

Einführung in Versionsverwaltung mit git (3 Punkte)

Im Folgenden werden Sie auf einem kleinen Test-Depot arbeiten, um sich mit wichtigen Operationen zur Versionsverwaltung mit git vertraut zu machen. Schlagen Sie die notwendigen Befehle unter <http://git-scm.com/book/de> nach. Die als Vorlesungsmaterial bereitgestellten Einführungsfolien zu git sind ebenfalls hilfreich. Notieren Sie zu *jedem* der folgenden Schritte die Befehle, die Sie verwendet haben.

1 Vorbereiten der git-Installation

1. Zuerst müssen Sie noch das fehlende Paket `ca-certificates-mozilla` installieren. Öffnen Sie die Konsole und führen Sie folgenden Befehl aus:

```
sudo zypper install ca-certificates-mozilla
```

Bestätigen Sie ggf. ihre Verantwortung als Administrator, geben Sie das Passwort `prog2` ein und bestätigen Sie das Installieren des Paketes mit `y`.

2. Setzen Sie global Ihren Namen und Ihre E-Mail-Adresse. Verwenden Sie hierfür die *selben* Angaben wie bei der Anmeldung zur Vorlesung. Dies ist wichtig, da jede ihrer Einbuchungen (engl. commits) mit diesen Informationen versehen werden.

```
git config --global user.name "Vorname Nachname"  
git config --global user.email "myemail@sth.uni-saarland.de"
```

3. Die Konfigurationsoptionen `merge.conflictstyle=diff3` und `push.default=upstream` haben sich für das tägliche Arbeiten als nützlich erwiesen. Setzen Sie diese Optionen global.

```
git config --global merge.conflictstyle diff3  
git config --global push.default upstream
```

4. Schlagen Sie die Bedeutung der eben gesetzten Optionen im eingebauten Hilfesystem nach. Verwenden Sie das eingebaute Hilfesystem neben dem oben angeführten Buch, um Detailinformationen zu Befehlen nachzuschlagen.

```
git help config
```

2 Arbeiten mit git

Nun werden Sie auf dem Test-Depot arbeiten. Serverseitig wird bei einem `push` halbstündig ein geheimer Test durchlaufen. Diesen müssen Sie bestehen, um die Projektpunkte zu erlangen. Für dieses Projekt gibt es keine weiteren Tests.

1. Klonen Sie das Depot [https://prog2scm.cdl.uni-saarland.de/git/project0/\\$NAME](https://prog2scm.cdl.uni-saarland.de/git/project0/$NAME), wobei `$NAME` durch Ihren Benutzernamen zu ersetzen ist. Geben Sie an, dass der Klon in ein Verzeichnis namens `project0` abgelegt wird.
2. Der Zweig (engl. branch) `master` enthält bereits eine Datei `hallo.txt`. Fügen Sie an diese eine weitere Zeile mit dem Inhalt `Ciao!` an.
3. Lassen Sie sich die Änderung anzeigen.

4. Merken Sie die Änderung zum Einbuchen vor.
5. Fügen Sie eine weitere Zeile mit dem Inhalt `Test` an die Datei an.
6. Lassen Sie sich diese Änderung anzeigen.
7. Lassen Sie sich die bereits vorgemerkten Änderungen anzeigen.
8. Lassen Sie sich beide Änderungen gemeinsam anzeigen.
9. Lassen Sie sich den Zustand der Arbeitskopie anzeigen.
10. Verwerfen Sie die letzte Änderung, jedoch nicht die zuvor vorgemerkten.
11. Buchen Sie die vorgemerkten Änderung ein.
12. Erzeugen Sie eine weitere Datei `test.txt` mit dem Inhalt `Test`. Stellen Sie diese Datei unter Versionsverwaltung und buchen Sie die Änderung ein.
13. Versehen Sie die ursprünglich geklonte Einbuchung mit der Marke (englisch "tag") `Start`.
14. Legen Sie einen Zweig namens `Welt` an, der ab der Marke `Start` beginnt. Wechseln Sie auf diesen Zweig. Begutachten Sie, ob ihre Arbeitskopie jetzt den erwarteten Stand hat.
15. Ändern Sie die einzige Zeile in der Datei `hallo.txt` zu `Hallo, Welt!`.
16. Buchen Sie diese Änderung ein.
17. Wechseln Sie zurück zum Zweig `master`.
18. Verschmelzen Sie den Zweig `Welt` nun in den Zweig `master`. Dies führt zu einem Konflikt.
19.
 - Lassen Sie sich den Zustand der Arbeitskopie anzeigen.
 - Hiervon gibt es auch eine Option, um eine Kurzversion anzuzeigen. Schlagen Sie nach, wie dies bewerkstelligt wird und wie dessen Ausgabe zu lesen ist.
20.
 - Lösen sie den Konflikt so auf, dass beide Änderungen übernommen werden.
 - Lassen Sie sich die Änderungen anzeigen. Schlagen Sie nach, wie dieses Diff zu lesen ist.
 - Schließen Sie das Verschmelzen ab.
21. Veröffentlichen Sie den neuen Zustand von `master`.
22. Für oft verwendete Befehlskombinationen kann man Kürzel anlegen, um sich Tipparbeit zu ersparen.
 - Z.B. ein Kürzel `co` für den Befehl `checkout`:

```
git config --global alias.co checkout
```
 - Z.B. ein Kürzel `graph` für die Ausgabe des Logbuchs in Graphform mit Ausgabe von Referenznamen und kurzer Statusangabe der Veränderungen:

```
git config --global alias.graph log --decorate --graph --name-status
```

Hinweise

Geben Sie auf Folgendes acht, um den Test zu bestehen:

- Achten Sie auf Groß- und Kleinschreibung.
- Achten Sie darauf, keine überschüssigen Leerzeichen oder Leerzeilen einzufügen.
- Achten Sie darauf, keine überschüssigen Dateien einzubuchen.
- Wir überprüfen lediglich den Endzustand des Depots. Wenn Sie also einen Fehler gemacht haben, können Sie noch beliebig nachbessern.

3 KWallet zum Speichern Ihres Passwortes benutzen (optional)

Jedesmal wenn Sie mit git auf unseren Server zugreifen, müssen Sie sich authentifizieren. Bei den folgenden Programmierprojekten finden Sie es vielleicht lästig, ständig ihren Benutzernamen und Ihr Passwort eingeben zu müssen. Sie können KWallet benutzen, um ihre Anmeldeinformationen zu speichern.

1. Installieren Sie dafür zuerst die Pakete `kwalletmanager` und `ksshaskpass`:

```
sudo zypper install kwalletmanager ksshaskpass
```

2. Loggen Sie sich aus und wieder ein. Das nächste Mal, wenn Sie nach Ihren Anmeldeinformationen gefragt werden, meldet sich KWallet und möchte eingerichtet werden.
3. Wählen Sie "Basic setup (recommended)".
4. Wählen Sie "Yes, I wish to use the KDE wallet to store my personal information".
5. Wählen Sie "Classic, blowfish encrypted file". Wenn Sie sich mit GPG auskennen, können Sie auch gerne die Verschlüsselung über GPG wählen.
6. Legen Sie ein Masterpasswort fest. Mit diesem Passwort werden alle Daten, die Sie in KWallet hinterlegen, verschlüsselt. Sie können das Passwort auch leer lassen. Dann öffnet sich KWallet automatisch, ohne nach einem Masterpasswort zu fragen. Allerdings liegen dann alle Daten, die Sie in KWallet speichern, quasi unverschlüsselt auf Ihrer Festplatte.

Wenn Sie nun mit git auf unseren Server zugreifen, erscheint ein grafischer Dialog.

- Klicken Sie auf "Remember password", wenn Sie Ihren Benutzernamen eingeben.
- Klicken Sie nochmal auf "Remember password", wenn Sie Ihr Passwort eingeben.

Von nun an, zieht KWallet automatisch ihre Anmeldeinformationen aus der virtuellen Brieftasche. Wenn Sie sich vertippt haben sollten, können Sie die Daten korrigieren, indem sie `kwalletmanager` starten und den Eintrag unter `ksshaskpass` → `passwords` entsprechend bearbeiten.

Abgabe Ihrer Lösung

git push bis spätestens 5. Mai 2015, 23:59.