

ITU – VOOP – F2002

Versionsstyringssystemet

CVS over SSH
Lars Thorup

1

Overblik

- ★ Behov
- ★ Løsninger
- ★ CVS–begreber
- ★ Daglig brug
- ★ Administration
- ★ Opsætning

2

Hvad er behovet?

- ★ Dokumenthåndtering
- ★ Flerbruger
- ★ Versionsstyring
- ★ Distribueret
- ★ Krypteret
- ★ Multiplatform

3

Behov: Dokumenthåndtering

- ★ Mange slags dokumenter
 - ★ kildeтекster
 - ★ databasescripts
 - ★ diagrammer
- ★ Central placering
- ★ Hierarkisk struktur
- ★ Rettigheder
- ★ Backup!

4

Behov: Flerbruger

- ★ Flere kan arbejde med dokumenter
- ★ Samtidig redigere dokumenter
- ★ Automatisk flette samtidige ændringer

5

Behov: Versionsstyring

- ★ Mulighed for at fortryde
- ★ Mulighed for at markere en *baseline*
- ★ Forgøring
- ★ Sammensmeltning af forgøring

6

Behov: Distribueret

- ★ Hver bruger har sin egen maskine
- ★ Arbejde her eller tage med hjem?
- ★ Adgang via internet

7

Behov: Krypteret

- ★ Sikker overførsel over internet
- ★ Undgå aflytning
- ★ Undgå sabotage

8

Behov: Multiplatform

- ★ Nogen bruger Linux
- ★ Andre bruger Windows
- ★ Er der nogen der bruger Mac?

9

Mulige løsninger

- ★ Microsoft Visual SourceSafe
- ★ Perforce
- ★ CVS

10

Løsning: CVS

- ★ Concurrent Versions System
- ★ www.cvshome.org
- ★ Fri software: GPL, open source, gratis
- ★ Server på Unix og Windows
- ★ Klienter på Unix, Windows, Mac

11

CVS-begreber

- ★ repository
- ★ working copy
- ★ checkout
- ★ update
- ★ commit

12



Begreb: repository

- ★ Ligger på serveren
- ★ Indeholder
 - * dokumenter
 - * versionsinformation
 - * låsning
- ★ Alle dokumenter hører til netop et projekt
- ★ Et repository kan indeholde mange projekter

13

Begreb: working copy

- ★ Ligger på klienten
- ★ Hver bruger har sin egen working copy
- ★ Indeholder
 - * dokumenter, evt ændrede
 - * versionsnummer på originaldokumentet

14

Begreb: checkout

- ★ Når man skaber sin working copy
- ★ Checkout fungerer pr projekt

15

Begreb: update

- ★ Når man opdaterer sin working copy
- ★ Hermed modtager man nye ændringer
- ★ Kan foretages på dele af et projekt

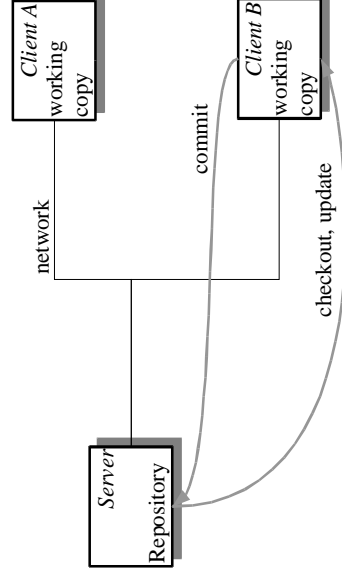
16

Begreb: commit

- ★ Når man opdaterer repository
- ★ Herved afleverer man sine ændringer
- ★ Kan foretages på dele af et projekt

17

CVS-begreber



18

Daglig brug

- ★ Update
- ★ Rediger
- ★ Update igen
- ★ Log
- ★ Fix conflicts
- ★ Diff
- ★ Add
- ★ Commit

19

Kommando: Update

- ★ cvs update
- ★ Bruges til at få det andre har lavet
- ★ Overskriver dine uændrede filer
- ★ Merger ændringer ind i dine ændrede filer
- ★ Hvis merge er umulig, så markeres filen som Conflict

20

Kommando: Update – output

```
* $ cvs update
? cvs-logo.png # ny
U intro.sdd # updated
P udvproj.sdd # updated (patch)
M cvs-ssh.sdd # modified
C allemail.txt # conflict
```

21

Kommando: Log

```
* cvs log <filename>
* Lister alt om filen:
* placering i repository
* nyeste version
* for hver version
  • tidspunkt
  • brugernavn
  • besked
```

22

Kommando: Log – output

```
* $ cvs log main.c
. . .
-----
revision 1.21
date: 2001/08/06 12:54:11; author: ith; state: Exp; lines: +5 -3
update references to tools
. . .
```

23

Conflicts

```
* $ cvs update
C main.c
* $ cat main.c
gencode()
<<<<<<<<
=====
exit(nerr == 0 ? OK : FAIL);
exit(!nerr);
>>>>>>> 1.6
* Skal fixes før man kan commit'te
```

24

Kommando: Diff

- ★ `cv diff <filename>`
- ★ Bruges til at se hvad der er ændret
- ★ Bruges typisk mellem egen og repository version
- ★ Så man ved hvilken kommentar man skal skrive
- ★ Kan bruges mellem to vilkårlig versioner

25

Kommando: Add

- ★ `cv add <filename>`
- ★ Bruges til at tilføje directories og filer
- ★ Er først "rigtig" tilføjet efter commit
- ★ Specielt for binære filer:
`cv add -kb <filename>`

26

Kommando: Commit

- ★ `cv commit -m "...kommentar..."`
- ★ Sender alle dine ændringer til repository
- ★ Medmindre du mangler at update'e

27

Administration

- ★ Slette
- ★ Omdøbe
- ★ Flytte
- ★ Nyt projekt

28

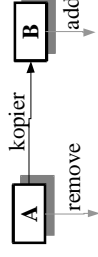
Adm: Slette

- ★ `cvs remove -f <filnavn>`
- ★ Bruges til at fjerne en fil fra repository
- ★ Er først "rigtig" fjernet efter commit

29

Adm: Omdøbe og flytte

- ★ Kopier filen
- ★ `cvs add` den nye fil
- ★ `cvs remove` den gamle fil
- ★ `cvs commit` ændringerne



30

Adm: Nyt projekt

```
★ mkdir project
  cd project
  cvs -d :ext:user$server:/repository import project v r
  cd ..
  rm -rf project
  cvs -d :ext:user$server:/repository co project
```

- ★ Importerer et project i repository
- ★ Checkout giver derefter en working copy

31

Opsætning

- ★ SSH på klienten
- ★ CVS på serveren
- ★ CVS på klienten

32

SSH – Linux klient

- ★ (Installer OpenSSH fra www.openssh.org)
- ★ Opret en nøgle med `ssh-keygen`
- ★ Læg nøglen i `.ssh/authorized_keys` på serveren

33

SSH – Windows klient

- ★ Installer PuTTY
- ★ Opret en nøgle med `puttygen.exe`
- ★ Opret en "saved session" i PuTTY.
- ★ Læg nøglen i `.ssh/authorized_keys` på serveren

34

CVS – server

- ★ Opret directory
- ★ Giv gruppen rettigheder
- ★ `cvsexec`

35

CVS – Linux klient

- ★ `export CVS_RSH=ssh`

36

CVS – Windows klient

- ✦ Installer WinCVS fra www.wincvs.org
- ✦ Få den til at bruge PuTTY