Git Kurs

1 Einleitung

In der Softwareentwicklung arbeiten meistens viele Entwickler zusammen an einem Projekt. Eines der größten freien Projekte ist der Kernel des freien Betriebssystems Linux mit über 1000 Entwicklern aus mehr als 30 Ländern. Alle diese Menschen müssen ihre Arbeit an den gleichen Dateien koordinieren und dabei noch die Übersicht über verschiedene Entwicklungsstände (Versionen) behalten. Zu diesem Zweck hat der Gründer des Linux Projekts, Linus Torvalds, das Versionskontrollsystem Git entwickelt.

Git lässt sich aber nicht nur zur kooperativen Softwareentwicklung nutzen sondern überall wo mehrere Menschen gemeinsam Dateien bearbeiten. Auch kann Git als eine Backuplösung verwendet werden so dass man immer zu einer älteren Version einer Datei zurückspringen kann.

Suchen Sie im Internet nach weiteren Informationen über Git und Linux! Was bedeuted der Name Git? Was ist das Maskottchen von Linux? Wie verbreitet ist Linux?

2 Ein neues Repository

Git verwaltet Dateien in einem Verzeichnis und dessen Unterverzeichnissen. Legen Sie daher bitte ein neues Verzeichnis an (Befehl `mkdir Verzeichnisname`), das danach zu einem sogenannten Repository gemacht werden soll. Öffnen Sie eine Konsole und wechseln Sie in das soeben angelegte Verzeichnis (Befehl `cd Verzeichnisname`). Überprüfen Sie mit dem Befehl `ls -a`[1], dass das Verzeichnis leer ist.

Ein neues Git repository im aktuellen Verzeichnis erzeugt der Befehl `git init`. Danach können Sie mit `ls -a` sehen, dass ein neues Verzeichnis ".git" angelegt wurde, in dem Git seine Hilfsdateien ablegt. Der nächste Befehl, `git status`, bestätigt, dass in dem Repository noch nichts passiert ist:

```
# On branch master
#
# Initial commit
#
nothing to commit (create/copy files and use "git add" to track)
```

3 Dateien hinzufügen

Die von Git verwalteten Versionen werden "Commit" und das Anlegen einer neuen Version "Committen" genannt. Legen Sie mit dem Befehl `touch` gefolgt von einem Dateinamen eine leere Datei an. Mit dem Befehl `git add` gefolgt vom Dateinamen wird diese Datei zum nächsten Commit hinzugefügt.

Erst mit dem Befehl `git commit` wird tatsächlich eine neue Version in Git angelegt. Jeder Commit muss auch eine kurze Erklärung erhalten, was in dieser Version verändert wurde. Daher öffnet der Aufruf von git commit einen Texteditor in dem man diese Erklärung eingeben kann, speichert und den Editor schließt. Alternativ kann man mit der Option `-m` direkt eine Erklärung angeben, z.B.:

```
git commit -m "3. Kapitel hinzugefuegt"
```

Git hat sich nun darüber beschwert, dass Sie sich noch nicht vorgestellt haben. Wir ignorieren diese Beschwerde im Moment.

Committen Sie weitere Dateien in ihr Repository. Überprüfen Sie nach jedem Befehl den Stand und die Historie Ihres Repositories mit den Befehlen `git status` und `git log`[2]!

[1] Mit der Option -a zeigt der Befehl ls auch versteckte Dateien und Verzeichnisse an

[2] Die Anzeige eines längeren Logs wird mit der Taste q verlassen.

4 Dateien bearbeiten

Kopieren Sie eine LaTeX-Datei (Endung ".tex") in das Repository. Machen Sie Änderungen an der Datei und comitten Sie neue Versionen der Datei mit `git add` und `git commit`. Der Befehl `git log -p` (Option `-p` für "patch") zeigt die Änderungen der einzelnen Commits an.

Mit dem Befehl `git rm` können Sie Dateien für den nächsten Commit löschen. Löschen Sie die mit `touch` angelegten leeren Dateien mit `git rm` und committen Sie die neue Version!

5 Graphische Tools

Die Konsole ist für den geübten Nutzer das schnellste und mächtigste Werkzeug. Trotzdem bevorzugen Anfänger oft grafische Werkzeuge. Zum Betrachten und Bearbeiten eines Git Repositories gibt es unter Linux die Programme gitk, qgit, giggle, gitg. Probieren Sie aus, ob diese Programme bei Ihnen installiert sind! Wechseln Sie dazu in der Konsole (mit `cd`) in ein Git Repository und geben Sie den Programmnamen als Kommando ein.

Zwei weitere grafische Tools, git gui und git-cola sollten auch installiert sein. Sie sind eher zur Bearbeitung und zum Committen gedacht.

6 Kontakt mit anderen

Am einfachsten ist es, für die Zusammenarbeit einen von zahlreichen[3] Git Hosting Dienst zu benutzen[4]. Der bekannteste und am einfachsten zu benutzende Dienst ist `github.com`. Dort findet man auch eine umfangreiche Hilfe für die nächsten Schritte:

1. `https://help.github.com/articles/set-up-git`
2. `https://help.github.com/articles/create-a-repo`
3. `https://help.github.com/articles/fork-a-repo`

Welche Daten bekommt Github von seinen Nutzern? Wie einfach kann man eine Sicherheitskopie aller seiner wichtigen Daten von Github auf den eigenen Rechner bekommen? Vergleiche die Situation mit Facebook!

7 Schluß

Der Quelltext für diesen Text liegt ebenfalls auf Github, unter `https://github.com/thkoch2001/git-kurs-fuer-schueler`. Sie können den Text dort "forken", verbessern und mir einen "pull request" schicken.

Achtung! Alle auf Github hochgeladen Daten sind für jeden öffentlichen einsehbar! Auch Dateien, die mit `git rm` gelöscht wurden!

[3] `https://git.wiki.kernel.org/index.php/GitHosting`

[4] Wenn man einen eigenen Server betreibt kann man auch sehr einfach einen eigenen Git Hosting Dienst einrichten.