

Theorieteil

1 Verteiltes Programmieren mit Subversion

1.1 Allgemeines

Subversion ist ein *Version Control System*, ein Programm zur Versionsverwaltung von Dateien, hauptsächlich Software-Quellcode. Subversion ist eine Weiterentwicklung des CVS (das noch heute viel eingesetzt wird, z.B. bei SourceForge), welches seinerseits eine Weiterentwicklung des RCS (Revision Control System) ist. Subversion (SVN) erlaubt mehreren Developers gemeinsam an einem Projekt zu arbeiten, ohne sich gegenseitig Code zu löschen.

1.2 Funktionsweise

Subversion vereinfacht die Verwaltung von Quellcode dadurch, dass es alle Dateien eines Software-Projektes an einer zentralen Stelle, einem so genannten *Repository*, speichert. Dabei können jederzeit einzelne Dateien verändert werden, es bleiben jedoch alle früheren Versionen erhalten, einsehbar und wiederherstellbar, auch können die Unterschiede zwischen bestimmten Versionen verglichen werden.

Ein Entwickler holt sich die Daten vom Server (meist über http/https) und arbeitet zu Hause an einer lokalen Kopie. Sobald er seine Arbeit veröffentlichen will, wird er die Daten (die Änderungen) wieder auf den Server laden (*commit*). Der Server wird dann seine eigene Kopie updaten und schauen, dass alle anderen Entwickler an eine aktuelle Version gelangen können. Probleme können sich ergeben, wenn mehrere Mitarbeiter gleichzeitig an der selben stelle in einer Datei Änderungen vornehmen. In diesem Fall muss der Entwickler manuell die verschiedenen Änderungen zusammenfügen (*merge*).

1.3 Anwendungslexikon

- Daten herunterladen: Dieser Befehl lädt die neueste Version von [repository] an die angegebene [destination].

```
svn checkout [repository] [destination]
svn co [repository] [destination]
```

- Dateien hinzufügen: So kannst du Daten oder Verzeichnisse ([file]) hinzufügen. Alle Daten die zum vollständigen Projekt gehören sollten auch hinzugefügt werden.

```
svn add [file]
```

- Dateien löschen: Um Daten im Repository zu löschen, musst du diesen Befehl verwenden.

```
svn rm [file]
```



- Änderungen anzeigen: So kannst du deine lokalen Änderungen einsehen.

```
svn diff
```

- Neue Version herunterladen: Synchronisiert das lokale Repository mit dem Server und lädt alle Änderungen herunter.

```
svn up
```

- Daten hochladen: Dieser Befehl ladet deine lokalen Änderungen auf den Subversion Server. Der Verständlichkeit zuliebe noch ein schöner Kommentar [message].

```
svn checkin -m "message"  
svn ci -m "message"
```

- Weitere Informationen: Falls du mehr über einen gewissen [command] erfahren willst¹.

```
svn help [command]
```

1.4 Subversion in Orxonox

Orxonox verwendet eine spezielle Syntax der Log-Nachrichten beim commiten von Änderungen. Diese Notation erlaubt dem Betrachter einer Log-Nachricht jederzeit die Tätigkeit und den Arbeitsort festzustellen:

```
svn ci -m "orxonox/[folder]: [message]"
```

Wobei [folder] den Ort bezeichnet, an dem du gearbeitet hast, als z.B. in *trunk* oder in *branches/student1*. Die Nachricht [message] sollte möglichst kurz und einfach beschreiben, was verändert worden ist und weswegen (dies ist insbesondere wichtig, wenn man Code von anderen Leuten verändert).

¹Alternative Informationsquelle: <http://svnbook.red-bean.com/>



Praktikum

2 Orxonox installieren, kompillieren und starten

In dieser ersten Lektion wirst du lernen, wie man Orxonox auf den Tardis Rechnern der ETH zum laufen bringt. Dazu wirst du die gebräuchlichen Open Source Tools *subversion* verwenden um die Daten zu *fetchen* und die Automake/Autoconf scripts um das Projekt zu komilieren.

Im zweiten Teil wirst du bekannte Tools ausprobieren, die zur Kommunikation zwischen den Entwicklern dient.

Source Code runterladen: Lade den Source mit Subversion runter, dabei verwenden wir die Syntax aus dem Theorieteil. Zuerst müssen wir aber ein Verzeichnis erstellen und wechseln in dieses:

```
mkdir svn
cd svn
```

Nun können wir den Source auschecken. Dazu wirst du deinen Benutzernamen mit dem dazugehörigen Passwort brauchen, dass wir dir auf einem separaten Blatt gegeben haben (das Blatt mit Passwort bitte nach Eingabe und Auswendiglernen aufessen :D). Orxonox braucht zur Ausführung zwei *Repositories*, eines für den Source (`./orxonox/`) und eins für Daten wie Grafiken, Level und Models (`./data/`)

```
svn co https://svn.orxonox.net/orxonox/trunk orxonox/trunk
svn co https://svn.orxonox.net/data orxonox/data
```

Source Code Compilieren Versuche nun Orxonox zu kompilieren. Dazu verwendest du die standard GNU Compiler Collection (GCC) wie `g++` und `make`. Die Schwierigkeit beim Compilieren liegt meistens darin, dass die Tool unmengen an Flags und command-line argumenten unterstützen. Lass dich nicht verwirren.

Die folgenden Einstellungen sind **ausschliesslich für die Tardis Rechner an der ETH**. Wenn du Orxonox zu Hause installieren willst, dann folgst du am besten den Tutorials auf der Website ²

1. Envirtonment Variablen setzen:

Die Tardis Rechner an der ETH verwenden (im Vergleich zu anderen Linux Distributionen) spezielle Verzeichnisse, die durch sogenannte *environement variables* gesetzt werden müssen. Um das zu tun, musst du folgende Eingaben in der Shell, in welcher du Orxonox comilieren willst tätigen:

```
set ARCH=i686-debian-linux3.1
set PREFIX=/usr/pack/gamelibs-1.0-to
setenv CPPFLAGS -I$PREFIX/include
setenv LDFLAGS "-L$PREFIX/$ARCH/lib -Wl,-rpath -Wl,$PREFIX/$ARCH/lib"
set path= $PREFIX/$ARCH/bin $path
```

²Website/Wiki unter www.orxonox.net



Genau diese Information findest du auch auf den Tardis im Verzeichnis (grosses Merci an den Cheffe Tobias Oetiker):

```
/usr/pack/gamelibs-1.0-to/SEPP/README
und
orxonox/trunk/scripts/tardis.txt
```

2. Konfigurieren, Kompilieren, Linken

```
./configure
make
```

3. Ausführen

```
./src/orxonox
```

3 Kommunikation

Mailinglisten: Trage dich in die folgenden Mailing Listen ein. Orxonox hat drei verschiedene Mailinglisten: **announce** (Ankündigungen von Releases oder Events), **dev** (Entwickler unter sich) und **commit** (die commit messages). Geht dazu auf die folgenden Webpages, um euch anzumelden:

```
https://mail.datacore.ch/mailman/listinfo/orxonox-announce
https://mail.datacore.ch/mailman/listinfo/orxonox-dev
https://mail.datacore.ch/mailman/listinfo/orxonox-commit
```

Wir empfehlen dir sehr für die drei Mailinglisten einen Filter in deinem Mailclient zu definieren (z.B. kannst du für jede Liste ein Verzeichnis erstellen). Dies hilft dir den Überblick zu wahren.

IRC: Der *Internet Relay Chat* ist gut geeignet für kleinere (real-time) Diskussionen. Wir werden viel im *Orxonox IRC* anzutreffen sein, damit wir euch allen bei euren Problemen so schnell wie möglich helfen können. Auf den Tardis sind auch IRC Clients installiert, starte *xchat* und *join* den Channel *orxonox*.

```
xchat irc://irc.datacore.ch
/join #orxonox
```

WIKI: Damit Informationen auch für die Leute zugänglich gemacht werden können, die nicht auf einer Mailingliste eingeschrieben sind oder sich im IRC aufhalten, werden wichtige Informationen direkt ins Wiki geschrieben. Das Wiki ist der wichtigste Datenpool von Orxonox.

```
https://www.orxonox.net/cgi-bin/trac.cgi
```

Login ins Wiki mit dem Benutzernamen und dem Passwort, dass du von uns erhalten hast. Suche den Bereich People unter Misc (unten auf der Seite), dort findest du deinen Namen, in einer komischen grauen Farbe und einem Fragezeichen am Ende (undefinierte Links nehmen im Trac-Wiki diese graue



Farbe an). Wenn du auf deinen Namen klickst, dann kommst du auf eine neue leere Seite. Schreibe hier etwas kleines über dich, damit du die Wiki Syntax kennlernst³.

4 Selbstständiges Arbeiten zu Hause

1. Versuche Orxonox nun mal zu Hause zum laufen zu bringen, damit du auch von dort arbeiten kannst. Wir empfehlen dir Orxonox unter einer Linux Distribution aufzusetzen und zu entwickeln, da dies am einfachsten geht. Orxonox funktioniert aber auch unter Windows und MacOSX (Panther). Falls du bei der Installation auf Probleme stossen solltest, komm doch einfach in den IRC Chat oder schreibe ein Mail an die development Mailingliste.
2. Schau dir unter https://www.orxonox.net/cgi-bin/trac.cgi/wiki/PPS_Topics die Projekt Themen an und überlege dir, woran du gerne arbeiten würdest.

³Wiki Syntax auch einsehbar unter: <https://www.orxonox.net/cgi-bin/trac.cgi/wiki/WikiFormatting>

