

Лабораторная работа №3

Система контроля версий Git

Коц Данила Богданович

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	10
5	Самостоятельная работа	17
6	Выводы	19
	Список литературы	20

Список иллюстраций

4.1	Создание учётной записи, заполнение данных.	10
4.2	Предварительная конфигурацию git и настройка utf-8 в выводе сообщений git.	11
4.3	Задание имени и настройка параметров.	11
4.4	Создание каталога для предмета «Архитектура компьютера». . . .	13
4.5	Создание репозитория.	13
4.6	Переход в каталог курса.	14
4.7	Загрузка ключа.	14
4.8	Удаление файлов.	15
4.9	Создание каталогов.	15
5.1	Отчёты.	17
5.2	Загрузка файлов.	18

Список таблиц

1 Цель работы

Целью работы является освоение процедуры оформления отчетов с помощью легковесного языка разметки Markdown

2 Задание

Создать аккаунт на Github, сделать конфигурацию git с помощью консоли, создать SSH ключ, рабочее пространство и репозиторий курса, настроив его.

3 Теоретическое введение

Системы контроля версий служат помощью для работы нескольких человек над одним проектом. Дерево проекта хранится в локальном или удалённом репозитории (к нему имеется доступ у всех участников работы). Внося изменения в содержание работы система контроля помогает фиксировать их, совместить изменения всех участников. В классических системах контроля используется централизованная модель, имеющая единый репозиторий 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, обеспечи-

вая таким образом, привилегированный доступ только одному пользователю, работающему с файлом. Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно доступна информация о том, кто из участников, когда и какие изменения вносил. Обычно такого рода информация хранится в журнале изменений, доступ к которому можно ограничить. В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным. Система контроля версий Git представляет собой набор программ команд-ной строки. Доступ к ним можно получить из терминала посредством ввода команды `git` с различными опциями. Основные команды `git`: `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 `git add имена_файлов` - добавить конкретные изменённые и/или созданные файлы и/или каталоги `git rm имена_файлов` - удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории) `git commit -am 'Описание коммита'` - сохранить все добавленные изменения и все изменённые файлы `git checkout -b имя_ветки` - создание новой ветки, базирующейся на текущей `git checkout имя_ветки` - переключение на некоторую ветку (при переключении на ветку, которой ещё нет в локальном репозитории, она будет создана и связана с удалённой) `git push origin имя_ветки` - отправка изменений конкретной ветки в центральный репозиторий `git merge --no-ff имя_ветки` - слияние ветки с текущим деревом `git branch -d имя_ветки` - удаление локальной уже слитой с основным деревом ветки `git branch -D имя_ветки` - принудительное удаление локальной

ветки `git push origin:имя_ветки` - удаление ветки с центрального репозитория

4 Выполнение лабораторной работы

1. Я создал учётную запись на сайте <https://github.com/> и заполнил основные данные.

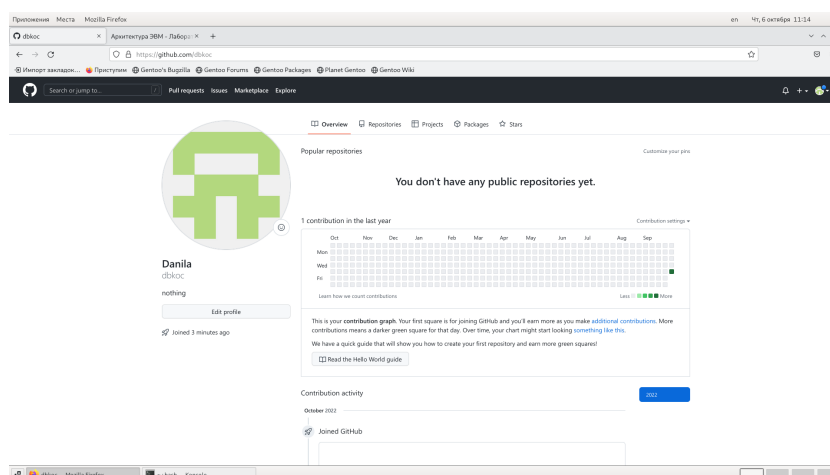


Рис. 4.1: Создание учётной записи, заполнение данных.

2. Я сделал предварительную конфигурацию git и настроил utf-8 в выводе сообщений git.

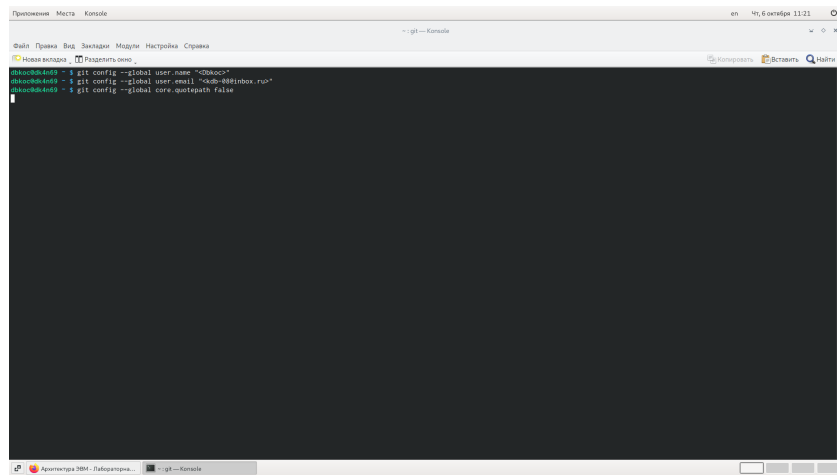


Рис. 4.2: Предварительная конфигурацию git и настройка utf-8 в выводе сообщений git.

3. Я задал имя начальной ветки, параметр autocrlf и параметр safecrlf.

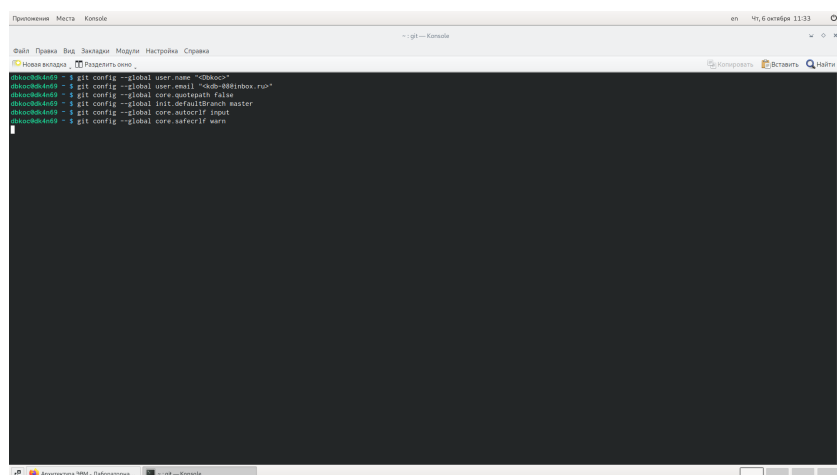


Рис. 4.3: Задание имени и настройка параметров.

4. Я сгенерировал пару ключей (приватный и открытый).

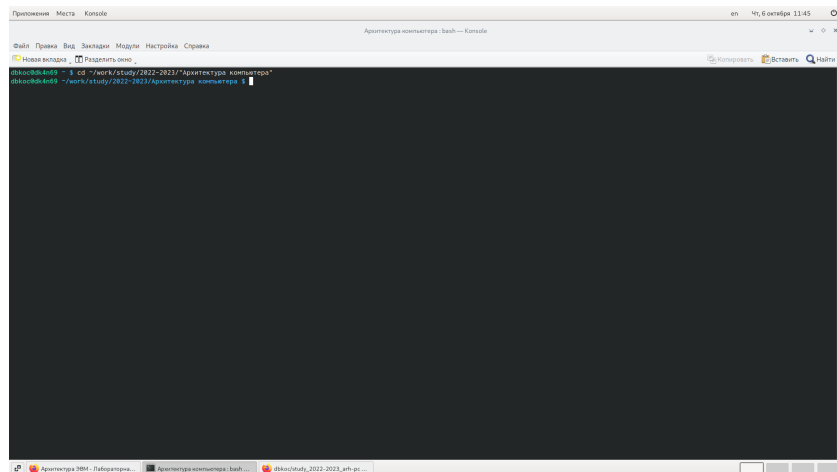


Рис. 4.6: Переход в каталог курса.

9. Я клонировал созданный репозиторий.

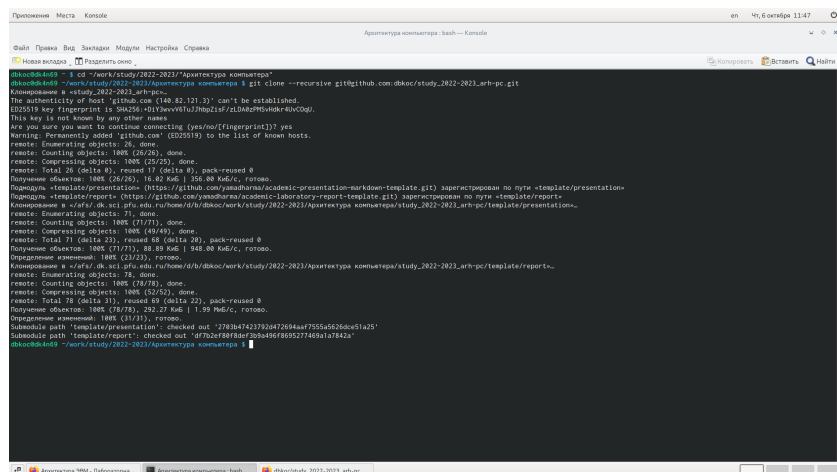


Рис. 4.7: Загрузка ключа.

10. Я удалил лишние файлы.

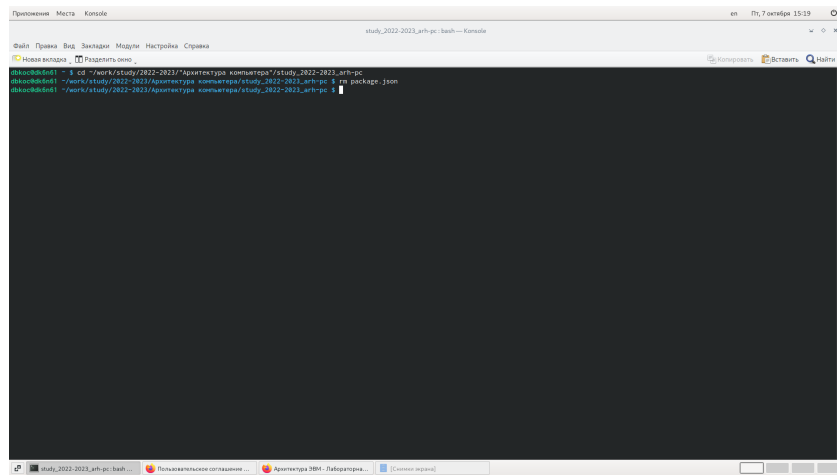


Рис. 4.8: Удаление файлов.

11. Я создал необходимые каталоги.

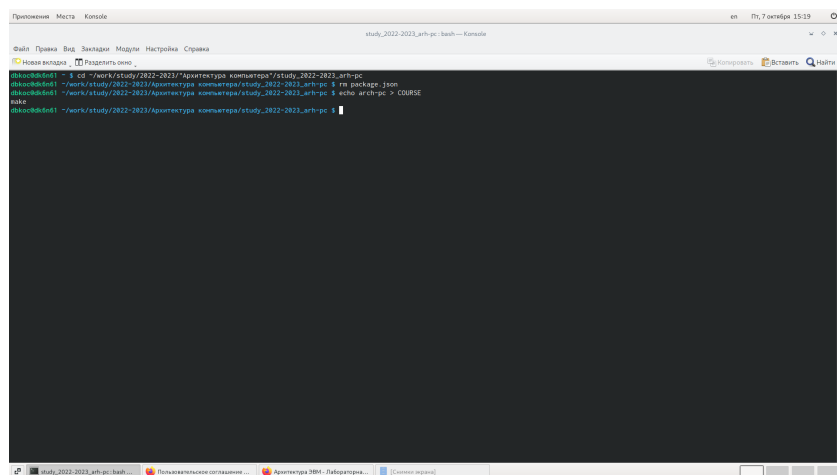
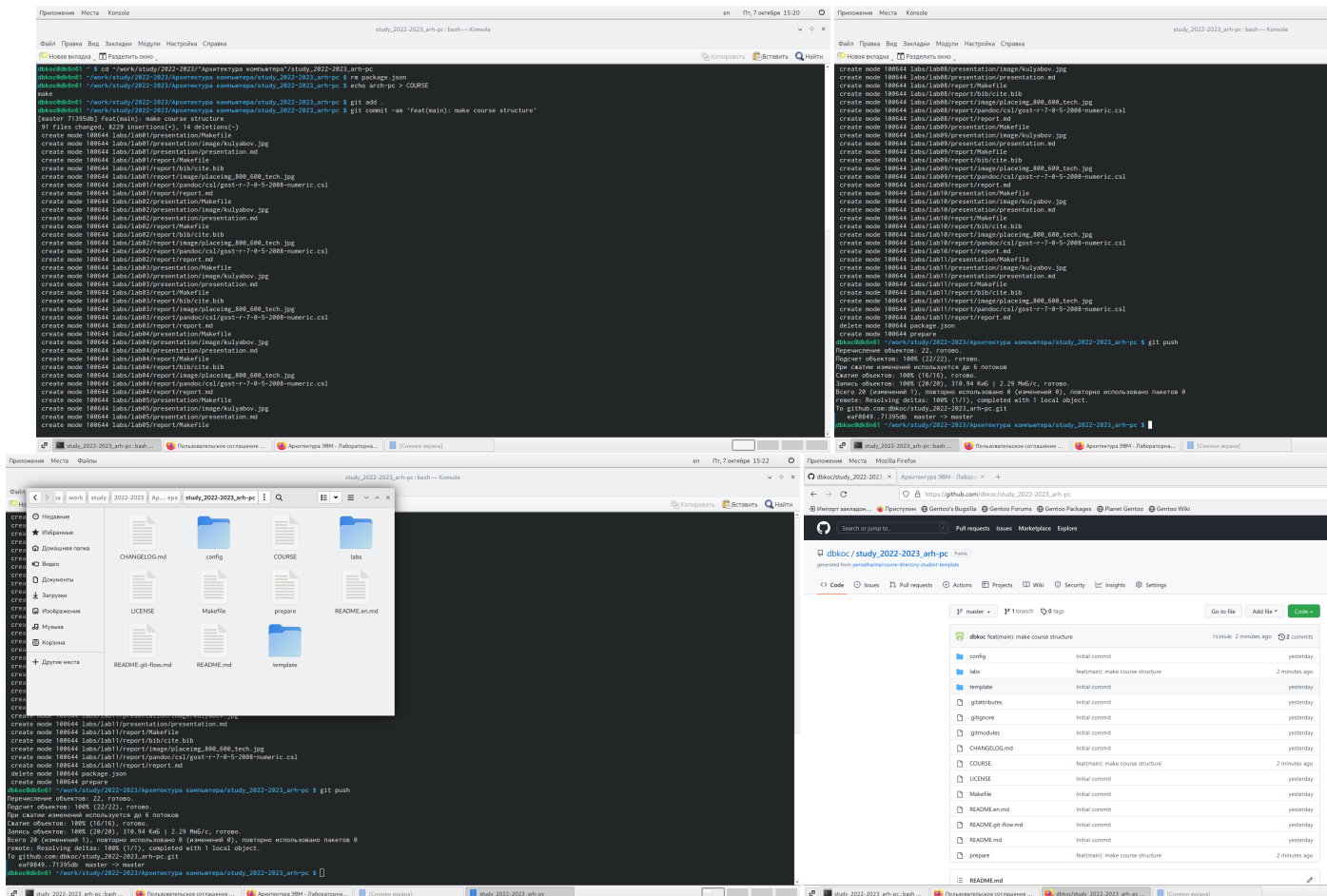


Рис. 4.9: Создание каталогов.

12. Я отправил файлы на сервер и проверил правильность создания иерархии рабочего пространства в локальном репозитории и на странице github.



5 Самостоятельная работа

1. Я скопировал отчёты по выполнению предыдущих лабораторных работ в соответствующие каталоги созданного рабочего пространства.

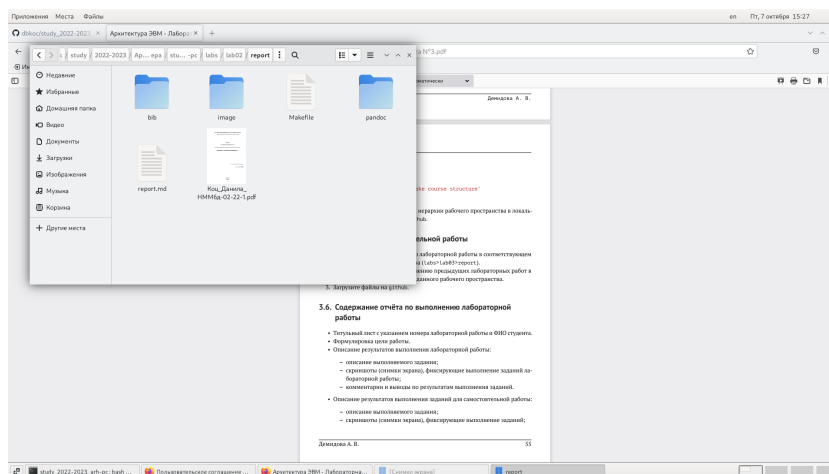


Рис. 5.1: Отчёты.

2. Я загрузил файлы на github.

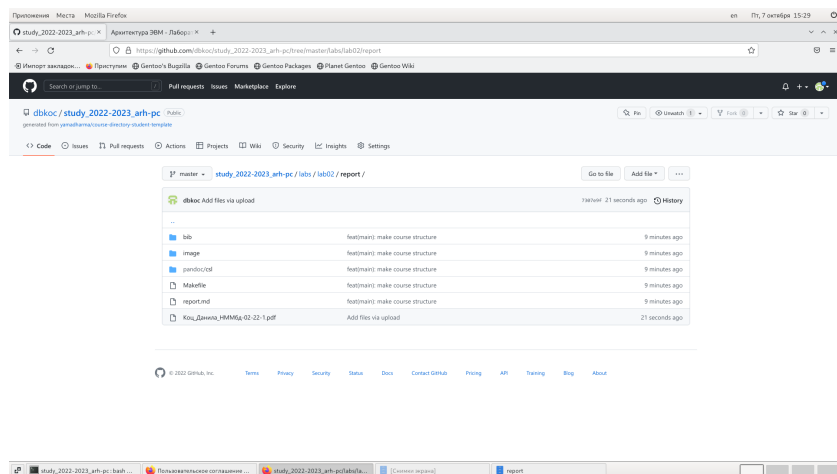


Рис. 5.2: Загрузка файлов.

6 Выводы

Я изучил идеологию и применение средств контроля версий и приобрёл практические навыки по работе с системой git.

Список литературы