Git und Lawa FAQ

July 2, 2010

[Ein Wort im Voraus: Dies soll eine kleine Starthilfe für alle diejenigen sein, die mit an der LAWA arbeiten und die Versionsverwaltung auf dem Apfel nutzen wollen. Ich hab bestimmt einiges vergessen oder nicht klar genug erklärt, deswegen findet sich im Lawa-Verzeichnis auf dem Apfel auch die .tex-Datei, so dass jeder hier noch Änderungen, Ergänzungen etc. einbauen kann. Bei Fragen stehe ich auch gerne zur Verfügung.
Kristina]

Was muss ich alles installieren und einrichten?

Auf dem eigenen Computer sollte Git installiert sein. Geht auf dem Mac über Macports oder Homebrew (http://mxcl.github.com/homebrew/) und vermutlich auch über Fink. Auf http://git-scm.com/download gibts auch Binaries.

Auf dem Apfel sollte, damit das `git-upload-pack` beim ssh-login gefunden wird, die folgende Zeile in die Datei `.bashrc` eingefügt werden (Datei notfalls im Heimatverzeichnis anlegen):

```
export PATH=/opt/local/bin:$PATH
```

Auf dem Mac kann man der Einfachheit halber einen ssh-Schlüssel anlegen: Auf dem eigenen Computer schauen, ob es ein `.ssh/id_rsa.pub` gibt. Wenn nein: mit `ssh-keygen` wird ein Paar öffentlicher und privater Schlüssel erzeugt. Den öffentlichen Teil auf den Apfel kopieren:

```
scp id_rsa.pub loginname@apfel.mathematik.uni-ulm.de:
```

und dort (auf dem Apfel) per

```
cat id_rsa.pub >> .ssh/authorized_keys
```

an die Datei mit den bisherigen Schlüsseln anhängen. Die Datei `id_rsa.pub` kann man dann auf dem Apfel auch einfach löschen. Beim nächsten Einloggen den Schlüsselbund authorisieren, das Passwort zu speichern, danach funktioniert ssh zum Apfel ohne explizite Eingabe des Passworts. Geht bestimmt so oder so ähnlich auch unter Linux.

Nützliche Git-Einstellungen (auf dem eigenen Computer) sind noch die folgenden:

```
git config --global color.ui auto
git config --global user.name 'benutzername'
git config --global user.email 'benutzermail@uni-ulm.de'
```

Die erste Zeile macht beim Arbeiten mit der Shell alles etwas bunter, mit den nächsten beiden legt man den Namen und die email-Adresse fest, die beim Commit mit angegeben werden (ansonsten wird immer der aktuelle Benutzername auf dem verwendeten Computer benutzt).

Was liegt auf dem Apfel?

Im Verzeichnis Groups/num/Documents/Lawa/ liegt ein Git Repository (bare repository) mit dem Haupt-Zweig namens `master` , der die von Stip überwachte und freigegebene aktuelle LAWA-Version enthält, sowie verschiedene Branches, in denen aktiv entwickelt wird.

Ich habe noch keine lokale LAWA-Version. Wie komme ich daran?

Im gewünschten Verzeichnis auf dem eigenen Computer kann man sich mit folgendem Befehl einen Klon des aktuellen LAWA-Verzeichnisses holen:

```
git clone loginname@apfel.mathematik.uni-ulm.de:/Network/Servers/apfel.
mathematik.uni-ulm.de/Volumes/AppleII/Groups/num/Documents/Lawa/LAWA-lite.git
```

Dabei wird im lokalen Verzeichnis ein Ordner `LAWA-lite` angelegt, in dem sich die aktuelle LAWA-Version (entsprechend dem master-Branch auf dem Apfel) befindet.

Ich habe schon ein LAWA. Wie kann ich mit dem Verzeichnis auf dem Apfel arbeiten?

In der LAWA sollte ein .git sein (das ist der Fall, wenn man es irgendwann mal auf dunklen Wegen von Stip bekommen hat). Dann reicht ein

```
git remote add origin loginname@apfel.mathematik.uni-ulm.de:/Network/
Servers/apfel.mathematik.uni-ulm.de/Volumes/AppleII/Groups/num/Documents/
Lawa/LAWA-lite.git
```

um das Apfel-Verzeichnis als Quelle hinzuzufügen.

Ansonsten (ohne .git) muss man leider so vorgehen, als hätte man noch keine LAWA-Version (also am besten in ein anderes Verzeichnis klonen und die lokalen Änderungen dort per Hand reinkopieren).

Wie kann ich jetzt lokal mit Git und LAWA arbeiten?

Git ist dafür geschrieben worden, verteilte Entwicklung zu unterstützen. Das heißt, dass man in der lokalen Kopie mit Git arbeiten kann (und das auch tun sollte!), als gäbe es gar kein zentrales Repository.

Die wichtigsten Befehle sind sicherlich `git add` und `git commit`, mit denen man neue und bearbeitete Dateien in einem neuen Commit speichert. Der Befehl `git status` zeigt Änderungen gegenüber dem letzten Commit an. Mit `git branch branchname` kann man neue Branches anlegen, mit `git checkout branchname` zwischen Branches hin und her wechseln und `git branch` zeigt an, welche Zweige es überhaupt gibt (der aktuelle ist mit einem * gekennzeichnet).

Es gibt außerdem einfache Möglichkeiten, Git mitzuteilen, dass er bestimmte Dateien ignorieren soll. In .git/info/exclude können z.B. Dateiendungen angegeben werden, die generell nicht in der Versionsverwaltung auftauchen sollen (z.B. *.o oder *.dSYM). Außerdem kann man in jedem Ordner eine Datei mit Namen .gitignore anlegen - alle Einträge darin werden im ensprechenden Ordner und allen Unterordnern ignoriert (sinnvoll zum Beispiel fürs Ignorieren von ganzen Verzeichnissen oder ausführbaren Dateien).

Eine sehr gutes Manual ist GitMagic (http://www-cs-students.stanford .edu/ blynn/gitmagic/), auf der Homepage http://git-scm.com/documentation gibt es auch noch viel mehr Doku.

Für den Mac gibt es das schöne Tool GitX (http://gitx.frim.nl/), mit dem man die bisherige Historie ansehen, Dateien zeilengenau vergleichen und auf sehr angenehme Weise Commits machen kann.

Wichtig: Es ist äußerst sinnvoll, viele kleine und thematisch sortierte Commits zu speichern, das macht es sehr viel leichter, bei Fehlern oder nachträglichen Umstrukturierungen die einzelnen Entwicklungs-Beiträge auseinander zu halten. Auch eigene Branches (um bestimmte Ideen zu verfolgen, einfach nur etwas auszuprobieren oder an einem ganz bestimmten Teil der LAWA herumzuspielen) sind extrem hilfreich, da dabei die ursprüngliche Version erstmal nicht verändert wird. Bei Erfolg können diese Branches über `git merge` wieder leicht in den Haupt-Zweig integriert werden.

Wie kann ich lokale Änderungen auf den Apfel bringen?

Der aktuelle Commit (im aktuell ausgewählten Branch) kann per

```
git push origin meinBranch
```

im Branch `meinBranch` auf dem Apfel zur Verfügung gestellt werden (dieser Branch wird dort auch angelegt, falls er noch nicht existiert).

Wichtig: Bitte **nicht** in den Branch `master` pushen! Das ist Stip vorbehalten, damit irgendjemand die Übersicht über LAWA behält. Wer tolle Änderungen hat, kann Stip Bescheid sagen, damit der diese in den `master` integriert.

Wie kann ich meine Version mit dem Apfel synchronisieren?

Der alles-auf-einmal-Befehl fürs Synchronisieren des eigenen Verzeichnisses mit dem auf dem Apfel ist `git pull`. Dabei wird die Apfel-Version geholt und sofort in das lokale Verzeichnis integriert. Ein `pull` ist nichts anderes als `git fetch` und `git merge` hintereinander ausgeführt, siehe unten.

Der Befehl für Vorsichtigere, mit dem alle Änderungen geholt, aber noch nicht sofort eingebaut werden, ist `git fetch`. Die Änderungen können dann mit GitX oder einem der folgenden Befehle angesehen werden:

```
git diff master..origin/master
git diff --stat master..origin/master
```

Ein `git merge origin/master` integriert diese Änderungen dann in die lokale Version. Falls beim Merge Probleme auftreten, wird Git dies melden. Diese müssen dann von Hand behoben werden. Ein einfaches Werkzeug dazu, dass genau die Problemstellen anzeigt und per Mausklick zeilenweise die jeweils gewünschte Version auswählen lässt, kann per `git mergetool` aufgerufen werden. Normalerweise erzeugt git bei einem Merge auch sofort einen neuen Commit. Falls jedoch Konflikte aufgetreten sind, muss nach deren Auflösung `git commit` noch einmal per Hand aufgerufen werden, um die Version ohne Konflikte zu speichern.

Ich hab das Apfel-Verzeichnis geklont und seh nur einen Zweig namens "master". Stand oben nicht noch was von anderen Entwicklungs-Branches?

Wenn man außer mit dem Master noch mit anderen Branches auf dem Apfel arbeiten möchte, muss man einmal einen lokalen Branch mit dem entsprechenden Zweig auf dem Apfel verbinden. Standardmäßig wird beim Clone immer erstmal nur der Master angezeigt, alle weiteren existierenden Branches auf dem Apfel sieht man mit diesem Befehl:

```
git branch -r
```

Dabei gilt: Zweige auf dem Apfel werden immer durch den Zusatz `origin` gekennzeichnet (also: `origin/master` ist die Version auf dem Apfel, `master` die lokale Entsprechung).

Möchte man nun einen eigenen Zweig anlegen und mit dem zugehörigen Branch auf dem Apfel "verknüpfen", so dass Änderungen dort ab da immer in den entsprechenden lokalen Branch geholt werden, so kann man das mit dem folgenden Befehl machen:

```
git branch -t localbranchname origin/apfelbranchname
```

Ein `pull` im Zweig `localbranchname` wird dann die aktuelle Version im Branch `apfelbranchname` herunterladen.