Отчёт по лабораторной работе 2

Система контроля версий Git

Ермолаев Денис Николаевич НБИбд-01-24

Содержание

# 1 Цель работы

Изучить идеологию и применение систем контроля версий, приобрести практические навыки по работе с системой Git.

# 2 Порядок выполнения лабораторной работы

## 2.1 Настройка Github

Для начала я создал учётную запись на сайте GitHub по адресу https://github.com/. После создания аккаунта я заполнил свои основные данные, такие как имя пользователя и email. Это нужно для того, чтобы Git мог идентифицировать мои действия при работе с репозиторием. Далее я настроил SSH ключ для безопасного доступа к репозиториям. Для этого я сгенерировал пару SSH ключей с помощью команды:

ssh-keygen -C “Мое Имя [мой\_email@mail.com](mailto:мой_email@mail.com)”.

Полученный публичный ключ был добавлен в мой аккаунт GitHub для последующей аутентификации.

## 2.2 Базовая настройка git

После настройки учётной записи на GitHub, я выполнил предварительную конфигурацию git. В терминале я ввёл следующие команды для настройки имени пользователя и электронной почты:

git config –global user.name “Мое Имя”

git config –global user.email “мой\_email@mail.com”

Это нужно для того, чтобы каждый мой commit был подписан моими данными.

Для корректного отображения сообщений в git я также настроил параметр utf-8, введя команду:

git config –global core.quotepath false.

(рис. 1)

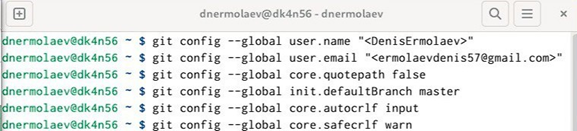


Рис. 1: Базовая настройка git

## 2.3 Создание SSH ключа

Для того чтобы безопасно подключаться к репозиторию, я сгенерировал SSH ключ с помощью команды:

ssh-keygen -C “Мое Имя [мой\_email@mail.com](mailto:мой_email@mail.com)”.

SSH ключ позволяет мне осуществлять безопасную аутентификацию на сервере без необходимости каждый раз вводить пароль. Полученный публичный ключ был загружен на GitHub в разделе “SSH and GPG keys”. (рис. 2)

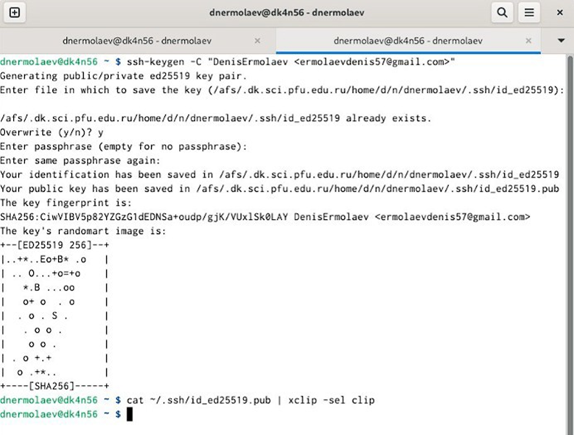


Рис. 2: Создание SSH ключа

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

Я создал рабочее пространство для лабораторной работы, следуя предложенной структуре. Каталоги были созданы с помощью команды:

mkdir -p ~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab02/.

Эта структура позволяет легко управлять файлами лабораторных работ.(рис. 3)

Создание рабочего пространства

Рис. 3: Создание рабочего пространства

## 2.5 Создание репозитория на основе шаблона

Для упрощения работы, я создал репозиторий на основе шаблона курса через веб- интерфейс GitHub. Я использовал готовый шаблон, который предоставил преподаватель, выбрав опцию “Use this template”. Это помогло мне быстро настроить все необходимые файлы и каталоги.(рис. 4)



Рис. 4: Клонирование репозитория

## 2.6 Настройка каталога курса

Перейдя в каталог курса, я удалил ненужные файлы, такие как package.json, и создал необходимые каталоги для курса. Затем я выполнил первичную настройку структуры репозитория с помощью команд:

git add .

git commit -am “feat(main): make course structure”

git push origin master

Эти действия загрузили структуру курса в центральный репозиторий на GitHub. (рис. 5, 6, 7, 8, )

Настройка каталога курса

Рис. 5: Настройка каталога курса

Настройка каталога курса

Рис. 6: Настройка каталога курса



Рис. 7: Настройка каталога курса



Рис. 8: Настройка каталога курса

## 2.7 Задание для самостоятельной работы

Для задания по самостоятельной работе я создал отчёт по выполнению лабораторной работы и разместил его в каталоге lab02. Затем я загрузил все файлы на GitHub с помощью команды:

git push.

# 3 Выводы

В ходе выполнения лабораторной работы я получил практические навыки работы с системой контроля версий Git. Я освоил основные команды, настроил рабочее пространство и репозиторий, а также успешно загрузил результаты на GitHub.