Санкт-Петербургский политехнический университет Петра Великого
Институт Информационных технологий и управления

Система контроля версий Git

Выполнил: Сухинин А.А. гр. 53501/3 _____
Принял: Выглежанина К.Д. _____

2015 г.

1 Цель работы

Изучить систему контроля версий Git, освоить основные приемы работы с ней.

2 Ход работы

Git - распределенная система контроля версий. Это подразумевает, что каждый разработчик хранит на собственной машине полноценную копию репозитория. Пока разработчик работает в рамках собственного репозиторя, все происходит в рамках обычной СКВ. Помимо этого, предусмотрены также команды для работы с удаленным репозиторием. Например, push для мержа изменений локального репозитория на удаленный и pull для обратной операции.

Изучить справку для основных команд

Ниже приведен список самых частоиспользуемых команд:

1. git add

   Добавление файла к репозиторию

2. git clone

   Создание локальной копии репозитория

3. git push

   Обновление удаленного репозитория

4. git pull

   Обновление локального репозитория

5. git rm

   Удаление файла из индекса

6. git diff

   Отображение внесенных в файл изменений

7. git merge

   Слияние двух и более веток разработки в одну.

Получить содержимое репозитория

Для получения содержимого репозитория необходимо сделать его копию:
git clone https://github.com/suhininalex/InfoSecCourse2015.git

Добавить новую папку и первого файла под контроль версий

git add /latex
git add lab1.tex

Зафиксировать изменения в локальном репозитории

Перед тем как зафиксировать изменения удалим ненужные файлы:
git rm myfirst.pdf
...

Фиксация:
git commit -a -m "Lab1"

Внести изменения в файл и посмотреть различия

Для этого добавим в репозиторий файл текущего отчета, зафиксируем изменение, после чего сравним с измененной версией. Изменения можно просматривать между текущим состоянием и коммитом, между любыми двумя коммитами, также можно уточнять изменения какого файла именно нам интересны.

git diff 3ca89 39cd07 lab2.tex

Отменить локальные изменения

git reset - -hard 3ca89

Внести изменения в файл и посмотреть различия

git diff lab2.tex

Зафиксировать изменения в локальном репозитории, зафиксировать изменения в центральном репозитории

git commit -a -m "just for push"
git push origin

Получить изменения из центрального репозитория

git pull

Поэкспериментировать с ветками

Создание новой ветки
git branch test

Переключение на другую ветку
git checkout test

Слияние
git merge test

Удаление ветки после слияния
git branch -d test

Просмотр списка локальных веток
git branch -a

3 Выводы

Git – это распределенная система контроля версий. Это означает, что кроме контроля версий git также предоставляет возможность удобной одновременной работы с репозиторием целой команды участников. Каждый участник хранит локальную копию репозитория, при необходимости, синхронизируя ее с центральным репозиторием. Кроме того, зачастую удобно создавать отдельные ветки для различных целей или, например, версий продуктов разного качества. Известным приемом является создание отдельной ветки, например, для реализации какой-то функциональности и последующая попытка ее слить с основной веткой. Такой подход основной команде спокойно продолжать работать над главными задачами не мешая ей экспериментами или нестабильными версиями.

Однако, следует помнить, что при редкой синхронизации изменения копятся и количество конфликтов растет. Разрешение конфликтов в такой ситуации может занимать значительное время, поэтому также известной рекомендацией является как можно более частая синхронизация с центральным репозиторием.

Кроме того, при одновременной работе нескольких людей над одним файлом возникает множество конфликтов, которые необходимо решать вручную. По возможности следует избегать таких ситуаций.

Таким образом, git - это удобная система контроля версий, которая позволяет небольшой команде разработчиков успешно взаимодествовать в рамках одного репозитория.