Отчёт по лабораторной работе №2

Дисциплина: Архитектура компьютера

Бережной Иван Александрович

Содержание

# 1 Цель работы

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

# 2 Задание

1. Настройка github
2. Базовая настройка github
3. Создание SHH ключа
4. Создание рабочего пространства и репозитория курса на основе шаблона
5. Создание репозитория курса на основе шаблона
6. Настройка каталога курса
7. Задание для самостоятельной работы

# 3 Выполнение лабораторной работы

## 3.1 Настройка GitHub

Перейдём на сайт https://github.com/. Справа вверху нажмём кнопку «sign up» и пройдём регистрацию (рис. [1](#fig:001)).

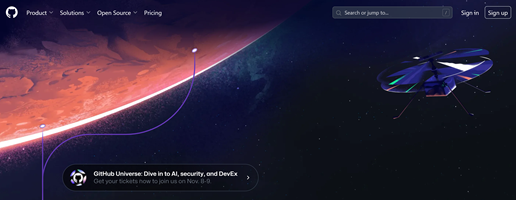


Figure 1: Кнопка регистрации GitHub

## 3.2 Базовая настройка git

Откроем терминал и укажем имя и email владельца репозитория с помощью следующих комманд: (рис. [2](#fig:002)).

Figure 2: Имя и email пользователя

Figure 2: Имя и email пользователя

Далее настроим utf-8 в выводе сообщений git для корректного отображения символов, а также зададим имя начальной ветки (master) и параметры autocrlf и safecrlf (рис. [3](#fig:003)).

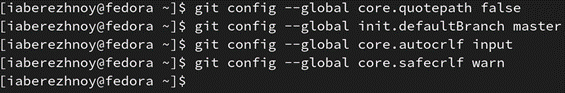


Figure 3: Продолжение настройки git

## 3.3 Создание SSH ключа

Сгенерируем пару ключей (приватный и открытый). Для этого введём команду ssh-keygen -C “Ivan Berezhnoy [darkgxd@vk.com](mailto:darkgxd@vk.com)”. Ключи автоматически сохранятся в каталоге ~/.ssh/ (рис. [4](#fig:004)).

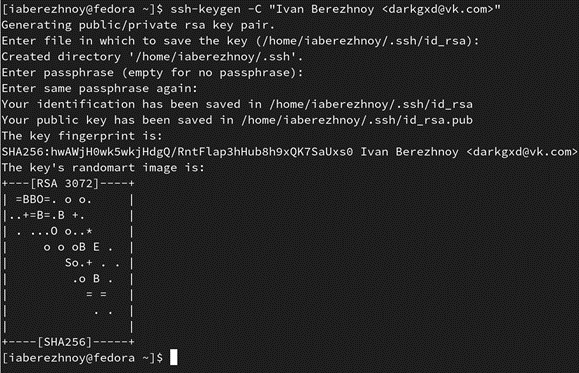


Figure 4: Генерация SHH ключа

Скопируем сгенерированный открытый ключ с помощью команды cat и утилиты xclip (рис. [5](#fig:005)).

Figure 5: Копирование открытого ключа

Figure 5: Копирование открытого ключа

Зайдём на сайт GitHub под своей учётной записью, перейдём в настройки, далее в меню «SSH and GPG keys», нажмём на кнопку «New SSH key». Теперь можем вставить скопированный ключ в соответствующую строку, также укажем имя ключа (рис. [6](#fig:006)).

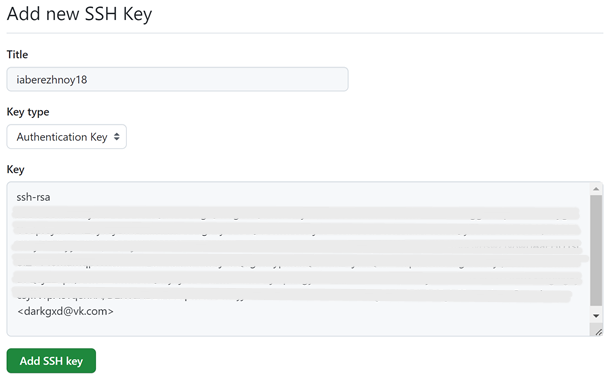


Figure 6: Загрузка SSH ключа

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

Будем придерживаться структуре рабочего пространства. Для этого перейдём в терминал и создадим иерархию каталогов для предмета «Архитектура компьютера» командой mkdir с ключом -p (рис. [7](#fig:007)).

Figure 7: Создание каталога «Архитектура компьютеров»

Figure 7: Создание каталога «Архитектура компьютеров»

## 3.5 Создание репозитория курса на основе шаблона

Снова откроем браузер и перейдём к репозиторию с шаблоном курса по адресу https://github.com/yamadharma/course-directory-student-template и нажмём кнопку«Use this template» (рис. [8](#fig:008)).

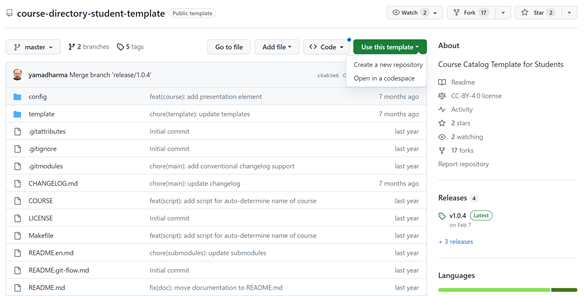


Figure 8: Шаблон курса

Далее создадим новый репозиторий, нажав кнопку «Create repository», предварительно задав ему имя «study\_2023-2024\_arhpc» (рис. [9](#fig:009)).

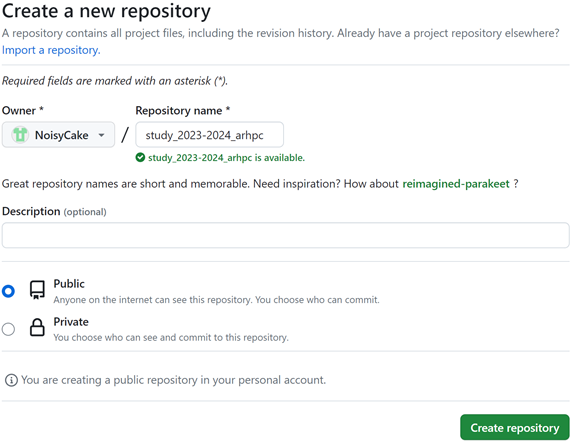


Figure 9: Создание репозитория

Теперь мы можем клонировать созданный репозиторий. Для этого воспользуемся командой git clone –recursive, добавив SSH ссылку на сам репозиторий (рис. [10](#fig:010)).

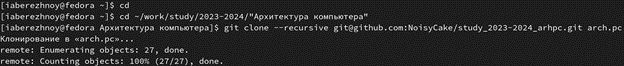


Figure 10: Клонирование репозитория

## 3.6 Настройка каталога курса

Переходим в каталог курса с помощью команды cd и удаляем ненужные файлы командой rm (рис. [11](#fig:011)).

Figure 11: Удаление файлов

Figure 11: Удаление файлов

Теперь создадим необходимые каталоги (рис. [12](#fig:012)) и отправим файлы на сервер (рис. [13](#fig:013)).

Figure 12: Создание каталогов

Figure 12: Создание каталогов

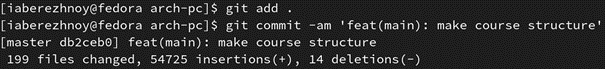


Figure 13: Отправление каталогов на сервер

Проверим, все ли папки и файлы созданы. Сначала откроем файловый менеджер Fedora (рис. [14](#fig:014)), затем проверим GitHub (рис. [15](#fig:015)). Всё получилось.

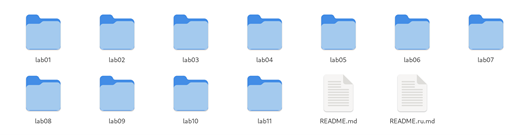


Figure 14: Проверка файлового менеджера

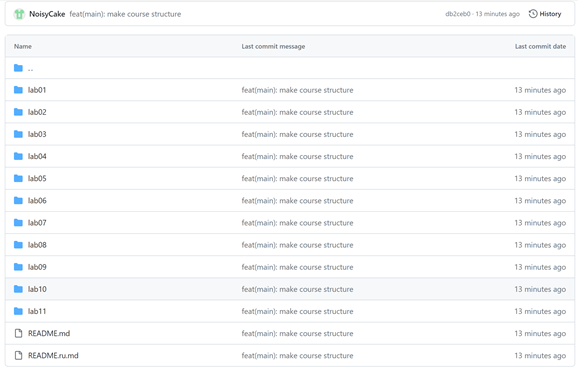


Figure 15: Проверка на GitHub

# 4 Задание для самостоятельной работы

Для начала создадим отчёт по лабораторной работе №2. Откроем LibreOffice Writer внутри Fedora, создадим и сохраним пустой файл с именем «Lab2\_report» в папке Документы (рис. [16](#fig:016)).

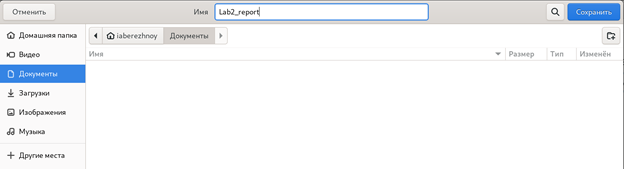


Figure 16: Создание файла для отчёта

После написания отчёта переместим его в нужную нам папку через терминал командой mv (рис. [17](#fig:017)).

Figure 17: Перемещение отчёта

Figure 17: Перемещение отчёта

Вторым шагом скопируем отчёт предыдущей лабораторной работы в соответствующую папку командой cp (рис. [18](#fig:018)).

Figure 18: Копирование старого отчёта

Figure 18: Копирование старого отчёта

Теперь загрузим файлы на GitHub. С помощью утилиты cd перейдём в папку каждого отчёта, а командой git add добавим файлы отчётов в коммит. Затем сохраним внесённые изменения командой git commit с ключом -am и отправим их в центральный репозиторий (рис. [19](#fig:019)).

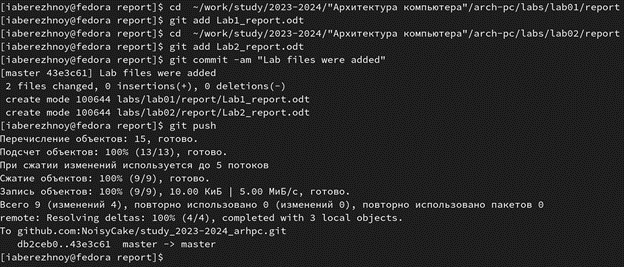


Figure 19: Загрузка отчётов на GitHub

Проверим верность выполнения команд. Откроем репозиторий в GitHub. В папке lab01/report видим наш файл Lab1\_report.pdf (рис. [20](#fig:020)). , а в папке lab01/report файл Lab2\_report.pdf (рис. [21](#fig:021)). Всё верно.

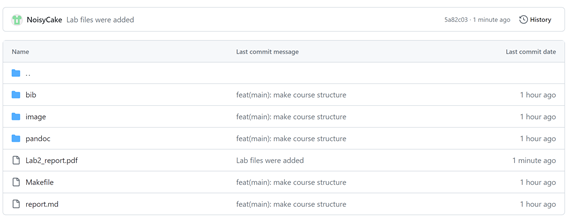


Figure 20: Проверка первого отчёта

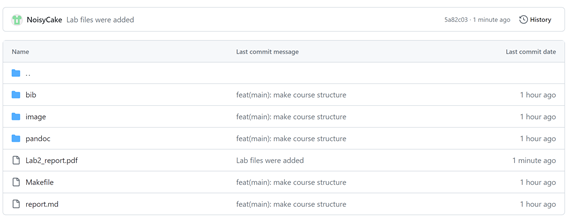


Figure 21: Проверка второго отчёта

# 5 Выводы

В ходе выполнения лабораторной работы мы изучили идеологию и применение средств контроля версий, а также приобрели практические навыки по работе с системой git.

# Список литературы

::: [Архитектура ЭВМ](https://esystem.rudn.ru/mod/resource/view.php?id=1030550)