

Лабораторная работа № 2

Первоначальная настройка git.

18.02.2023

Российский университет дружбы народов, Москва, Россия

Информация

- Касымов Заур
- студент группы НКАбд-04-22
- Российский университет дружбы народов

Цели и задачи

- Изучить идеологию и применение средств контроля версий.
- Освоить умения по работе с git.

Теоретические сведения

Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен доступ для участников проекта.

В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную ему версию файлов. После внесения изменений, пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент.

Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.

Примеры использования git

- Система контроля версий Git представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды `git` с различными опциями.
- Благодаря тому, что Git является распределённой системой контроля версий, резервную копию локального хранилища можно сделать простым копированием или архивацией.

- Создание основного дерева репозитория: `git init`
- Получение обновлений (изменений) текущего дерева из центрального репозитория: `git pull`
- Отправка всех произведённых изменений локального дерева в центральный репозиторий: `git push`
- Просмотр списка изменённых файлов в текущей директории: `git status`

Просмотр текущих изменений: `git diff`

Стандартные процедуры работы при наличии центрального репозитория

- Работа пользователя со своей веткой начинается с проверки и получения изменений из центрального репозитория (при этом в локальное дерево до начала этой процедуры не должно было вноситься изменений)
- Затем можно вносить изменения в локальном дереве и/или ветке.
- После завершения внесения какого-то изменения в файлы и/или каталоги проекта необходимо разместить их в центральном репозитории. Для этого необходимо проверить, какие файлы изменились к текущему моменту.
- При необходимости удаляем лишние файлы, которые не хотим отправлять в центральный репозиторий.
- Затем полезно просмотреть текст изменений на предмет соответствия правилам ведения чистых коммитов:

- Если какие-либо файлы не должны попасть в коммит, то помечаем только те файлы, изменения которых нужно сохранить. Для этого используем команды добавления и/или удаления с нужными опциями.
- Если нужно сохранить все изменения в текущем каталоге, то используем: `git add .`
- Затем сохраняем изменения, поясняя, что было сделано: `git commit -am "Some commit message"`
- Отправляем изменения в центральный репозиторий.

Базовая настройка git

- Зададим имя и email владельца репозитория:
- Настроим utf-8 в выводе сообщений git.
- Настройте верификацию и подписание коммитов git.
- Зададим имя начальной ветки (будем называть её master)

Последовательность выполнения работы

- Установка программного обеспечения
- Базовая настройка git
- Создайте ключи ssh
- Создайте ключи pgr

Последовательность выполнения работы

- Настройка github
- Добавление PGP ключа в GitHub
- Настройка автоматических подписей коммитов git
- Настройка gh
- Шаблон для рабочего пространства

Вывод



- Изучили идеологию и применение средств контроля версий.
- Освоить умение по работе с git.

```
slide_level: 2  
aspectratio: 169  
section-titles: true  
theme: metropolis
```

- Используется фреймворк `reveal.js`
- Используется тема `beige`

- Тема задаётся в файле `Makefile`

```
REVEALJS_THEME = beige
```