

Peter Toft <pto@sslug.dk>

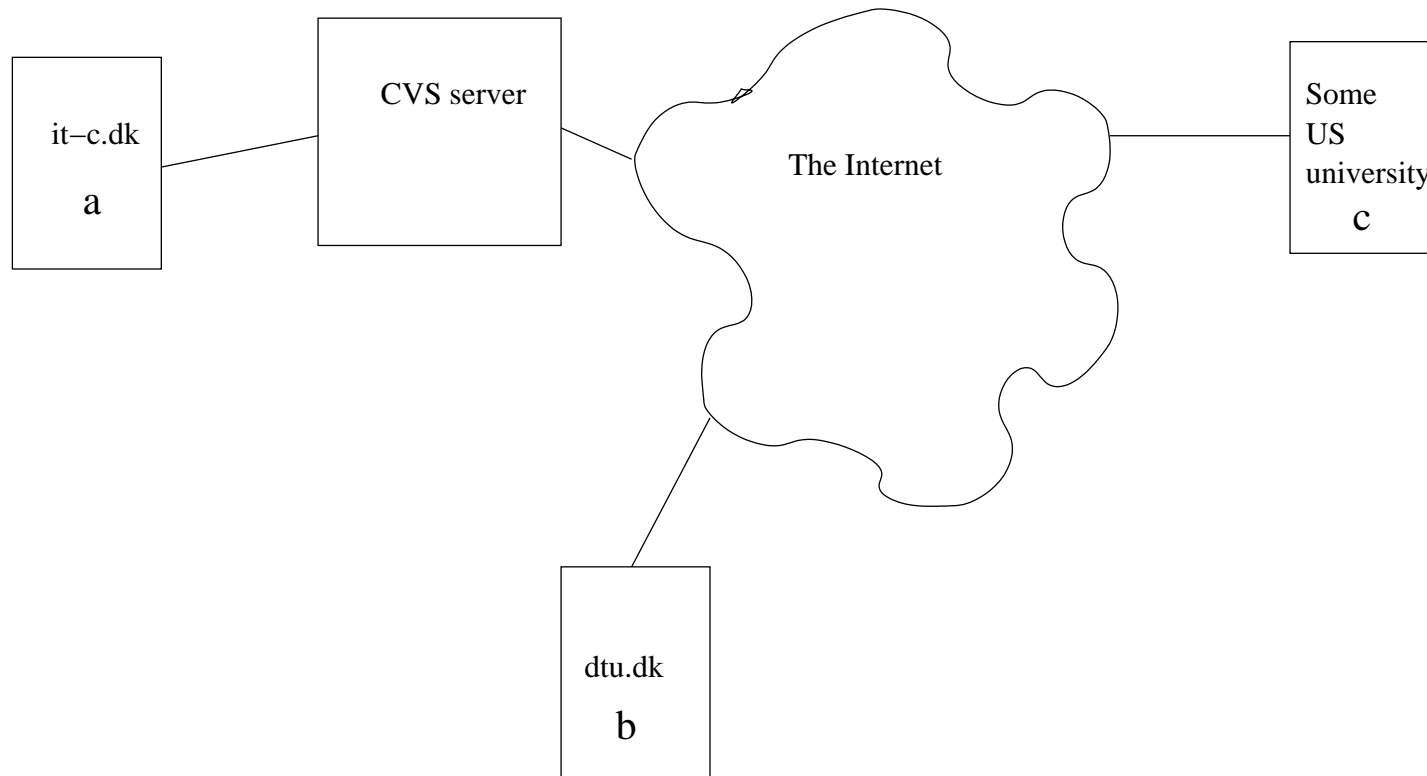
Hanne Munkholm <hanne@geekgirl.dk>

CVS (Concurrent Versions System)

Hvad vil vi tale om?

- Hvorfor versionskontrol?
- Forskellige måder at lave versionskontrol på
- Basal CVS
- CVS over netværk
- Opstart af CVS repository
- Mærkning af filer; CVS tag/update -r
- Følge adgang til filer; CVS watches
- Ulemper ved CVS

Hvorfor versionskontrol?



Forskellige måder at lave versionskontrol på

- Lås af en fysisk fil (RCS)
- Masser af parallelle kopier (CVS)
- Manuel - sende vedhæftede filer frem og tilbage, eller udveksle disketter
- Andre?

Revisionsnumre

- Hver fil har et revisionsnummer der tælles op hvergang filen ændres i CVS repository (ved commit)

```
+-----+      +-----+      +-----+      +-----+      +-----+
! 1.1 !-----! 1.2 !-----! 1.3 !-----! 1.4 !-----! 1.5 !
+-----+      +-----+      +-----+      +-----+      +-----+
```

- Snart vil de forskellige filer i et projekt have forskellige revisionsnumre:

parser.c revision 1.7

compiler.c revision 1.4

preprocessor.c revision 1.2

linker.c revision 1.5

Filer og deres versionsnumre forklaret på tavlen

Basal CVS

`cvs checkout cvs-foredrag`

- Bruges første gang man skal hente et modul (projekt) fra et CVS repository
- Laver et katalog der hedder “cvs-foredrag” i det katalog man står i
- Efter “cvs checkout” går man bare i gang med at redigere i filerne

Basal CVS - cvs commit

```
cvs-foredrag $ cvs commit cvs.tex
```

- Lægger ændringer tilbage i repository
- Hvis man ikke angiver et filnavn laves “commit” på alle ændrede filer
- Hvis andre har ændret i filen i mellemtiden flettes disse sammen
- Der kan opstå konflikter der skal løses manuelt

Basal CVS - cvs update

cvs-foredrag \$ cvs update

- Opdaterer til nyeste version fra repository
- Hvis der er lokale ændringer flettes disse sammen
- \$ cvs update

? cvs.eps

cvs server: Updating .

RCS file: /home/cvs/jedis/cvs-foredrag/cvs.tex,v

retrieving revision 1.7

retrieving revision 1.9

Merging differences between 1.7 and 1.9 into cvs.tex

M cvs.tex

U test.c

Basal CVS - cvs update

- U - Updated
- P - Patched
- M - Locally modified
- C - Conflicts
- A - Added
- R - Removed
- ? - CVS kender ikke filen

Basal CVS - cvs add/remove

```
cvs-foredrag $ cvs add test.c
```

- Lægger filen test.c i kø til at komme med i CVS repository
- Filen kommer først reelt ind i CVS når “cvs commit” køres efterfølgende

```
cvs-foredrag $ cvs remove test.c
```

- Lægger filen test.c i kø til at slettes fra CVS repository
- Filen slettes først reelt fra CVS når “cvs commit” køres
- En fil slettes ikke for alvor fra CVS - de gamle versioner gemmes “på loftet” (Attic)
- Hvis der laves “cvs add” og derefter “cvs remove” på en fil uden at der køres “cvs commit” når den således aldrig ind i CVS repository

Basal CVS - cvs diff

```
cvs diff cvs.tex
```

- Giver forskellen mellem den lokale fil og den tilsvarende fil i repository

- `$ cvs diff cvs.tex`

```
Index: cvs.tex
```

```
=====
```

```
RCS file: /home/cvs/jedis/cvs-foredrag/cvs.tex,v
```

```
retrieving revision 1.6
```

```
diff -r1.6 cvs.tex
```

```
283c283,284
```

```
< \item hvad diff gør
```

```
---
```

```
> \item Giver forskellen mellem den lokale fil og den
```

```
> tilsvarende fil i repository
```

Basal CVS - cvs status

- Fortæller om hvilke filer der er ændret i repository hhv. lokalt.
- `cvs-foredrag $ cvs status`

```
cvs server: Examining .
```

```
=====
```

```
File: cvs.fig          Status: Up-to-date
```

```
Working revision:     1.1
```

```
Repository revision: 1.1   /home/cvs/jedis/ \
```

```
cvs-foredrag/cvs.fig,v
```

```
...
```

```
=====
```

```
File: cvs.tex         Status: Locally Modified
```

```
Working revision:     1.6
```

```
Repository revision: 1.6
```

```
/home/cvs/jedis/cvs-foredrag/cvs.tex,v
```

```
...
```

cvschk - <http://cvs.sslug.dk/cvs2html/>

- Hvordan overskuer man hvad der er ændret?
- “cvs status” er ofte uoverskueligt!

New files

./cvsignore

./cvschk2

Changed files

./cvs2html

cvschk kræver ikke adgang til CVS repository

cvsstat - <http://cvs.sslug.dk/cvs2html/>

- cvsstat er langsom men giver fuldt overblik
- Kan også vise andre personers commit-operationer

```
1 file(s) need(s) patch - update !
```

```
===== My == CVS =====  
README          1.2    1.3
```

```
1 file(s) need(s) merge - update !
```

```
===== My == CVS =====  
cvs.tex         1.1    1.5
```

```
2 file(s) are new to the local CVS
```

```
=====
```

```
Newfilegen
```

Basal CVS - konflikter

- Hvis to har ændret i samme linie, eller meget tæt på hinanden, kan CVS ikke automatisk flette de to versioner sammen - så er der en konflikt (markeret med et "C").

- Eksempel:

I den oprindelige fil: `printf("Hello!");`

Bruger 1 ændrer til: `printf("Hello CVS world!");`

Bruger 2 ændrer samtidig til: `printf("Test program");`

- Er der konflikter, får man ikke lov at køre "cvs commit":

```
$ cvs commit
cvs commit: Examining .
cvs server: Up-to-date check failed for `test.c'
cvs [server aborted]: correct above errors first!
cvs commit: saving log message in /tmp/cvs43JKel
```

Basal CVS - konflikter

- Kør i stedet "cvs update":

```
$ cvs update
cvs server: Updating .
M cvs.tex
RCS file: /home/cvs/jedis/cvs-foredrag/test.c,v
retrieving revision 1.1
retrieving revision 1.2
Merging differences between 1.1 and 1.2 into test.c
rcsmerge: warning: conflicts during merge
cvs server: conflicts found in test.c
C test.c
```

Basal CVS - konflikter

- Når der er konflikt, fletter “cvs update” ændringerne ind med konflikterne markeret:

```
/*This file has the purpose to create conflict*/

#include <stdio.h>

void main(){
<<<<<<< test.c
    printf("Test program");
=====
    printf("Hello CVS world!");
>>>>>>> 1.2
    return;
}
```

Basal CVS - konflikter

- Den fil der er lokalt ændret gemmes som `.#test.c.1.1`
- Løs konflikten filen `test.c` ved at slette den uønskede tekst, og køre "cvs commit":

```
/*This file has the purpose to create a conflict*/  
#include <stdio.h>
```

```
void main(){  
    printf("Test program");  
    return;  
}
```

- `cvs commit test.c`

CVS over netværk - Adgang til CVSROOT

- Vælg CVS server lokalt

```
export CVSROOT=/home/cvs/jedis
```

- Vælg CVS server på en fjern-server

```
export CVSROOT=r2d2.linuxlab.dk:/home/cvs/jedis
```

- Vælg netværks håndtering

```
export CVS_RSH=rsh (for at bruge remote shell rsh over netværket)
```

```
export CVS_RSH=ssh (for at bruge secure shell ssh over netværket)
```

- Kombiner `export CVS_RSH=ssh` med

```
ssh-add ~/.ssh/id_dsa for at få transparent tilgang til CVSROOT over netværket
```

Opstart af CVS repository

Har man ikke et CVS repository (CVSROOT), så dannes denne ved

```
# mkdir /home/cvs
# chmod 755 /home/cvs
# mkdir /home/cvs/jedis
# chmod 2770 jedis/
# groupadd jedis
# vi /etc/group
  jedis:x:513:hm,luke
# chown hm.jedis jedis/

[r2]$ cvs -d /home/cvs/jedis/ init
```

Opstart af CVS repository

- At hente et eksisterende katalog kaldet “cvs-foredrag” ind i CVS:
- `cd cvs-foredrag`
`cvs import cvs-foredrag sslug roskilde_6_januar_2003`
hvor “sslug” kaldes “vendor tag” og “roskilde_6_januar_2003” kaldes “release tag”. Disse navne finder man selv på til sine egne projekter.

- Hvis du ikke har sat din CVSROOT eller hvis du vil importere til et andet CVS repository end din CVSROOT peger på, kan du bruge “cvs -d” til at angive stien:

```
cvs -d r2d2.linuxlab.dk:/home/cvs/jedis import \  
cvs-foredrag sslug roskilde_6_januar_2003
```

Release tag

- Til et vilkårligt tidspunkt kan man mærke samtlige filer med et “tag”
- Man kan altid gå tilbage til et “tag”

```
$ cvs tag rel-1-0
cvs tag: Tagging .
T parser.c
T compiler.c
T preprocessor.c
T linker.c
```

- “rel-1-0” vil mærke filerne ud fra deres nuværende revisions-nummer
- Man kan senere få eksakt få genskabt “rel-1-0” ved at skrive

```
cvs checkout -r rel-1-0 mycompiler
eller
cvs update -d -r rel-1-0
```

CVS watches

- CVS erstatter ikke kommunikation mellem udviklere. Men det kan alligevel være meget rart at kunne se, hvem der retter i en fil på et givent tidspunkt.
- CVS kan sættes til at virke sådan, at en fil er read only indtil man “checker den ud” med en “cvs edit” kommando. Andre kan så se, hvem der har hvilke filer checket ud i øjeblikket.
- Dette sættes op med kommandoen `cvs watch on test.c`. Filen `test.c` vil nu få read only rettigheder, næste gang den opdateres i cvsbrugernes lokale arbejdskataloger. (`cvs watch off test.c` slår “watch” fra igen).
- Derefter bruges `cvs edit test.c` før man retter i filen.
- Når `cvs commit test.c` eller `cvs unedit test.c` køres bliver filen read only igen.

CVS watches - Få email når en fil ændres

- Indsæt `ALL mail %s -s "CVS notification"` i systemfilen `CVSROOT/notify`.
- `cvswatch add test.c` sætter et watch der sender en email hvis andre kører “`cvswatch edit`”, “`cvswatch unedit`” eller “`cvswatch commit`” på `test.c`.
- `cvswatch remove test.c` fjerner dit watch på filen.
- `cvswatch add -a edit test.c` giver kun en mail på “`cvswatch edit`”. Tilsvarende kan man skrive “`unedit`” eller “`commit`”.
- Emailen bliver per default sendt til dit brugernavn på det lokale system (hvor `CVSROOT` ligger). Du kan specificere hver brugers email adresse i `CVSROOT/users` filen:

```
hm:hm@it-c.dk
```

```
luke:luke@geekirl.dk
```

CVS watches - Hvem redigerer en fil?

- `cv`s `editors test.c` viser hvem der pt. har filen `test.c` checket ud med ”`cv`s `edit`”:

```
$ cvs editors test.c
test.c  hm  Mon Nov  4 19:44:33 2002 GMT  hal.palombia.dk
/home/hm/sslug/cvs-foredrag
```

- `cv`s `watchers test.c` viser hvem der pt. holder øje med filen `test.c` (med `cv`s `watch add`):

```
$ cvs watchers test.c
test.c  hm          edit      unedit  commit
        luke       tedit   tunedit tcommit
```

- Eller, når `luke` ikke har `edit` på filen:

```
$ cvs watchers test.c
test.c  hm          edit      unedit  commit
```

Ulemper ved CVS

- Et commit er ikke en atomarisk handling: Når filer i flere directories committes i samme kommando, og nogen samtidig laver en “cvs update”, kan de risikere at få nogle af de opdaterede filer med, men ikke alle.

http://www.cvshome.org/docs/manual/cvs_10.html#IDX182

- Man kan ikke omdøbe en fil på en konsistent måde i CVS.

http://www.cvshome.org/docs/manual/cvs_7.html#SEC70

- Der er andre projekter der prøver at løse disse problemer
- CVS er mest udbredt, og godt nok til de fleste projekter

Referencer:

- Version Management with CVS - The official CVS manual *Per Cederqvist* <http://www.cvshome.org/docs/manual/>
- The CVS home page <http://www.cvshome.org/>
- CVS Pocket Reference *Gregor N. Purdy* ISBN 0-596-00003-0
- 3 artikler af *Jennifer Vesperman*:
Introduction to CVS http://linux.oreillynet.com/pub/a/linux/2002/01/03/cvs_intro.html
CVS Administration <http://linux.oreillynet.com/pub/a/linux/2002/01/17/cvsadmin.html>
Tracking Changes in CVS <http://linux.oreillynet.com/pub/a/linux/2002/06/13/cvs.html>