Лабораторная работа №2

Дисциплина: Архитектура компьютера

Баранов Георгий Павлович

Содержание

# 1 Цель работы

Целью работы является изучить идеологию и применение системы контроля версий. Приобрести практические навыки по работе с системой git.

# 2 Задание

На основе методических указаний провести работу с базовыми командами системы контроля версий git, выучить применение команд для разных случаев использования, настроить GitHub.

# 3 Теоретическое введение

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. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещать изменения, произведённые разными участниками проекта, производить откат к любой более ранней версии проекта, если это требуется. 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. Сервер может сохранять не полную версию изменённых файлов, а производить так называемую дельта-компрессию — сохранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных. Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Можно объединить (слить) изменения, сделанные разными участниками (автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменения. В зависимости от настроек блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая таким образом, привилегированный доступ только одному пользователю, работающему с файлом.

# 4 Выполнение лабораторной работы

## 4.1 Техническое обеспечение

Лабораторная работа была выполнена на домашнем компьютере под управлением операционной системы Fedora Workstation 40.

## 4.2 Базовая настройка Git

Для начала я проведу предварительную конфигурацию Git, для этого открываю терминал и ввожу команды на (рис. -fig. 1).

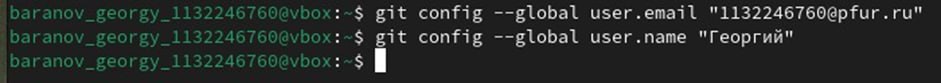


Рис. 1: Предварительная конфигурация Git

Далее настраиваю параметры utf-8, имя начальной ветки, autocrlf и safecrlf. (рис. -fig. 2)

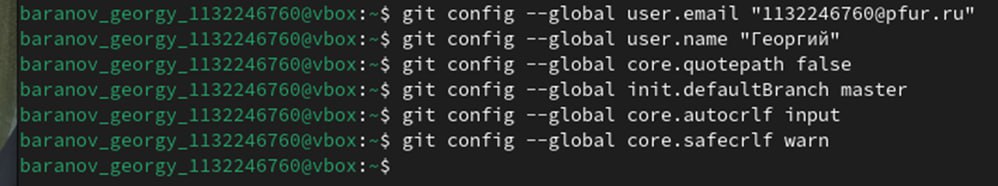


Рис. 2: Настройка параметров Git

Далее создаю пару ssh ключей для интеграции с платформой GitHub. (рис. -fig. 3)

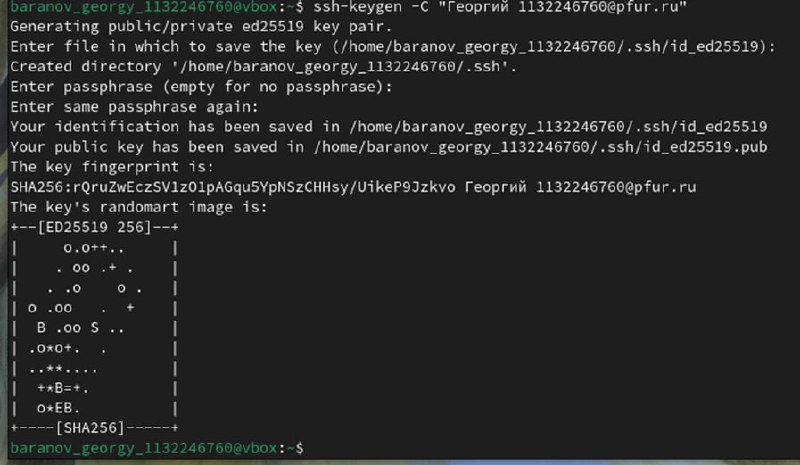


Рис. 3: Создание пары ssh ключей

Далее я перехожу на сайт GitHub, авторизуюсь, перехожу в настройки аккаунта, вставляю публичный ключ в предназначенном для этого поле. (рис. -fig. 4)

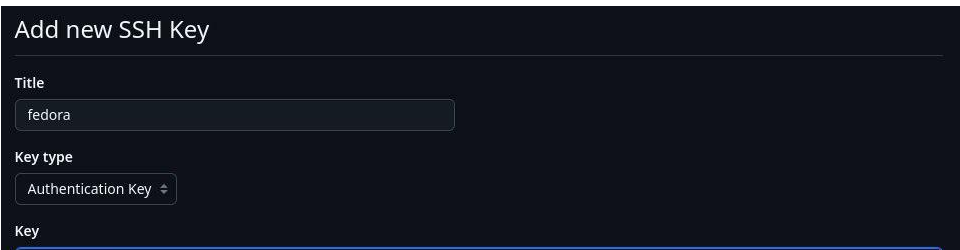


Рис. 4: Установка пары ключей

## 4.3 Создание рабочего пространства и репозитория курса на основе шаблона

Рабочее пространство при выполнении лабораторных работ должно придерживаться определённой структурной иерархии, для этого я создаю директорию на своем рабочем компьютере. (рис. -fig. 5)

Создание рабочей директории

Рис. 5: Создание рабочей директории

## 4.4 Создание репозитория курса на основе шаблона

Создаю репозиторий на основе имеющего шаблона (рис. -fig. 6) через функционал клонирования интерфейса GitHub.

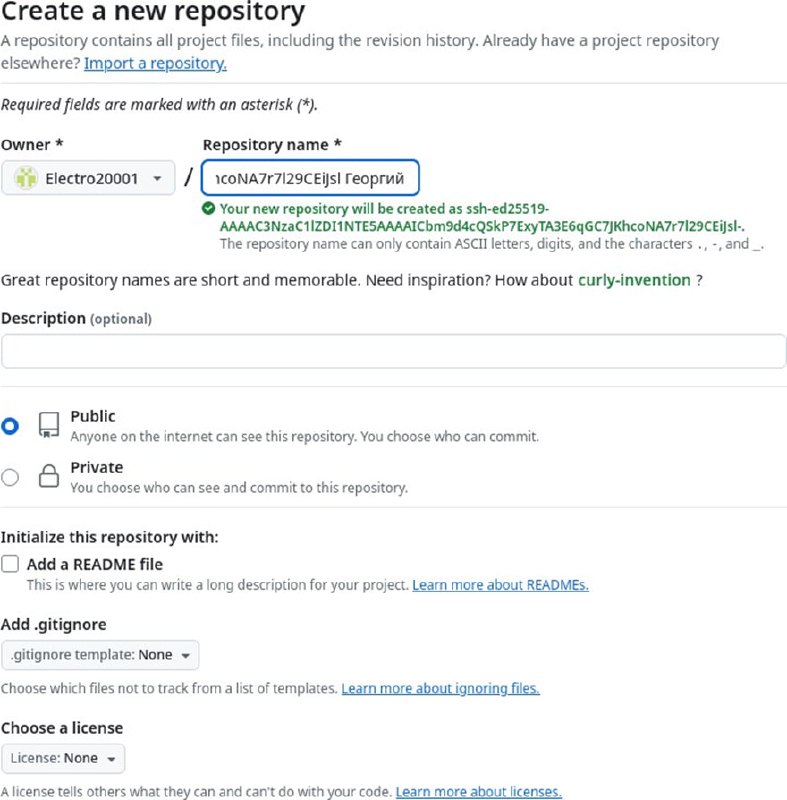


Рис. 6: Создание репозитория на GitHub

Перехожу в каталог и клонирую репозиторий. (рис. -fig. 7)

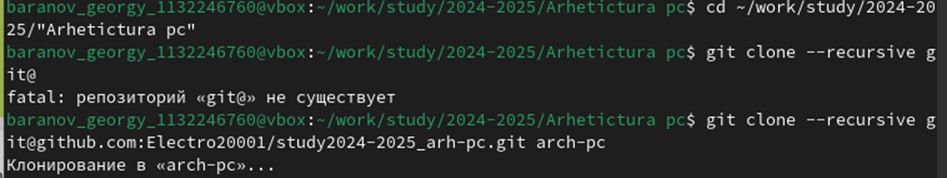


Рис. 7: Копирование репозитория на рабочий компьютер

Создание необходимых каталогов курса (рис. -fig. 8)

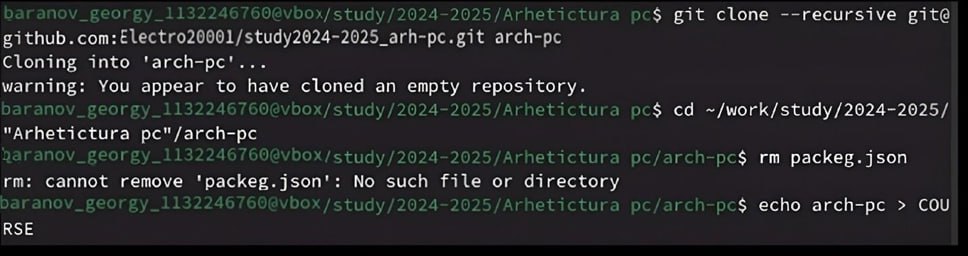


Рис. 8: Настройка каталога

## 4.5 Задания для самостоятельный работы

Через терминал отправляю предыдущий отчет по лабораторной работе на свой удаленный репозиторий в GitHub

Отслеживание файлов, их отправление в репозиторий и проверка выполнения кода (рис. -fig. 9)



Рис. 9: Заключительные действия

# 5 Выводы

Мы познакомились с системой контроля git, выучили команды для работы с ним, создали свой репозиторий на платформе github, где в последующем будут храниться все будущие отчеты по лабораторным работам.

# Список литературы

1. [Лабораторная работа](https://esystem.rudn.ru/pluginfile.php/2089082/mod_resource/content/0/%D0%9B%%20D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%%20BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%%20B0%20%E2%84%962.%20%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%%20BC%D0%B0%20%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%20%BB%D1%8F%20%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D0%B9%20%20Git.pdf)
2. [Курс на ТУИС](https://esystem.rudn.ru/mod/page/view.php?id=1030492)
3. [Методические указания](https://esystem.rudn.ru/mod/resource/view.php?id=1030495)