

CVS

Tilman Rassy

Technische Universität Berlin
Fakultät II – Mathematik und Naturwissenschaften
Institut für Mathematik

1 Was ist CVS?

1 Was ist CVS?

- ▶ CVS : Cuncurrent Versions System

1 Was ist CVS?

- ▶ CVS : Concurrent Versions System
- ▶ Frühere Versionen einer Datei können rekonstruiert werden

1 Was ist CVS?

- ▶ CVS : Concurrent Versions System
- ▶ Frühere Versionen einer Datei können rekonstruiert werden
- ▶ Mehrere Entwickler können gleichzeitig an einer Datei arbeiten (Konflikt-Auflösung, Merging)

1 Was ist CVS?

- ▶ CVS : Concurrent Versions System
- ▶ Frühere Versionen einer Datei können rekonstruiert werden
- ▶ Mehrere Entwickler können gleichzeitig an einer Datei arbeiten (Konflikt-Auflösung, Merging)
- ▶ *Repository*: Zentraler Aufbewahrungsort für alle Dateien eines Projekts unter Versionsverwaltung, einschl. alter Versionen

1 Was ist CVS?

- ▶ CVS : Concurrent Versions System
- ▶ Frühere Versionen einer Datei können rekonstruiert werden
- ▶ Mehrere Entwickler können gleichzeitig an einer Datei arbeiten (Konflikt-Auflösung, Merging)
- ▶ *Repository*: Zentraler Aufbewahrungsort für alle Dateien eines Projekts unter Versionsverwaltung, einschl. alter Versionen
- ▶ *Working directory*: Lokale Version der Dateien eines Entwicklers

1 Was ist CVS?

- ▶ CVS : Concurrent Versions System
- ▶ Frühere Versionen einer Datei können rekonstruiert werden
- ▶ Mehrere Entwickler können gleichzeitig an einer Datei arbeiten (Konflikt-Auflösung, Merging)
- ▶ *Repository*: Zentraler Aufbewahrungsort für alle Dateien eines Projekts unter Versionsverwaltung, einschl. alter Versionen
- ▶ *Working directory*: Lokale Version der Dateien eines Entwicklers
- ▶ Repository und Working Directory müssen nicht auf demselben Rechner liegen (Client-Server-Architektur)

2 Benutzung

2 Benutzung

► Aufruf:

```
cvs [-d cvsroot] cmd params
```

cv^sroot = Wurzelverzeichnis der Repositories (s.u.)

cmd = CVS-Commando

params = Parameter für *cmd*

2 Benutzung

- ▶ Aufruf:

`cvs [-d cvsroot] cmd params`

cvsroot = Wurzelverzeichnis der Repositories (s.u.)

cmd = CVS-Commando

params = Parameter für *cmd*

- ▶ Format von *cvsroot*:

2 Benutzung

► Aufruf:

```
cvs [-d cvsroot] cmd params
```

cv^sroot = Wurzelverzeichnis der Repositories (s.u.)

cmd = CVS-Commando

params = Parameter für *cmd*

► Format von *cv^sroot*:

- Repository auf selben Rechner: Absoluter Pfad

2 Benutzung

► Aufruf:

```
cvs [-d cvsroot] cmd params
```

cvsroot = Wurzelverzeichnis der Repositories (s.u.)

cmd = CVS-Commando

params = Parameter für *cmd*

► Format von *cvsroot*:

- Repository auf selben Rechner: Absoluter Pfad
- Repository auf anderem Rechner:

```
:ext:user@host:absoluter_pfad
```

user = Benutzername, *host* = Rechner-Adresse.

3 Umgebungsvariablen

3 Umgebungsvariablen

- ▶ CVSROOT – Default für *cvsroot*

3 Umgebungsvariablen

- ▶ CVSROOT – Default für *cvsroot*
- ▶ CVS_RSH – Muss auf `ssh` gesetzt sein, damit Remote-Zugriff über `ssh` erfolgt

4 Wichtigste Befehle

4 Wichtigste Befehle

- ▶ `cvs checkout name`

Checkt das Repository *name* aus

4 Wichtigste Befehle

- ▶ `cvs checkout name`
Checkt das Repository *name* aus
- ▶ `cvs update [dat1 dat2 ...]`
Aktualisiert Dateien im Working Directory

4 Wichtigste Befehle

- ▶ `cvs checkout name`

Checkt das Repository *name* aus

- ▶ `cvs update [dat1 dat2 ...]`

Aktualisiert Dateien im Working Directory

- ▶ `cvs commit -m 'logmeldung' [dat1 dat2 ...]`

Überträgt Änderungen vom Working Directory ins Repository

4 Wichtigste Befehle

- ▶ `cvs checkout name`
Checkt das Repository *name* aus
- ▶ `cvs update [dat1 dat2 ...]`
Aktualisiert Dateien im Working Directory
- ▶ `cvs commit -m 'logmeldung' [dat1 dat2 ...]`
Überträgt Änderungen vom Working Directory ins Repository
- ▶ `cvs add [dat1 dat2 ...]`
Meldet neue Dateien für Versionskontrolle an

4 Wichtigste Befehle

- ▶ `cvs checkout name`
Checkt das Repository *name* aus
- ▶ `cvs update [dat1 dat2 ...]`
Aktualisiert Dateien im Working Directory
- ▶ `cvs commit -m 'logmeldung' [dat1 dat2 ...]`
Überträgt Änderungen vom Working Directory ins Repository
- ▶ `cvs add [dat1 dat2 ...]`
Meldet neue Dateien für Versionskontrolle an
- ▶ `cvs status dat`
Gibt den Status einer Datei an

4 Wichtigste Befehle

- ▶ `cvs checkout name`
Checkt das Repository *name* aus
- ▶ `cvs update [dat1 dat2 ...]`
Aktualisiert Dateien im Working Directory
- ▶ `cvs commit -m 'logmeldung' [dat1 dat2 ...]`
Überträgt Änderungen vom Working Directory ins Repository
- ▶ `cvs add [dat1 dat2 ...]`
Meldet neue Dateien für Versionskontrolle an
- ▶ `cvs status dat`
Gibt den Status einer Datei an
- ▶ `cvs log dat`
Gibt die Logmeldungen zu einer Datei aus