

## Aufgabe 1

Vorbereitung zu Aufgabe 2. (Auf einem Techfak-Rechner, oder per Fernzugang auf compute:) Wechseln Sie in das Verzeichnis `/vol/lehre/Linux`. Erzeugen Sie ein neues Verzeichnis `git-XXX` (wobei `XXX` für eine von Ihnen gewählte Zeichenkette steht). Wechseln Sie in das neue Verzeichnis und legen dort mit `git init --bare` ein neues, leeres Repository an. Dieses wird im Folgenden als *globales* Repository bezeichnet.

## Aufgabe 2

Welche Befehle brauchen Sie für folgende Aufgaben:

1. Das globale git-repo aus Aufgabe 1 in Ihr Home-Verzeichnis klonen (Beim ersten Mal gibt es eine Warnung: "leeres Verzeichnis geklont" oder so. Egal, das legt sich, sobald im Verzeichnis was liegt, also ab Punkt 2)
2. Eine neu erstellte Datei `dat.txt` in Ihrem lokalen Verzeichnis `git-XXX` dem globalen Repository in `/vol/lehre/Linux` hinzufügen
3. Die aktuelle Version einer lokal geänderten Datei `dat.txt` in das globale Repository kopieren
4. Eine lokal geänderte Datei `dat.txt` wieder auf den Zustand im globalen Repository zurücksetzen
5. Eine Datei `dat.txt` aus dem lokalen und globalen Repository entfernen

## Aufgabe 3

Erzeugen Sie einen Versionskonflikt: Arbeiten Sie mit jemandem zusammen an demselben (nichtleeren) Repository. Pullen Sie beide, bearbeiten beide dieselbe Datei `xxx.txt`, und pushen Sie diese in etwa gleichzeitig.

Für den push, der *nicht* durchgeführt wurde, pullen Sie und reparieren Sie die Datei `xxx.txt` sinnvoll in ihrem lokalen Verzeichnis. Pushen Sie anschließend wieder.

Probieren Sie verschiedenen Möglichkeiten aus: Einmal ändern Sie nur verschiedene Zeilen (Person 1 Zeile 1, Person 2 Zeile 3), einmal ändern Sie beide dieselbe Stelle (etwa Zeile 1), einmal fügen Sie Zeilen hinzu (z.B. Person 1 ändert Zeile 1 und fügt dahinter eine Zeile hinzu, Person 2 ändert Zeile 4), einmal fügen Sie skrupellos viele Zeilen hinzu (z.B. Person 1 ändert Zeile 1 und fügt dahinter 10 Zeilen hinzu, Person 2 ändert Zeile 4).

## Aufgabe 4

Starten sie ein neues git-Projekt, dass Dateien `eins.dat`, `zwei.dat` und `drei.dat` enthält. Erzeugen Sie branches `testen` und `testen2` und manipulieren Sie sie so, dass die Ausgabe von `git log --oneline --decorate --graph --all` folgendes Bild ergibt:

```
*    d6f79c8 (HEAD -> master) Merge branch testen2 -> master
|\
| * 83a8321 (testen2) zwei.dat weiter optimiert
| * 80ee055 zwei.dat optimiert
* |   5340274 erster merge testen -> master
|\ \
| * | 86a2a66 (testen) eins.dat erweitert
| | /
| * 53864bc eins.dat verbessert
* | e169af4 drei.dat optimiert
* | 60254ac drei.dat verbessert
| /
* cba0022 eins.dat gefixt
* 009357a Neues Projekt
```

## Hausaufgabe

Von zu Hause: Loggen Sie sich ein auf `compute` (siehe <https://www.techfak.net/dienste/remote/shell>). Legen Sie nun wie in Aufgabe 1 ein neues, leeres, globales git-Repo `git-XXX` in ihrem remote-Verzeichnis `/media/remote/USER/` an. Führen Sie dann auch hier die Schritte aus Aufgabe 2 durch. Der `clone`-Befehl muss nun lauten `git clone USER@files.techfak.de:git-XXX`.

Von einem Laptop innerhalb des Uni-WLANs geht es genau so wie von einem Techfak-Rechner (z.B. im GZI): Um von außen drauf zugreifen zu können, legen Sie das globale git-Repo wie oben beschrieben in ihrem remote-Verzeichnis an.

(Von außen gesehen liegt eine Dateien `/media/remote/USER/dat.txt` dann unter `git clone USER@files.techfak.de:dat.txt`)

---

### 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.