

Software-Safe

Private Softwareprojekte repräsentieren einen enormen Wert. Ein von überall zugängliches und dennoch geschütztes CVS-Repository auf einem professionell gewarteten Rechner stellt sicher, dass sich mühevoll entwickelte Programme nicht versehentlich in Luft auflösen. Michael Schilli

Inhalt

- 104 Flash-Filme mit Ming**
Mit dieser Bibliothek produziert selbst programmierte Software Flash-Files.
- 108 Coffee-Shop**
Ant ist das Make-Äquivalent für Java-Projekte, kann aber viel mehr.
- 112 Perl-Snapshot**
Ein Perl-Modul stellt Gleichungen mit mehreren Variablen auf und löst sie, ein weiteres zeichnet damit Graphen.
Außerdem: Der **GCC-Perfomancetest** auf Seite 50 mit Hintergründen zu modernen Optimierungstechniken.

Preiswertes Webhosting läuft oft auf Linux-Rechnern mit SSH-Zugang. Oft lässt sich mehr als nur Hosting-Leistung herausholen, zum Beispiel ein zentrales CVS-Repository, um die privaten Softwareprojekte zu verwalten. Die Server-Admins kümmern sich um die lästigen Aufgaben wie Update und Backup, der Kunde loggt sich einfach unter seiner SSH-Kennung auf dem Shared Host ein und legt mit »cvs -d /home/Username/mycvs init« sein neues Repository an.

Falls das Programm »cvs« auf dem Server nicht vorliegt, funktioniert auch ein kopiertes Binary von der daheim benutzten Linux-Distribution. Damit steht ein weltweit zugängliches CVS-System zur Verfügung. Auf dem Client bestimmen zwei in »~/bashrc« gesetzte Variablen die Lage des neuen CVS-Servers:

```
export CVS_RSH="/usr/bin/ssh"
export CVSROOT=":ext:Username@Host.Domain.com:/home/Username/mycvs"
```

Um ein neues Projekt im Repository anzulegen genügt es, auf dem Client in ein bestehendes Projektverzeichnis abzuweichen und es ins CVS zu importieren:

```
client $ cd myproject
client $ cvs import myproject Tagv Tagr
N myproject/Makefile
N myproject/hello.c
N myproject/test/Makefile
N myproject/test/test.c

No conflicts created by this import
```

Die beiden Strings »Tagv« und »Tagr« bezeichnen die Vendor- und Release-Tags, sie interessieren aber meist nicht weiter.

Damit ist das Projekt im Kasten und lässt sich an jedem Ort im Netz auschecken und weiterentwickeln. »cvs co myproject« holt die Sourcen ins aktuelle Verzeichnis auf dem Client – wenn sich der Benutzer per SSH authentifizieren kann.

Aus- und einchecken

Die beiden Kommandos »cvs add« und »cvs commit« checken neue Entwicklungen und Änderungen in das Repository ein und »cvs upd« bringt den lokalen Baum auf den neuesten Stand, falls ein anderer Client den Code in der Zentrale verändert hat.

Das ständige Passworttippen ist freilich lästig. Doch es geht einfacher: Ein SSH-Schlüsselpaar ist auf dem Client mit »ssh-keygen -b 1024 -t rsa« schnell erzeugt. Der in »~/ssh/id_rsa.pub« entstandene Public Key muss auf dem Webhost in die Datei »~/ssh/authorized_keys«. Ab dann lässt der Server den Benutzer per SSH von seinem Client aus ohne Passwort rein und erlaubt CVS-Aktionen. Einfach praktisch! (fjl) ■

Programmierer-Software

<p>Versionsnummern der aktuellen stabilen Releases am 17.06.2004 und Änderungen gegenüber den beiden Vormonaten.</p> <p>Skriptsprachen Guile 1.6.4 Object Rexx 2.3.3 Perl 5.8.4 PHP 4.3.7 Python 2.3.4 Ruby 1.8.1 Tcl 8.4.6</p>	<p>Java Blackdown 1.4.2-rc1 IBM Java2 SDK 1.4.1 SR2 Kaffee 1.0.7 Sun SDK 1.4.2_04</p> <p>Compiler Binutils 2.15 Clisp 2.33.2 Free Pascal 1.0.10 GCC 3.4.0 Gforth 0.6.2 GNU Pascal 2.1 Intel C++ 8.0</p>	<p>Libraries Coin 2.2.1 FLTK 1.1.4 glibc 2.3.2 glut 3.7 GTK+ 2.4.3 libpng 1.2.5 libstd 1.2.7 libxml 2.6.10 Mesa 6.0.1 OpenSSL 0.9.7d QT 3.3.2 wxWindows 2.4.2 zlib 1.2.1</p>	<p>Debugger Code Medic 2.0.1 DDD 3.3.8 GDB 6.1.1 Insight 5.3 Mpatrol 1.4.8 Valgrind 2.0.0</p> <p>Build-Tools Autoconf 2.59 Automake 1.8.5 Libtool 1.5.6 GNU Make 3.80 SWIG 1.3.21</p>	<p>IDEs Anjuta 1.2.2 Code Crusader 4.0.1 C-Forge 4.0 Eclipse 2.1.3 Emacs 21.3 KDevelop 3.0.4 Komodo 2.5.2 Kylix 3 Motor 3.2.4 Source-Navigator 5.1.4 VIDE 2.00 VIM 6.3 XEmacs 21.4.15</p>
--	--	--	--	--