

## Отчет по лабораторной работе ?2

\_ Управление версиями git \_

Олейник Артём Александрович.

# Содержание

1 Цель работы 1

2 Задание 1

3 Теоретическое введение 1

4 Выполнение лабораторной работы 2

5 Выводы 9

6 Ответы на контрольные вопросы 9

Список литературы 11

-

## 1 Цель работы

– Изучить идеологию и применение средств конт

-

## 2 Задание

1. Зарегистрироваться на Github;
2. Создать базовую конфигурацию для работы
3. Создать ключ SSH;
4. Создать ключ PGP;
5. Настроить подписи git;
6. Создать локальный каталог для выполнения;

-

## 3 Теоретическое введение

В этой лабораторной работе мы познакомимся с (Version Control System, VCS) применяются пр  
Обычно основное дерево проекта хранится в лок  
к которому настроен доступ для участников прое  
Привнесении изменений в содержание проекта с  
совмещать изменения, произведённые разным  
производить откат к любой более ранней версии и  
Существуют классические и распределённые  
Сегодня мы будем работать с распределённой V

Darcs) клиенты не просто скачивают снимок всего. В этом случае, если один из серверов, через который клиентский репозиторий может быть скопирован, каждая копия репозитория является полным бэком. Многие РСКВ могут одновременно взаимодействовать, благодаря этому вы можете работать с различными приложениями, применяя различные подходы одновременно. Это позволяет применять сразу несколько подходов, что совершенно невозможно в централизованной системе.

-

## 4 Выполнение лабораторной работы

?1 Создаем учетную запись на Github и заполняем ее.

*Рис. 1: Создание учетной записи на GitHub*

?2 Далее установим программное обеспечение.

*Установка git-flow в Fedora Linux*

Установим gh в Fedora Linux.

*Установка gh в Fedora Linux*

Перейдем к базовой настройке Git: зададим имя, введем сообщения git; настроим верификацию (будем называть ее master), параметр autocrlf,

*: Базовая настройка*

*Создания ключа ssh по алгоритму rsa ключем*

по алгоритму ed25519:

*Создания ключа ssh по алгоритму ed25519*

?4 Создаем ключ gpg. Генерируем ключи из пары 4096; • Выберите срок действия; значение по умолчанию 0 (бесконечно). • Выберите тип информации, которая сохранится в ключах. • Введите email, который вы хотите использовать на GitHub. • Комментарий. Можно оставить пустым. (рис. 7)

*Рис. 7: Создание ключа gpg*

Добавим gpg ключ в GitHub. Выводим список ключей.

Перейдем в настройки GitHub (<https://github.com/settings/keys>) нажмем кнопку New GPG key и вставим полученный ключ.

*Добавление PGP ключа в GitHub*

?5 Настроим автоматические подписки коммитов и применять его при подписке коммитов. (рис. 11)

*Настройка автоматических подписей коммитов*

Настроим gh.

*Настройка gh*

Для начала необходимо авторизоваться. Ответ который задаст утилита. Авторизуемся можн

-

## 5 Выводы

Изучила средства контроля версий и научилась  
научилась подключать репозитории, добавлять

-

## 6 Ответы на контрольные

1. Что такое системы контроля версий (VCS) и Система контроля версий — программное обеспечение для облегчения Система управления версиями позволяет при необходимости возвращаться к более ранней версии кода, сделанной ранее. Система (System, VCS) применяются для:
  - Хранения истории изменений
  - Отслеживания изменений
  - Поиска причины и ответственного за появление изменений
  - Совместной работы группы над одним проектом, не мешая работе других пользователей
2. Объясните следующие понятия VCS и их термины:
  - Репозиторий - хранилище версий - место, где хранятся все документы в системе контроля версий
  - Commit — отслеживание изменений, сохранение состояния репозитория (текущее состояние (обычно на последней))
  - История хранит все изменения в проекте и позволяет вернуться к предыдущим версиям
3. Что представляют собой и чем отличаются децентрализованные и централизованные VCS. Приведите примеры VCS каждого вида.
  - Централизованные VCS (например, Subversion, CVS, Perforce, Team Foundation Server, AccuRev):
    - Одноосновное хранилище версий
    - Каждый пользователь копирует себе необходимые файлы
    - Добавляет свои изменения обратно в централизованное хранилище
    - У каждого пользователя свой вариант (возможны конфликты)
    - Присутствует возможность добавлять и удалять файлы
  - Децентрализованные VCS (например, Git, Mercurial, Darcs):
    - В классических системах контроля версий предполагается наличие единого репозитория
    - В децентрализованных системах каждый пользователь имеет полную копию репозитория
    - Выполнение большинства функций по управлению версиями происходит локально

1. Осистемеконтроляверсий [Электронный  
[Введение-О-системе-контроля-версий](http://scm.com/book/ru/v2/%D0%92%D0%B2%D) ([ht  
scm.com/book/ru/v2/%D0%92%D0%B2%D](http://scm.com/book/ru/v2/%D0%92%D0%B2%D))

[%D0%9E-%D1%81%D0%B8%D1%81%D1%D0%BA%D0%BE%D0%BD%D1%82%D1%D0%B2%D0%B5%D1%80%D1%81%D0%](#)

2. ЕвгенийГ. Системыконтроляверсий [Элек  
<https://glebradchenko.susu.ru/courses/bach>  
(<https://glebradchenko.susu.ru/courses/bach>
3. Системыконтроляверсий [Электронныйре  
<http://uii.mpei.ru/study/courses/sdt/16/lectur>  
(<http://uii.mpei.ru/study/courses/sdt/16/lectu>

В [ ]: