

Отчёт по лабораторной работе № 3

Система контроля версий Git

Мокочунина Влада Сергеевна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	Выводы	17

Список иллюстраций

4.1	Создание учетной записи, заполнение данных.	9
4.2	Предварительная конфигурация git.	9
4.3	Настройка utf-8 в выводе сообщений git.	10
4.4	Задание имени и настройка параметров.	10
4.5	Генерация ключей.	11
4.6	Загрузка ключа.	11
4.7	Создание каталога для предмета «Архитектура компьютера». . . .	12
4.8	Страница репозитория с шаблоном курса.	12
4.9	Создание репозитория.	13
4.10	Переход в каталог курса.	13
4.11	Клонирование репозитория.	14
4.12	Удаление файлов.	14
4.13	Создание каталогов.	15
4.14	Отправка файлов.	15
4.15	Правильность иерархии рабочего пространства.	16
4.16	Отчеты.	16
4.17	Загрузка файлов.	16

List of Tables

1 Цель работы

Целью работы является изучение идеологии и применения средств контроля версий. Приобретение практических навыков по работе с системой git.

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 версий используется централизованная модель, имеющая единый репозиторий для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Пользователь перед началом работы посредством определённых команд может получить нужную ему версию файлов. После внесения изменений, новая версия размещается в хранилище (предыдущие версии не удаляются из центрального хранилища). Дельта-компрессия — сохранение только изменения между последовательными версиями. Она позволяет уменьшить объём хранимых данных. При помощи системы контроля версий можно отслеживать и разрешать конфликты. Изменения можно объединять, отменять, вручную выбирать нужную версию, блокировать файлы, что даёт привилегированный доступ только одному пользователю, работающему с файлом. Системы контроля версий обеспечивают дополнительные функциональные возможности: поддерживают работу с несколькими версиями одного файла, сохраняя общую историю изменений. Имеется доступ информация о том, кто из участников, когда и какие изменения вносил. В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным. Система контроля версий Git - набор программ командной строки. Доступ к ним

можно получить из терминала при помощи ввода команды `git` с различными опциями. Основные команды `git`: `git init` создание основного дерева репозитория, `git pull` получение обновлений 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 .` добавление всех изменённых или созданных файлов или каталогов, `git add имена_файлов` добавить конкретные изменённые файлы, `git rm` удалить файл и/или каталог из индекса, `git commit -am` сохранить все добавленные изменения и все изменённые файлы, `git checkout -b` создание новой ветки, базирующейся на текущей, `git checkout имя_ветки` переключение на некоторую ветку (при переключении на ветку, которой ещё нет в локальном репозитории, `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 создала учётную запись на сайте <https://github.com/> и заполнила основные данные (рис. 4.1).

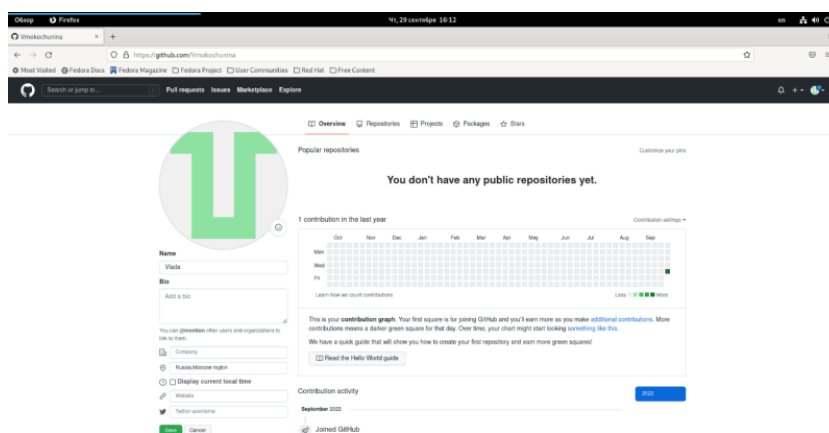


Рис. 4.1: Создание учетной записи, заполнение данных.

2. Я сделала предварительную конфигурацию git.

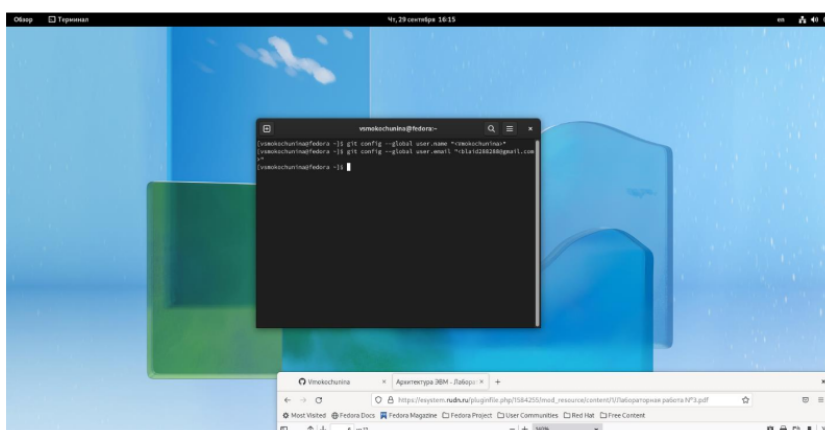


Рис. 4.2: Предварительная конфигурация git.

3. Я настроила utf-8 в выводе сообщений git.

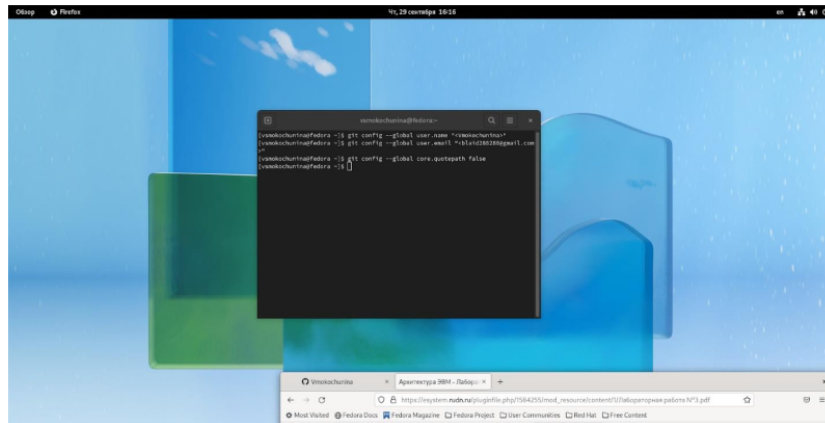


Рис. 4.3: Настройка utf-8 в выводе сообщений git.

4. Я задала имя начальной ветки, параметр autocrlf и параметр safecrlf.

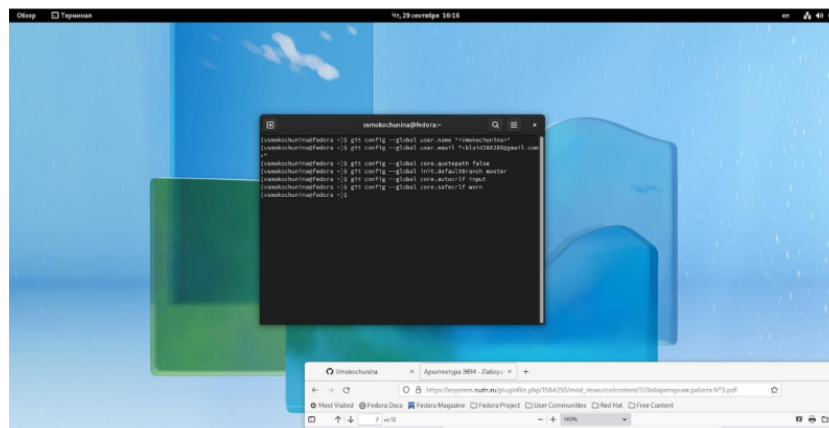


Рис. 4.4: Задание имени и настройка параметров.

5. Я сгенерировала пару ключей (приватный и открытый).

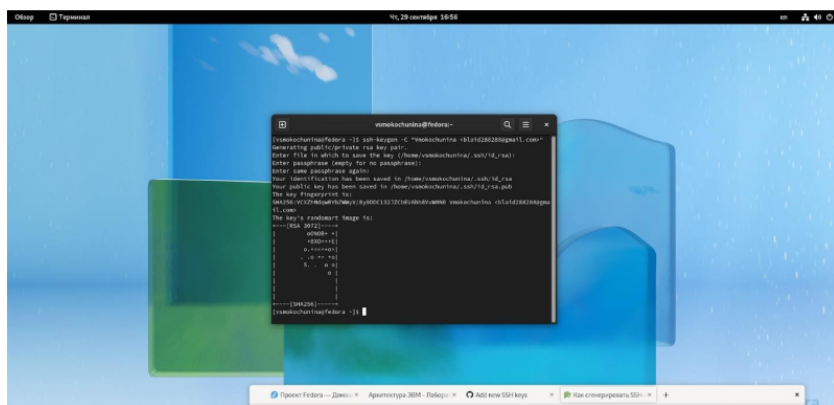


Рис. 4.5: Генерация ключей.

6. Я загрузила сгенерённый открытый ключ.

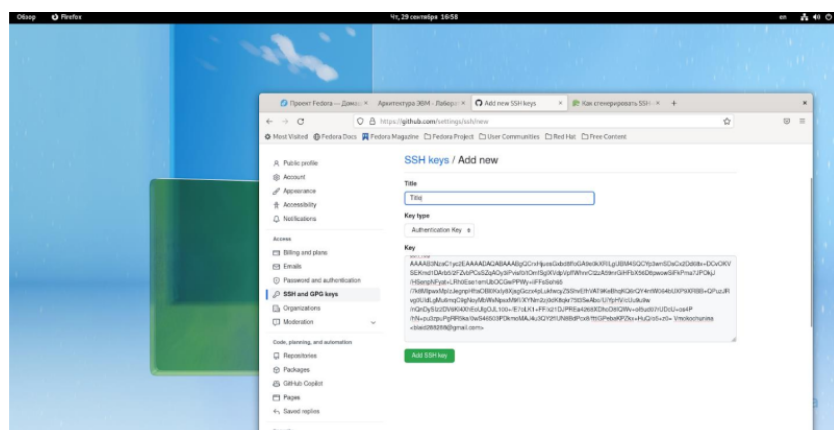


Рис. 4.6: Загрузка ключа.

7. Я создала каталог для предмета «Архитектура компьютера».

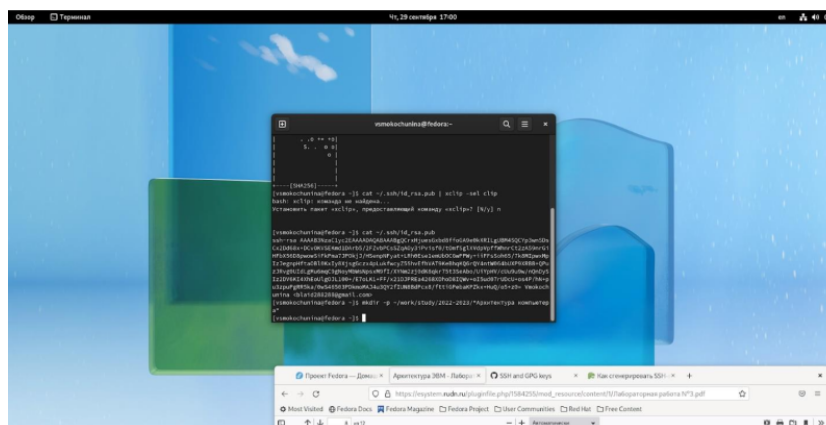


Рис. 4.7: Создание каталога для предмета «Архитектура компьютера».

8. Я перешла на страницу репозитория с шаблоном курса и выбрала Use this template.

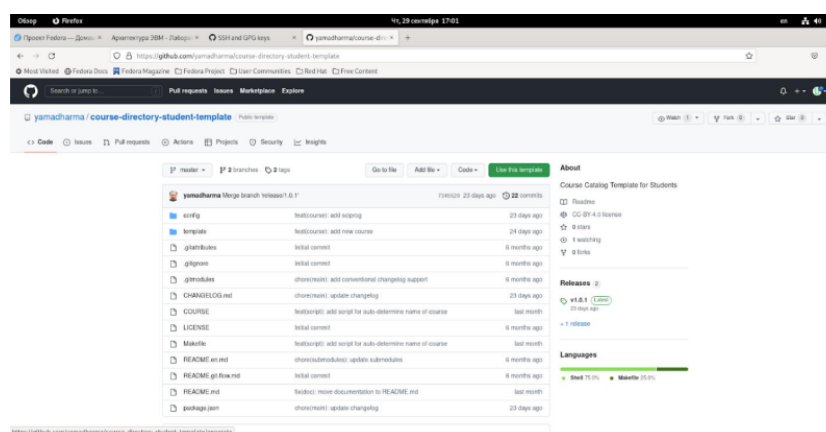


Рис. 4.8: Страница репозитория с шаблоном курса.

9. В открывшемся окне я задала имя репозитория и создала его.

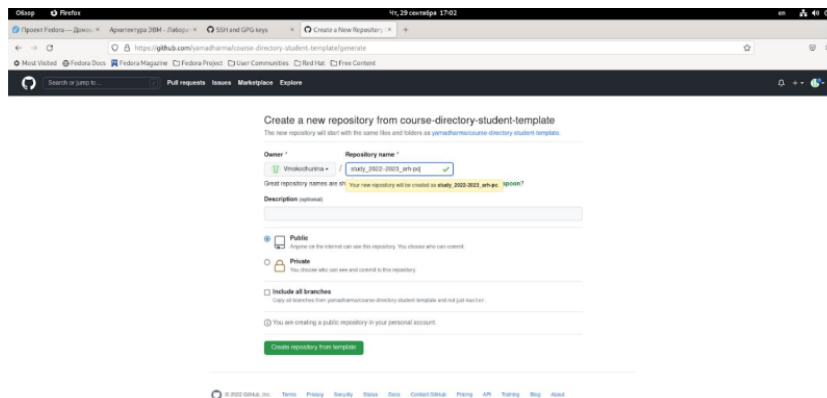


Рис. 4.9: Создание репозитория.

10. С помощью терминала я перешла в каталог курса.

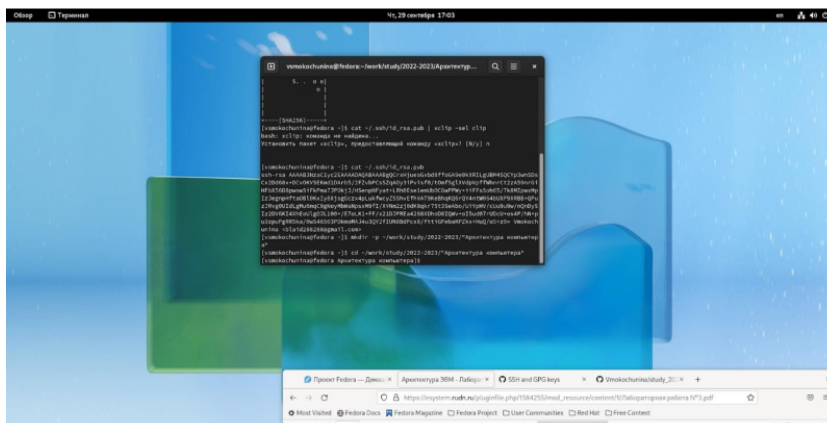


Рис. 4.10: Переход в каталог курса.

11. Я клонировала созданный репозиторий.

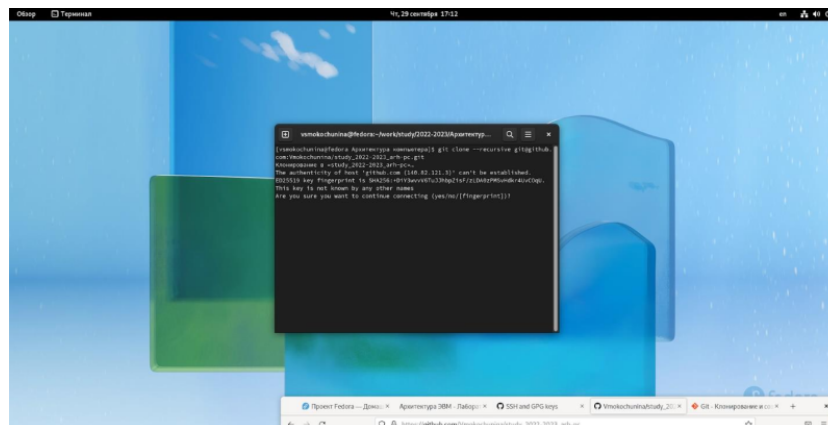


Рис. 4.11: Клонирование репозитория.

12. Я перешла в каталог курса и удалила лишние файлы.

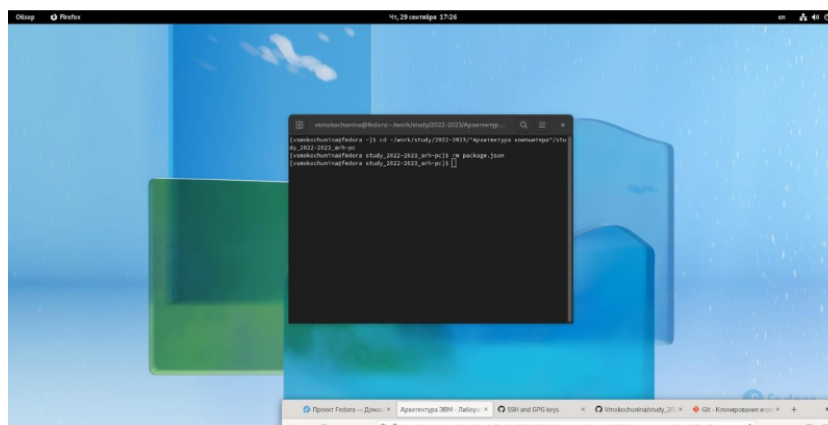


Рис. 4.12: Удаление файлов.

13. Я создала необходимые каталоги.

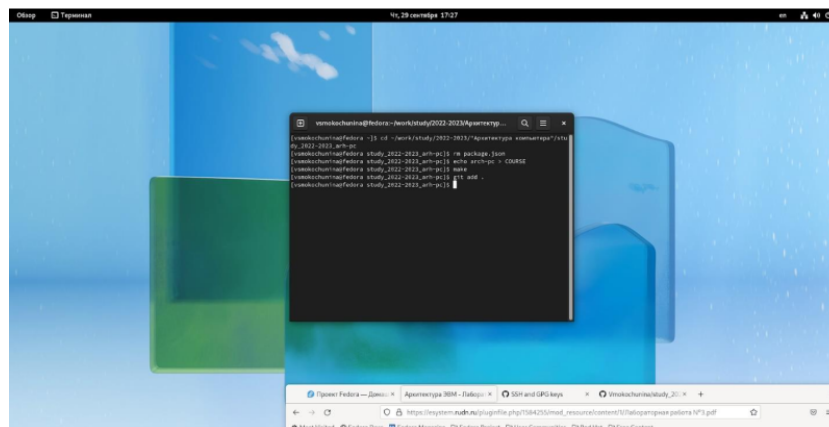


Рис. 4.13: Создание каталогов.

14. Я отправила файлы на сервер.

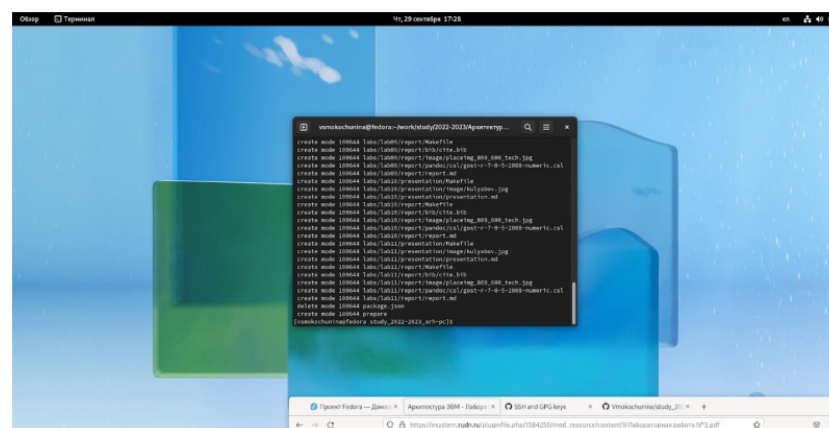


Рис. 4.14: Отправка файлов.

15. Я проверила правильность создания иерархии рабочего пространства.

5 Выводы

Я изучила идеологию и применение средств контроля версий, приобрела практические навыки по работе с системой git.