Отчёт по лабораторной работе №3: Система контроля версий Git

дисциплина: Архитектура компьютера

Ибатулина Дарья Эдуардовна

Содержание

# 1 Цель работы

Целью работы является изучение идеологии и применения средств контроля версий, приобретение практических навыков работы с системой git.

# 2 Задание

1. Ознакомиться с теоретическим введением к лабораторной работе №3;
2. Создать учётную запись в системе контроля версий GitHub;
3. Произвести базовую настройку Git;
4. Сгенерировать ключи для дальнейшей идентификации пользователя на сервере;
5. Создать репозиторий и каталоги курса;
6. Сделать выводы по данной работе;
7. Оформить отчёт по данной работе;
8. Загрузить файлы этой и предыдущих лабораторных работ на GitHub, а также ссылку на собственный Github.

# 3 Теоретическое введение

Система контроля версий (также носит название Version Control System, VCS) – система, записывающая изменения в файл или набор файлов в течение всего времени работы над проектом. Применяется для групповой разработки приложений, файлов, программ, позволяет ограничивать и разрешать доступ определённой группе пользователей. Также система позволяет возвращаться к более ранней версии проекта, если это необходимо. Можно даже увидеть, кто именно вносил изменения, и ограничить доступ к журналу изменений. Примеры систем контроля весрий: CVS, Subversion, Git, Bazaar, Mercurial. В данной работе мы будем использовать систему GitHub.

Система контроля версий Git представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды git с различными опциями.

В табл. 1 приведены основные команды для работы с системой контроля версий Git:

Таблица 1: Основные команды для работы с системой контроля версий Git

| Имя команды | Описание команды |
| --- | --- |
| git init | Создание основного дерева репозитория |
| git pull | Получение обновлений (изменений) текущего дерева из центрального репозитория |
| git push | Отправка всех произведённых изменений локального дерева в центральный репозиторий |
| git status | Просмотр списка изменённых файлов в текущей директории |
| git diff | Просмотр текущих изменений |
| git add | Добавить все изменённые и/или созданные файлы и/или каталоги пользователя |
| git rm имена\_файлов | Удаление файлов и/или каталогов |
| git commit -am 'Описание коммита' | Сохранить все добавленные изменения и все изменённые файлы |
| git push origin имя\_ветки | Отправка изменений конкретной ветки в центральный репозиторий |

# 4 Выполнение лабораторной работы

Для начала необходимо зарегистрироваться на сайте Github, создать учётную запись. (рис. 1 2 3)

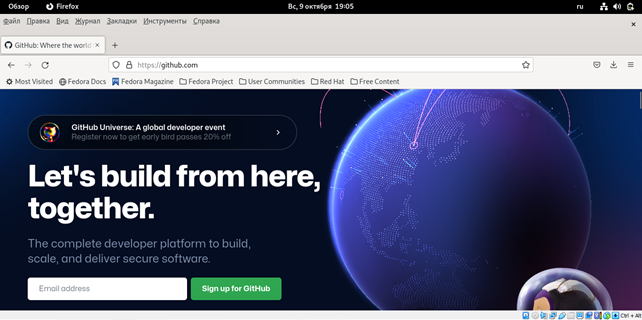


Рис. 1: Открытие стартовой страницы Гитхаб

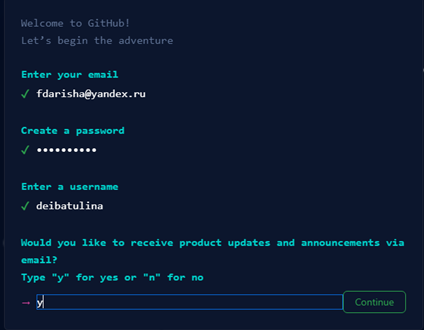


Рис. 2: Регистрация и ввод основынх данных

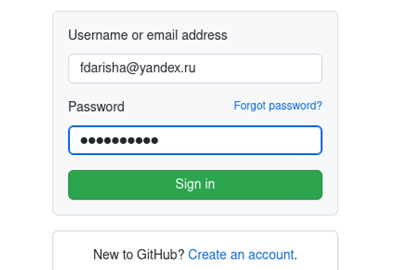


Рис. 3: Вход в свой аккаунт

Сначала сделаем предварительную конфигурацию Git. Для этого введём в терминал следующие команды. (рис. 4, 5, 6, 7, 8, 9, 10)

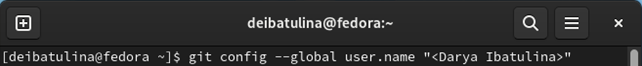


Рис. 4: Ввод имени и фамилии

Рис. 5: Ввод электронного адреса пользователя

Рис. 5: Ввод электронного адреса пользователя

Рис. 6: Задание кодировки utf-8

Рис. 6: Задание кодировки utf-8

Рис. 7: Задание имени начальной ветки

Рис. 7: Задание имени начальной ветки

Рис. 8: Параметр autocrlf

Рис. 8: Параметр autocrlf

Рис. 9: Параметр safecrlf

Рис. 9: Параметр safecrlf

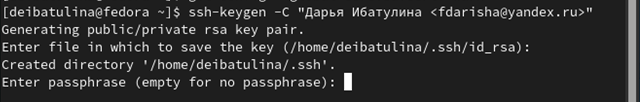


Рис. 10: Генерация SSH-ключа

Мы видим, что система ждёт от пользователя ввода сгенерённого открытого ключа. Далее с помощью команды cat копируем ключ (предварительно установив пакет xclip). (рис. 11)

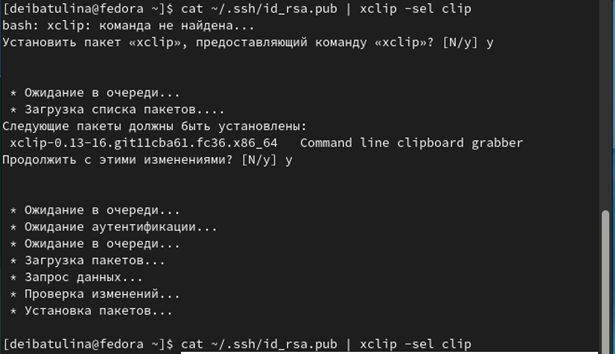


Рис. 11: Установка пакета xclip

Переходим на [сайт](http://github.org/) и вставляем скопированный ключ. (рис. 12)

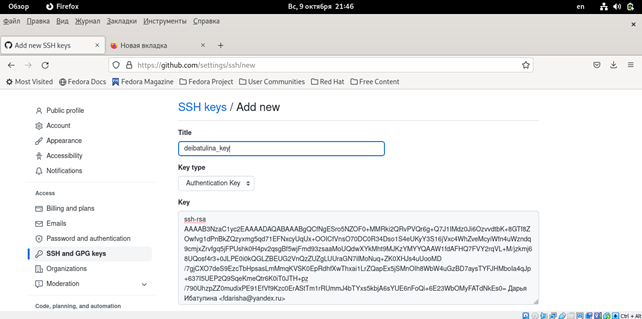


Рис. 12: Ввод скопированного ключа на Гитхаб

Ключ создан. (рис. 13)

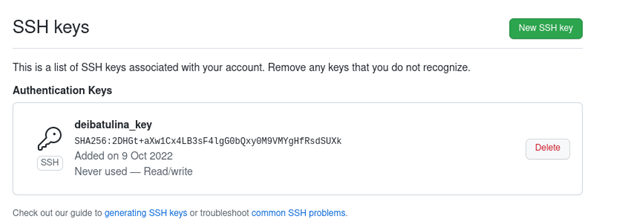


Рис. 13: Окно созданного ключа

Создаем директорию для предмета «Архитектуры компьютеров». (рис. 14)

Рис. 14: Создание директории предмета

Рис. 14: Создание директории предмета

Переходим на [веб-сайт](https://github.com/yamadharma/course-directory-student-template) и нажимаем Use this template. (рис. 15)

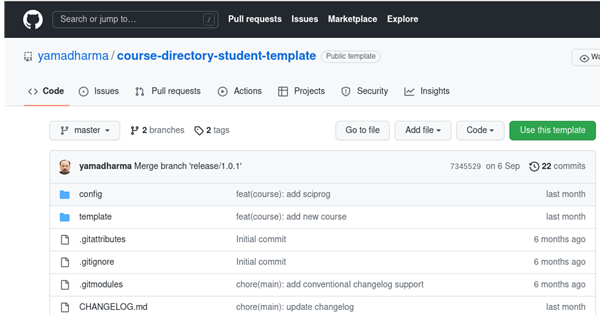


Рис. 15: Выбор шаблона репозитория

Далее вводим имя репозитория, как приведено в указаниях, и нажимаем Create repository from template. (рис. 16)

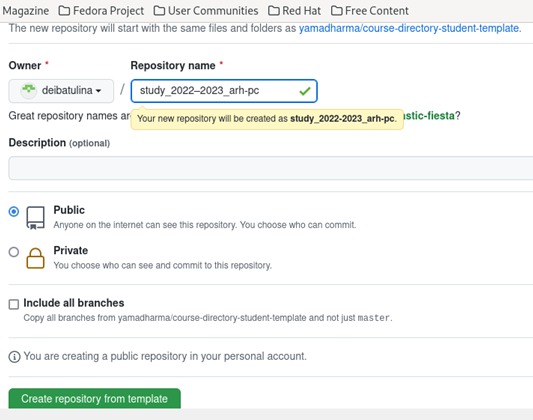


Рис. 16: Задание имени репозитория

Открываем терминал и переходим в каталог «Архитектура компьютера». (рис. 17)

Рис. 17: Переход в созданный каталог «Архитектура компьютера»

Рис. 17: Переход в созданный каталог «Архитектура компьютера»

Выполняем клонирование в каталог arch-pc, предварительно скопировав ссылку на репозиторий на гитхаб. (рис. [-@fig:019], [-@fig:020])

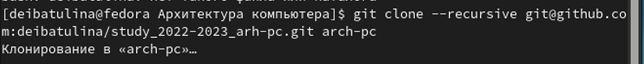


Рис. 18: Клонирование репозитория в каталог arch-pc

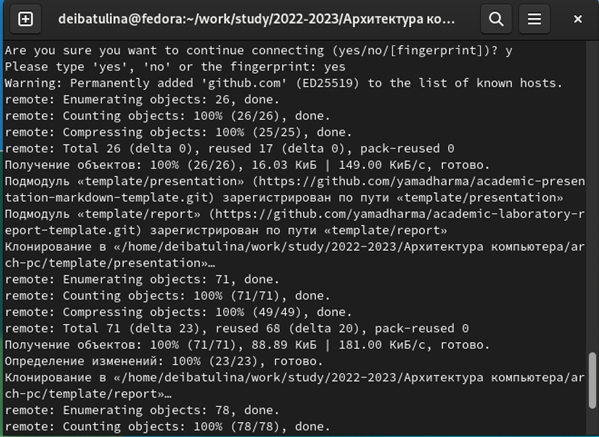


Рис. 19: Выполнение клонирования

Далее мы переходим в каталог курса с помощью команды cd. (рис. 20)

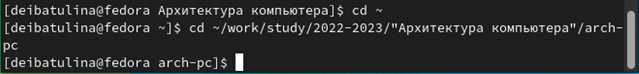


Рис. 20: Переход в каталог курса

Удаляем лишние файлы посредством команды rm, изученной на прошлой лабораторной работе. (рис. 21)

Рис. 21: Удаление лишних файлов

Рис. 21: Удаление лишних файлов

Вводим команды для создания каталога (используя команды echo >). (рис. 22)

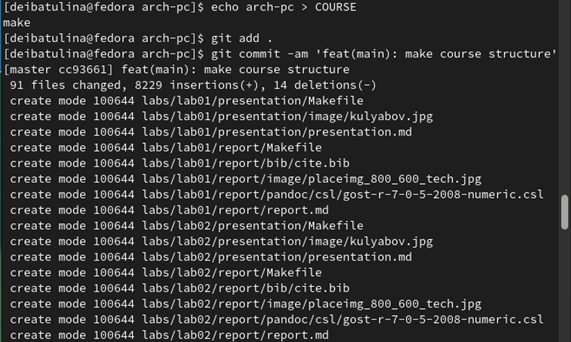


Рис. 22: Создание курса «Архитектура компьютера», его структуры

Отправляем файлы на сервер. (рис. 23)

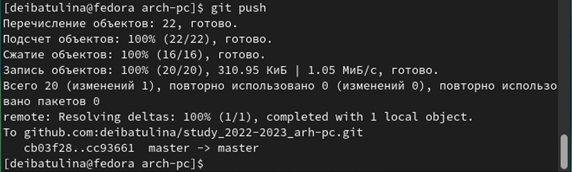


Рис. 23: Отправка файлов на сервер

Проверяем правильность создания рабочего пространства в локальном репозитории (рис. 24) и на странице Гитхаб. (рис. 25, 26)

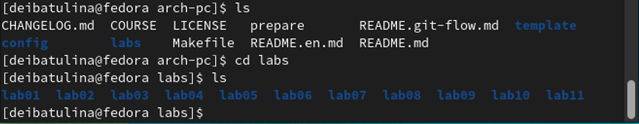


Рис. 24: Проверка правильности создания рабочего пространства в локальном репозитории

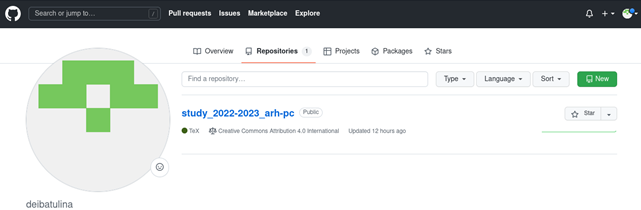


Рис. 25: Проверка правильности создания рабочего пространства на веб-странице Гитхаб

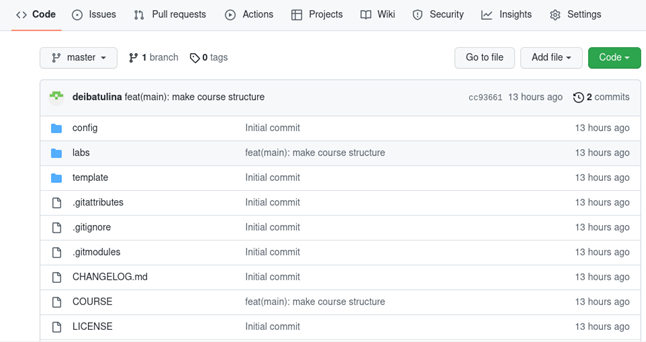


Рис. 26: Проверка правильности создания рабочего пространства на веб-странице Гитхаб

Переходим в подкаталог labs. (рис. 27)

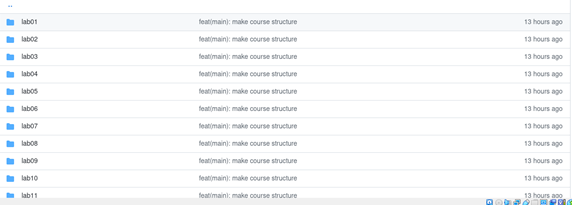


Рис. 27: Переход в подкаталог labs

Вывод: все каталоги созданы верно, иерархия соблюдена.

Далее мы переходим в подкаталог lab01/report, согласно практическому заданию, требуется загрузить отчёт на лабораторную работу №1. Лабораторная работа №2 загружается в соответствующий подкаталог lab02. (рис. 28)

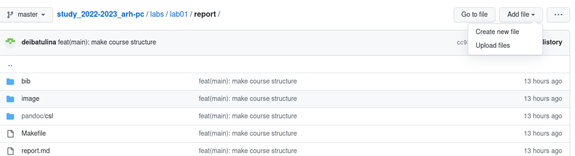


Рис. 28: Переход в подкаталог lab01/report

Далее необходимо переместить отчёты по лабораторным работам в соответствующие каталоги GitHub: lab01/report, lab02/report, lab03/report. (рис. 29)

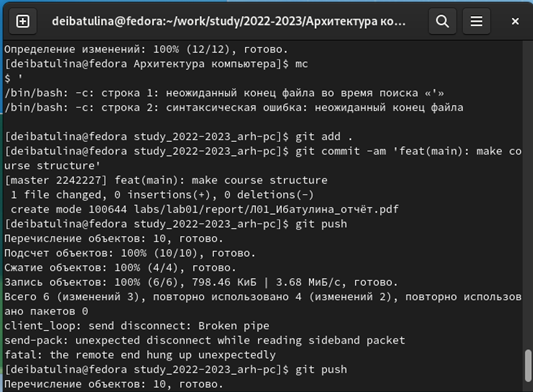


Рис. 29: Перемещение отчётов по лабораторным работам на GitHub посредством локального репозитория

# 5 Выводы

В результате выполнения лабораторной работы я научилась работать с системой контроля версий Git через командную строку и непосредственно на самом сайте GitHub. Узнала об идеологии системы контроля версий, как и для чего она используется.

# Список литературы

1. Руководство по выполнению лабораторной работы №3 “Система контроля версий Git”.