af projektet ned lokalt på din maskine, hvor <checkout_mappenavn> angiver, hvilken mappe du vil hente filerne ned i. Bemærk at man angiver stien helt ind til 'trunk', hvor selve projektfilerne ligger.

```
(<imadalogin>@<maskinenavn>) ~/<projektmappe>/trunk> cd ~/
```

```
(<imadalogin>@<maskinenavn>) ~> svn checkout file://localhost/home
/<imadalogin>/SVN/<projektnavn>/trunk <checkout_mappenavn>
```

```
(arvad04@gogol) ~/testprojekt/trunk> cd ~/
(arvad04@gogol) ~> svn checkout file://localhost/home/arvad04/SVN/testprojekt/tr
unk SVN_testprojekt
A    SVN_testprojekt/projektfil.txt
Checked out revision 1.
```

4.2. Update/commit

Åbn mappen som du har hentet filerne ned i.

```
(<imadalogin>@<maskinenavn>) ~> cd <checkout_mappenavn>
```

Lav de ændringer du ønsker i dine filer, inden du vil commit'e for første gang. Her i eksemplet benyttes en enkelt fil 'projektfil.txt'. Filen åbnes med en tekst-editor og der bliver tilføjet en linje.

```
(<imadalogin>@<maskinenavn>) ~<checkout_mappenavn» gedit projektfil.txt
```

```
(arvad04@gogol) ~> cd SVN_testprojekt
(arvad04@gogol) ~/SVN_testprojekt> cat projektfil.txt
linie 1
linie 2
linie 3
(arvad04@gogol) ~/SVN_testprojekt> gedit projektfil.txt
(arvad04@gogol) ~/SVN_testprojekt> cat projektfil.txt
linie 1
linie 2
linie 3
linie 4
```

Når man er klar til at commit'e, opdateres først den version man har liggende af projektet, så det

afspejler evt. ændringer som er committed af andre siden sidste update. Derefter commit'es filen.

```
(<imadalogin>@<maskinenavn>) ~> svn update
```

```
(<imadalogin>@<maskinenavn>) ~/<checkout_mappenavn» svn commit -m «kommentar_til_ændringerne
```

Det ses i eksemplet, at revisionsnummeret ved en update stadig er 1. Dvs. der har ikke været andre, som har commit'ed noget i mellemtiden.

```
(arvad04@gogol) ~/SVN_testprojekt> svn update
At revision 1.
(arvad04@gogol) ~/SVN_testprojekt> svn commit -m "Tilføjet linje 4"
Sending        projektfil.txt
Transmitting file data .
Committed revision 2.
```

4.3. Konfliktløsning

I de fleste tilfælde får man ved en update, den nyeste version af projektet, hvor ens egne ændringer og evt. andre gruppemedlemmers ændringer er flettet sammen. Men i nogle tilfælde kan det ske, at 2 personer har ændret i samme fil på en måde, så SVN ikke er i stand til afgøre hvordan de 2 forskellige filer skal flettes sammen til en. I sådant et tilfælde får man en konflikt. Dette markeres med et 'C' i terminalen, når man updater. Bemærk også at man ikke kan commit'e så længe der er en uløst konflikt.

En konflikt kræver, at der er mindst 2 parter som redigerer i samme fil. Vi betragter derfor 2 brugere, 'bruger1' og 'bruger2' i det følgende eksempel hvor en konflikt først fremprovokeres og derefter løses. 'bruger1' antages at være den som har oprettet repositoriet, mens 'bruger2' er et gruppemedlem.

bruger2 starter med at lave en checkout til en ønsket mappe og redigerer i 'projektfil.txt'. Der tilføjes en linje 5, mens der ændres i linje 2. Til sidst commit'es ændringerne.

```
(<imadalogin_bruger2>@<maskinenavn>) ~> newgrp testgruppe
```

```
(<imadalogin_bruger2>@<maskinenavn>) ~> umask 007
```

```
(<imadalogin_bruger2>@<maskinenavn>) ~> svn checkout
file://localhost/home/<imadalogin_bruger1>/SVN/<projektnavn>/trunk <checkout_mappenavn>
```

```
(<imadalogin_bruger2>@<maskinenavn>) ~> cd <checkout_mappenavn>
```

```
(<imadalogin_bruger2>@<maskinenavn>) ~<checkout_mappenavn» gedit projektfil.txt
```

```
(<imadalogin_bruger2>@<maskinenavn>) ~<checkout_mappenavn» svn update
```

```
(<imadalogin_bruger2>@<maskinenavn>) ~<checkout_mappenavn» svn commit -m "Ændret
i linje 2, tilføjet linje 5"
```

```
(arvad04@gretchen) ~> newgrp testgruppe
(arvad04@gretchen) ~> umask 007
(arvad04@gretchen) ~>  svn checkout file://localhost/home/arvad04/SVN/
testprojekt/trunk SVN_testprojekt_bruger2
A    SVN_testprojekt_bruger2/projektfil.txt
Checked out revision 2.
(arvad04@gretchen) ~> cd SVN_testprojekt_bruger2
(arvad04@gretchen) ~/SVN_testprojekt_bruger2> cat projektfil.txt
Linje 1
Linje 2
Linje 3
Linje 4
(arvad04@gretchen) ~/SVN_testprojekt_bruger2> gedit projektfil.txt
(arvad04@gretchen) ~/SVN_testprojekt_bruger2> cat projektfil.txt
Linje 1
Linje 2 - Ændret af bruger2
Linje 3
Linje 4
Linje 5 - Tilføjet af bruger2
(arvad04@gretchen) ~/SVN_testprojekt_bruger2> svn update
At revision 2.
(arvad04@gretchen) ~/SVN_testprojekt_bruger2> svn commit -m "Ændret i
linje 2, tilføjet linje 5"
Sending        projektfil.txt
Transmitting file data .
Committed revision 3.
```

bruger1 har i mellemtiden også ændret linje 2, og tilføjet en linje 5. Ved forsøg på at lave update/commit, viser der sig at være en konflikt.

```
(<imadalogin_bruger2>@<maskinenavn>) ~<checkout_mappenavn» gedit projektfil.txt
```

```
(<imadalogin_bruger2>@<maskinenavn>) ~<checkout_mappenavn» svn update
```

```
(arvad04@gogol) ~/SVN_testprojekt> cat projektfil.txt
Linje 1
Linje 2
Linje 3
Linje 4
(arvad04@gogol) ~/SVN_testprojekt> gedit projektfil.txt
(arvad04@gogol) ~/SVN_testprojekt> cat projektfil.txt
Linje 1
Linje 2 - Ændret af bruger1
Linje 3
Linje 4
Linje 5 - Tilføjet af bruger1
(arvad04@gogol) ~/SVN_testprojekt> svn update
C    projektfil.txt
Updated to revision 3.
```

Ved hver konflikt oprettes 3 nye filer, samt originalen ændres så den indeholder oplysninger om konflikten:

<filnavn>.mine
Dette er din version af filen. Dvs. filen er identisk med den fil du havde inden update, hvor dine ændringer er lavet i.

<filnavn>.rOLDREV (hvor OLDREV er forskellig fra det nyeste revisionsnummer i repositoriet)
Dette er din version af filen, uden dine ændringer. Dvs. den fil som var basis for dine ændringer.

<filnavn>.rNEWREV (hvor NEWREV er den nyeste revision som ligger i repositoriet.)
Dette er den version som blev hentet ned ved update, og som har forårsaget konflikten. Dvs. en

version som et gruppemedlem har commit'et.

<filnavn>
Endeligt er der en fil med det originale filnavn, hvori differences mellem din version og den version som blev hentet ned er fremhævet.

Der er 3 muligheder ved en konflikt.

- Vurdere at de ændringer man har lavet ikke er relevante længere.
  - Kopier indholdet af <filnavn>.rNEWREV ind i <filnavn>

- Vurdere at de ændringer det andet gruppemedlem har lavet ikke er relevante længere.
  - Kopier indholdet af <filnavn>.mine ind i <filnavn>

- Vælge at se på hvad forskellen er imellem de to filer og rette filen til så de relevante af begges ændringer beholdes.
  - Rediger <filnavn> og behold de af ændringerne som ønskes, mens resten slettes.

Den sidste mulighed er, hvad man oftest vil anvende. Vi kigger derfor på denne.

```
(<imadalogin_bruger2>@<maskinenavn>) ~> gedit testprojekt.txt
```

Når man har løst konflikten, skal man angive at konflikten er løst.

```
(<imadalogin_bruger2>@<maskinenavn>) ~> svn resolved testprojekt.txt
```

Herefter er det som sædvanligt bare at lave en update/commit.

```
(<imadalogin_bruger2>@<maskinenavn>) ~> svn update
```

```
(<imadalogin_bruger2>@<maskinenavn>) ~> svn commit -m "Ændret i linje 2, tilføjet
linje 5, løst konflikt"
```

```
(arvad04@gogol) ~/SVN_testprojekt> cat projektfil.txt
Linje 1
<<<<<<< .mine
Linje 2 - Ændret af bruger1
=======
Linje 2 - Ændret af bruger2
>>>>>>> .r3
Linje 3
Linje 4
<<<<<<< .mine
Linje 5 - Tilføjet af bruger1
=======
Linje 5 - Tilføjet af bruger2
>>>>>>> .r3
(arvad04@gogol) ~/SVN_testprojekt> gedit projektfil.txt
(arvad04@gogol) ~/SVN_testprojekt> cat projektfil.txt
Linje 1
Linje 2 - Manuelt flettet ændringerne sammen.
Linje 3
Linje 4
Linje 5 - Tilføjet af bruger1 - Vurderet at begge linjer er relevante
Linje 5 - Tilføjet af bruger2 - Vurderet at begge linjer er relevante

(arvad04@gogol) ~/SVN_testprojekt> svn resolved projektfil.txt
Resolved conflicted state of 'projektfil.txt'
(arvad04@gogol) ~/SVN_testprojekt> svn update
At revision 3.
(arvad04@gogol) ~/SVN_testprojekt> svn commit -m "Ændret linje 2, tilføjet linje
 5, løst konflikt"
Sending        projektfil.txt
Transmitting file data .
Committed revision 4.
```

5. Opsummering

5.1. Håndtering af grupper

- ssh bach
  - log ind på bach
- addGroup <gruppenavn>
  - Tilføj/opret en gruppe
- removeGroup <gruppenavn>
  - Fjern/slet en gruppe
- addToGroup <gruppenavn> <brugernavn>
  - Tilføj en bruger til gruppen
- removeFromGroup <gruppenavn> <brugernavn>
  - Fjern en bruger fra gruppen
- exit
  - log af en session

5.2. Rettigheder

- newgrp <gruppenavn>
  - Skift til en gruppe (gælder kun for den pågældende session).

- umask 007
  - Filer og mapper oprettes så gruppen også har adgang (rwx,rwx,---)(gælder kun for den pågældende session).

- umask 077
  - Filer og mapper oprettes så kun ejeren har adgang (rwx, ---, ---)(gælder kun for den pågældende session).

- chgrp <gruppenavn> -R <sti til mappe>
  - Skift gruppen på en mappe og alt indholdet.

- chgrp <gruppenavn> <sti til fil>
  - Skift gruppen på en fil.

- chmod u+rwx,g+rwxs,o-rwx -R <sti til mappe>
  - Giv rettigheder til dig selv og gruppen, samt fjern rettigheder for andre for en mappe og alt indholdet.

- chmod u+rw,g+rw,o-rw <sti til fil>
  - Giv rettigheder til dig selv og gruppen, samt fjern rettigheder for andre til en fil.

5.3. Oprettelse af repository

- svnadmin create <sti_til_mappe> --fs-type fsfs
  - Opret et repository.

5.4. Import af projekt

1) Opret og navngiv en mappe med et sigende navn for projektet.

2) Opret tre undermapper: "branches", "tags"og "trunk".

3) Placer alle projektrelevante filer i mappen "trunk".

- svn import <sti_til_projektmappen> file://localhost/home /<sti_til_repository>/<projektnavn> -m "Import af projektet"
  - Importerer et projekt til repository'et.

5.5. Anvendelse af SVN

- svn checkout file://localhost/home/<sti_til_repository>/<projektnavn>/trunk <checkout_mappenavn>
  - Henter en kopi af den nyeste version af projektet fra repository'et.

- svn checkout file://localhost/home/<sti_til_repository>/<projektnavn>/trunk -r <revisionsnummer> <destinationsmappe>
  - Giver mulighed for at vælge en kopi af en ældre revision.

- svn update
  - Opdater din lokale version så evt. ændringer i repository'et flettes ind.

- svn commit -m «kommentar_til_ændringerne>"
  - Tilføj din version af projektet til repository'et.

- svn resolved <filnavn>
  - Marker at konflikten er løst for den pågældende fil.

- svn add <fil_eller_mappenavn>
  - Næste gang du commit'er vil filen/mappen blive tilføjet repository'et. Hvis det er en mappe, bliver alt indholdet af mappen også tilføjet.
- svn delete fil_eller_mappenavn
  - Næste gang du commit'er vil filen/mappen blive slettet fra repository'et. Bemærk at intet indhold forsvinder. Det er muligt at gå tilbage til enhver revision inden sletningen, hvis man ønsker at fortryde sletningen.
- svn copy <fil_eller_mappenavn> <ny_fil_eller_mappenavn>
  - Laver en kopi af en eksisterende fil eller mappe.
- svn move <fil_eller_mappenavn> <ny_fil_eller_mappenavn>
  - Svarer til en udføre en 'svn copy' efterfulgt af en 'svn delete'
- svn mkdir nymappe
  - Denne kommando svarer til 'mkdir <nymappe>' efterfulgt af 'svn add <nymappe>'
- svn status
  - Viser de lokale ændringer som er foretaget.
- svn diff
  - Viser detaljer i forbindelse med de lokale ændringer. Dvs. hvad der er ændret.
- svn revert <sti_til_fil> eller svn revert -R <sti_til_mappe>
  - Fortryder lokale ændringer. Bemærk at en revert på en konflikt-fil, automatisk også løser konflikten.

Det er vigtigt at benytte de passende svn-kommandoer for at kopiere, flytte, slette osv. for at ændringerne bliver overført til repositoriet.