Übungen Unix-Praktikum Blatt 5

Dr. Dirk Frettlöh (dfrettloeh@techfak.de) 14.11.2022

Aufgabe 1

Als Vorbereitung für Aufgabe 2: Erzeugen Sie ein leeres Verzeichnis namens `projekt`. Schreiben Sie ein Shellskript `neue-datei.sh`, das mit einem Parameter i aufgerufen wird und dann die Datei i.dat erzeugt, deren Inhalt aus drei Zeilen der Form `i i i i i i i` besteht. Der Aufruf

```
./neue-datei.sh 2
```

erzeugt also die Datei `2.dat`, deren Inhalt so aussieht:

```
2 2 2 2 2 2 2
2 2 2 2 2 2 2
2 2 2 2 2 2 2
```

Erzeugen Sie nun damit die drei Dateien `1.dat, 2.dat` und `3.dat`. (Die Lösung darf ruhig etwas unelegant sein, da wir noch keine Schleifen behandelt haben.)

Aufgabe 2

Erledigen Sie folgende Aufgaben. Welche Befehle brauchen Sie jeweils?

1. In eine globale git-Konfigurationsdatei `.gitconfig` Ihren Namen und Ihre Emailadresse eintragen,
2. Im Verzeichnis `projekt` aus Aufgabe 1 ein git-*repository* anlegen,
3. Alle Dateien in `projekt` *stagen* und dem *repository* hinzufügen,
4. Eine im lokalen git-Ordner `projekt` geänderte Datei *stagen* und im *repository* ablegen,
5. Eine Datei im lokalen git-Ordner löschen und die Datei aus dem *repository* wiederherstellen,
6. Alle Dateien im lokalen git-Ordner löschen und aus dem *repository* wiederherstellen,
7. Die Datei `neue-datei.sh` aus dem Repository entfernen und unstagen,
8. Eine Datei `.gitignore` so anlegen, dass die Datei `neue-datei.sh` im lokalen git-Ordner ab jetzt von git ignoriert wird (also z.B. von `git add *` nicht mehr dem *repository* hinzugefügt),
9. Die Datei `.gitignore` so ändern, dass ab jetzt alle Dateien der Form `*.sh` ignoriert werden.

Aufgabe 3

1. Schreiben Sie ein Shellskript namens `wieviele`, das anzeigt, wieviel Unterverzeichnisse und wieviel Dateien das Unterverzeichnis `.git` eines als Argument angegebenen Verzeichnisses (Z.B. `projekt/`) aktuell enthält. Ein Aufruf soll z.B. so aussehen:
`$ wieviele ~/projekt/`
Die Ausgabe dann etwa so:
`Das Verzeichnis ~/projekt/.git enthält 17 Unterverzeichnisse und 7 Dateien.`
2. Wie können Sie geschickt herausfinden, in welcher Datei in in welchem Unterverzeichnis von `projekt/.git` ihre commit-Kommentare gespeichert werden? *Tipp: `grep "^abc"` liefert alle Zeilen, die mit abc anfangen.*

Aufgabe 4

Legen Sie ein `git`-Repository an, und ändern Sie einige der Dateien. Probieren Sie `git status -s` aus. Welche Befehle brauchen Sie für folgende Aufgaben:

1. Die Datei `2.dat` so manipulieren, dass sie von `git status -s` angezeigt wird als
   `M 2.dat`?
   (Also Leerzeichen-M-Leerzeichen-2.dat).

2. Die Datei `2.dat` so manipulieren, dass sie von `git status -s` angezeigt wird als
   `M  2.dat`?
   (Also M-Leerzeichen-Leerzeichen-2.dat).

3. Die Datei `2.dat` so manipulieren, dass sie von `git status -s` angezeigt wird als
   `MM 2.dat`?

4. Die Datei `1.dat` so manipulieren, dass sie von `git status -s` angezeigt wird als
   `D  1.dat`?

5. Kniffflig: die Datei `3.dat` so manipulieren, dass sie von `git status -s` angezeigt wird als
   `?? 3.dat`?

Aufgabe 5

Insbesondere für das Modul *Grundlagen des Software Engineering* sowie im *Software-Gruppenprojekt* ist ein routinierter Umgang mit *git* wichtig. Deshalb ist es sehr hilfreich, regelmäßig mit *git* zu arbeiten. Daher sollen Sie sowohl für das Linux-Praktikum als auch für die A&D Abgaben jeweils ein git-Repository anlegen und führen. Hierzu sollen Sie:

1. Die Repositories initialisieren und jeweils eine `.gitignore` anlegen. Überlegen Sie zunächst gemeinsam, welche Dateien aus dem Repository ausgeschlossen werden sollten.

   Für verschiedene Betriebssysteme sowie Programmiersprachen existieren bereits Vorlagen. Diese können zur Erzeugung der `.gitignore` kombiniert werden:

   `https://www.toptal.com/developers/gitignore`

2. Die bisherigen Abgaben schrittweise dem Repository hinzufügen.

3. Das Repository für den Rest des Semesters nutzen.

Um im Laufe des zweiten Semesters die Routine nicht zu verlieren, wird dringend empfohlen, auch für das Modul *Objektorientiertes Programmieren* ein Repository zu führen.

Downloads (Folien, Übungsblätter)

`https://www.math.uni-bielefeld.de/~frettloe/teach/unix22.html`

Hinweise zu den Übungen

Die Übungen dienen dem Erlernen von Linux. Es gibt keine Abgabepflicht, es gibt überhaupt keine Abgaben. In der **A&D-Klausur** am Ende des Semesters werden allerdings Linux-Fragen vorkommen. Außerdem brauchen Sie im Verlaufe des Studiums solide Linuxkenntnisse. Daher ist es sinnvoll, dass Sie die Übungen entweder selbständig lösen, oder aber eines der Tutorien besuchen und die Übungen dort bearbeiten. Darüber hinaus können Sie in den Tutorien den Tutoren Fragen stellen zu Übungen und Vorlesung.