Отчёт по лабораторной работе №2

Дисциплина: Архитектура компьютера

Осина Виктория Александровна

Содержание

# 1 1. Цель работы

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

# 2 2. Задание

1. Настройка github.
2. Базовая настройка git.
3. Создание SSH 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 3. Теоретическое введение

Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещать изменения, произведённые разными участниками проекта, производить откат к любой более ранней версии проекта, если это требуется.   
В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория 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, обеспечивая 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. Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.  
Система контроля версий git представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды git с различными опциями. Благодаря тому, что Git является распределённой системой контроля версий, резервную копию локального хранилища можно сделать простым копированием или архивацией.

# 4 4. Выполнение лабораторной работы.

## 4.1 4.1. Настройка github.

У меня уже создана учетная запись в github и заполнены основные данные. (рис. [1](#fig:001))

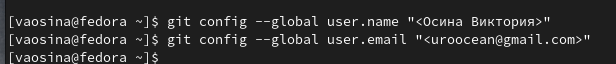


Figure 1: Рис. 1. Учетная запись в github

## 4.2 4.2. Базовая настройка git.

Чтобы сделать предварительную конфигурацию git, открываю терминал и ввожу необходимые команды, указав свои имя и email. (рис. [2](#fig:002))

Figure 2: Рис. 2. Конфигурация git

Figure 2: Рис. 2. Конфигурация git

Настраиваю utf-8 в выводе сообщений git. (рис. [3](#fig:003))

Figure 3: Рис. 3. Настройка вывода сообщений git

Figure 3: Рис. 3. Настройка вывода сообщений git

Задаю имя начальной ветки (будем называть её master). (рис. [4](#fig:004))

Figure 4: Рис. 4. Именование начальной ветки

Figure 4: Рис. 4. Именование начальной ветки

Задаю параметр autocrlf. (рис. [5](#fig:005))

Figure 5: Рис. 5. Параметр autocrlf

Figure 5: Рис. 5. Параметр autocrlf

Задаю параметр safecrlf. (рис. [6](#fig:006))

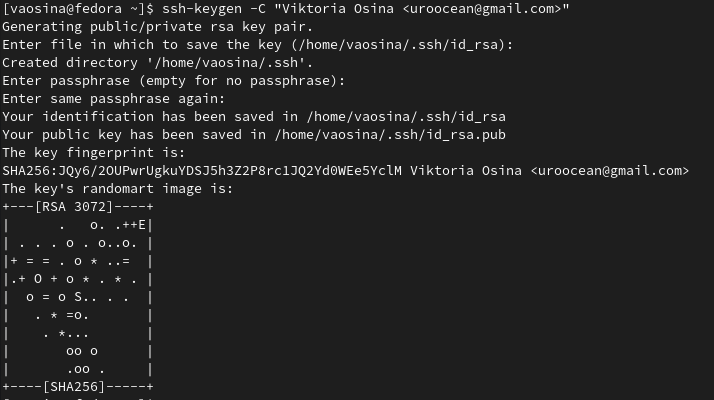


Figure 6: Рис. 6. Параметр safecrlf

## 4.3 4.3. Создание SSH ключа.

Для последующей идентификации пользователя на сервере репозиториев необходимо сгенерировать пару ключей (приватный и открытый), поэтому я прописываю команду ssh-keygen -C “Имя Фамилия [work@mail](mailto:work@mail)”, указав своё имя и фамилию, а также свою почту. (рис. [7](#fig:007))

Figure 7: Рис. 7. Генерация ключей (приватного и открытого)

Figure 7: Рис. 7. Генерация ключей (приватного и открытого)

Эти ключи автоматически сохранятся в каталоге ~/.ssh/.

Далее необходимо загрузить сгенерированный открытый ключ, поэтому я захожу на сайт http://github.org/ под своей учётной записью и перехожу в меню Settings. (рис. [8](#fig:008))

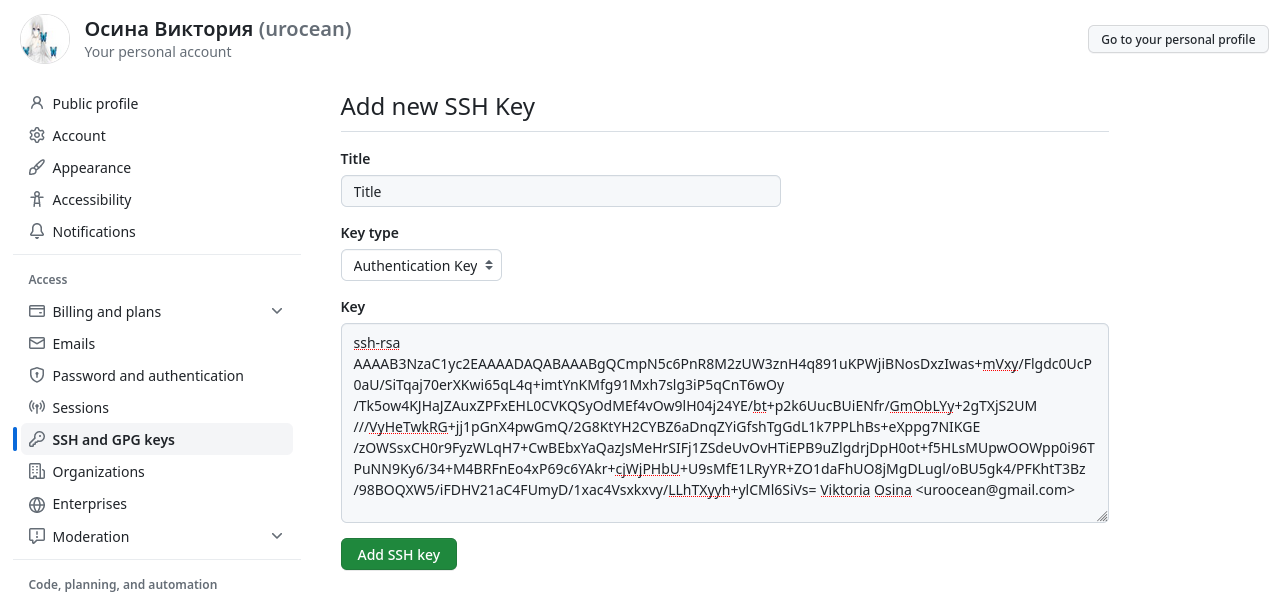


Figure 8: Рис. 8. Переход в настройки

После этого выбираю в боковом меню SSH and GPG keys (рис. [9](#fig:009))

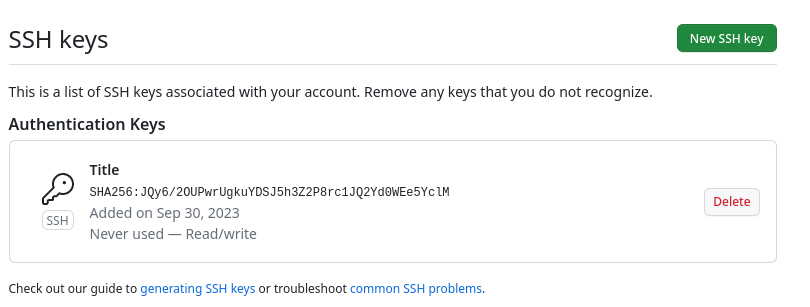


Figure 9: Рис. 9. Выбор в боковом меню SSH and GPG keys

и нажимаю кнопку New SSH key. (рис. [10](#fig:010))

Figure 10: Рис. 10. Создание New SSH key

Figure 10: Рис. 10. Создание New SSH key

Далее копирую из локальной консоли ключ в буфер обмена при помощи команды cat ~/.ssh/id\_rsa.pub | xclip -sel clip. (рис. [11](#fig:011))

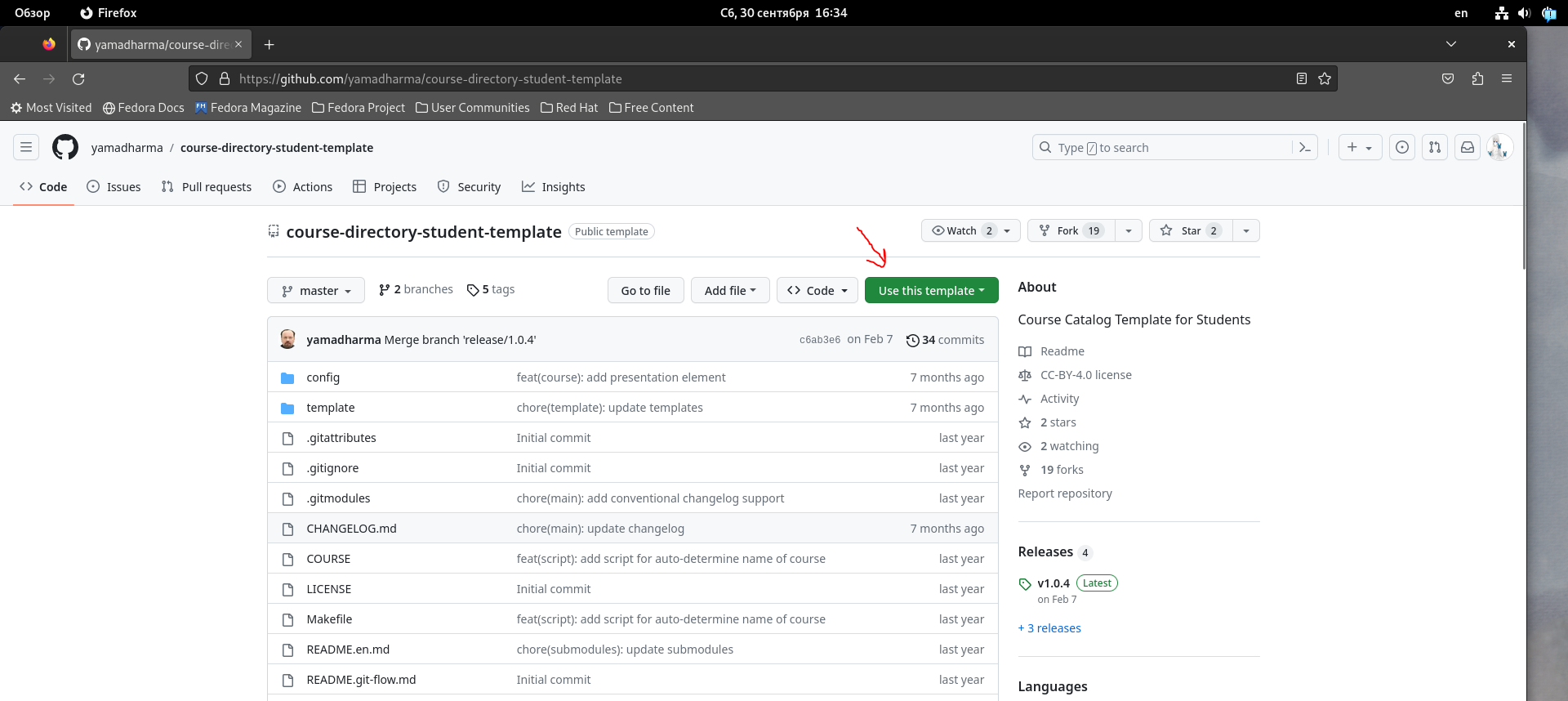


Figure 11: Рис. 11. Копирование ключа из локальной консоли

Теперь вставляю этот ключ в появившееся на сайте поле и указываю для ключа имя (Title). (рис. ??) и (рис. [13](#fig:013))

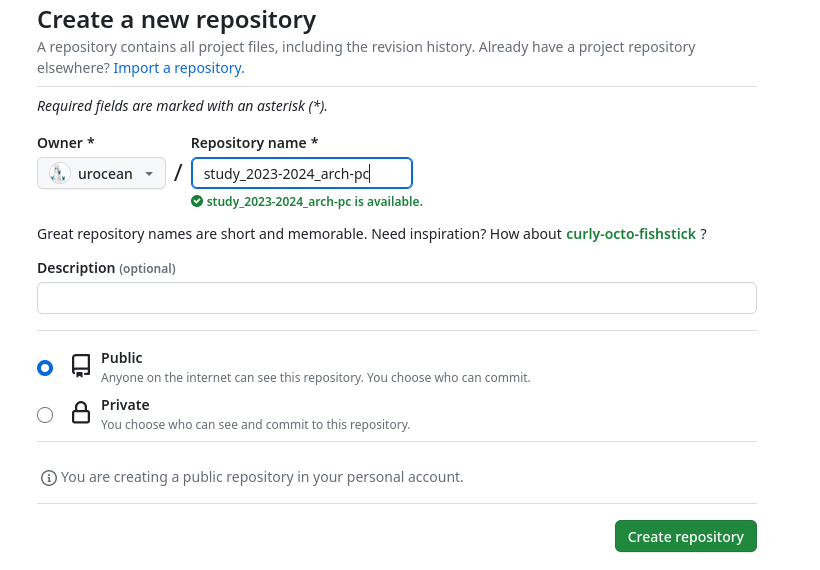


Figure 12: Рис. 12. Создание SSH ключа

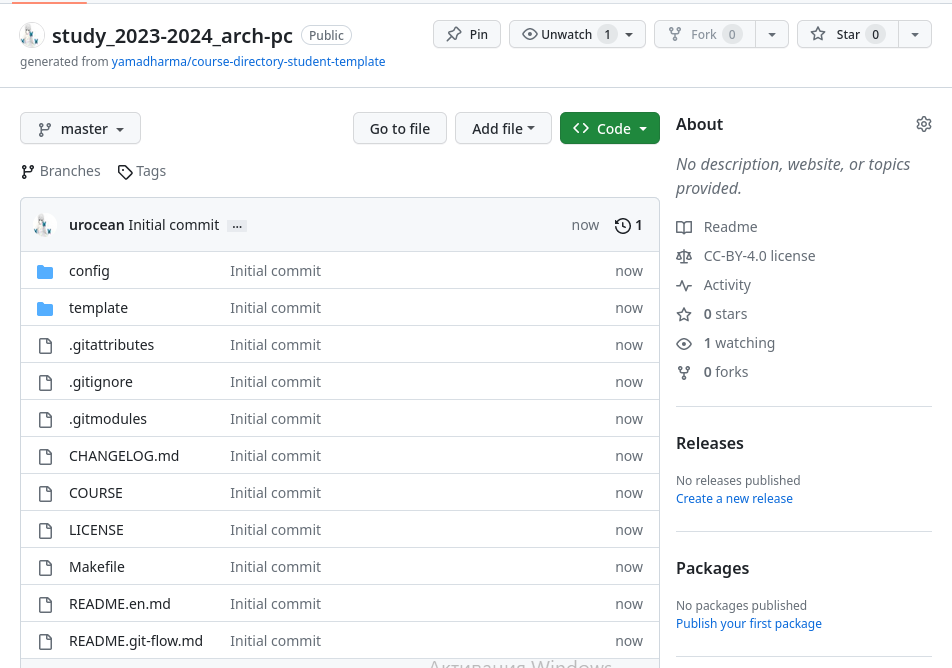


Figure 13: Рис. 13. Ключ SSH создан

## 4.4 4.4. Создание рабочего пространства и репозитория курса на основе шаблона.

При выполнении лабораторных работ следует придерживаться структуры рабочего пространства, которая должна располагаться в следующей иерархии: ~work/study/ └── 2023–2024/ └──Архитектура компьютера/ └── arch-pc/ └── labs/ └── lab01 └── lab02 └── lab03 …. Поэтому при помощи команды mkdir с ключом -p создаю каталог для предмета «Архитектура компьютера». (рис. [14](#fig:014))

Figure 14: Рис. 14. Создание каталога для предмета

Figure 14: Рис. 14. Создание каталога для предмета

## 4.5 4.5. Создание репозитория курса на основе шаблона

Перехожу на страницу репозитория с шаблоном https://github.com/yamadharma/course-directory-student-template и выбираю “Use this template”. (рис. [15](#fig:015)) и (рис. [16](#fig:016))

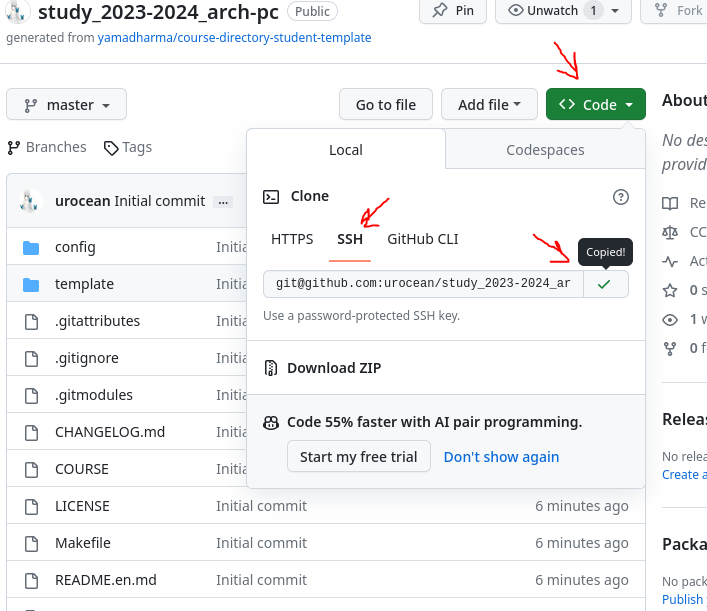


Figure 15: Рис. 15. Выбор ”Use this template”

Figure 16: Рис. 16. Выбор “Create a new repository“

Figure 16: Рис. 16. Выбор “Create a new repository“

В открывшемся окне задаю имя репозитория study\_2023–2024\_arch-pc и создаю репозиторий (кнопка Create repository). (рис. [17](#fig:017))

Figure 17: Рис. 17. Добавление имени репозитория

Figure 17: Рис. 17. Добавление имени репозитория

Репозиторий создан. (рис. [18](#fig:018))

Figure 18: Рис. 18. Просмотр репозитория

Figure 18: Рис. 18. Просмотр репозитория

На странице созданного репозитория копирую ссылку для клонирования (Code –> SSH –> Copy). (рис. [19](#fig:019))

Figure 19: Рис. 19. Копирование SSH ссылки

Figure 19: Рис. 19. Копирование SSH ссылки

Затем в терминале перехожу в каталог курса (рис. [20](#fig:020))

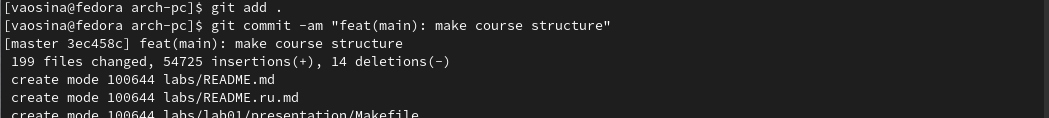


Figure 20: Рис. 20. Перемещение в каталог курса

и клонирую созданный репозиторий. (рис. [21](#fig:021))

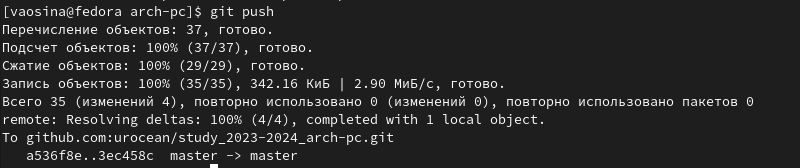


Figure 21: Рис. 21. Клонирование созданного репозитория

## 4.6 4.6. Настройка каталога курса

Перехожу в каталог курса. (рис. [22](#fig:022))

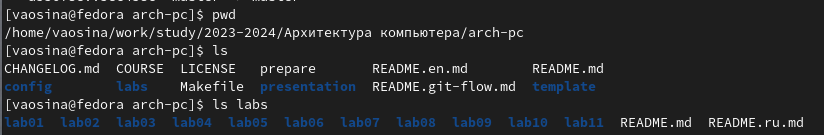


Figure 22: Рис. 22. Переход в каталог курса

Удаляю лишний каталог. (рис. [23](#fig:023))

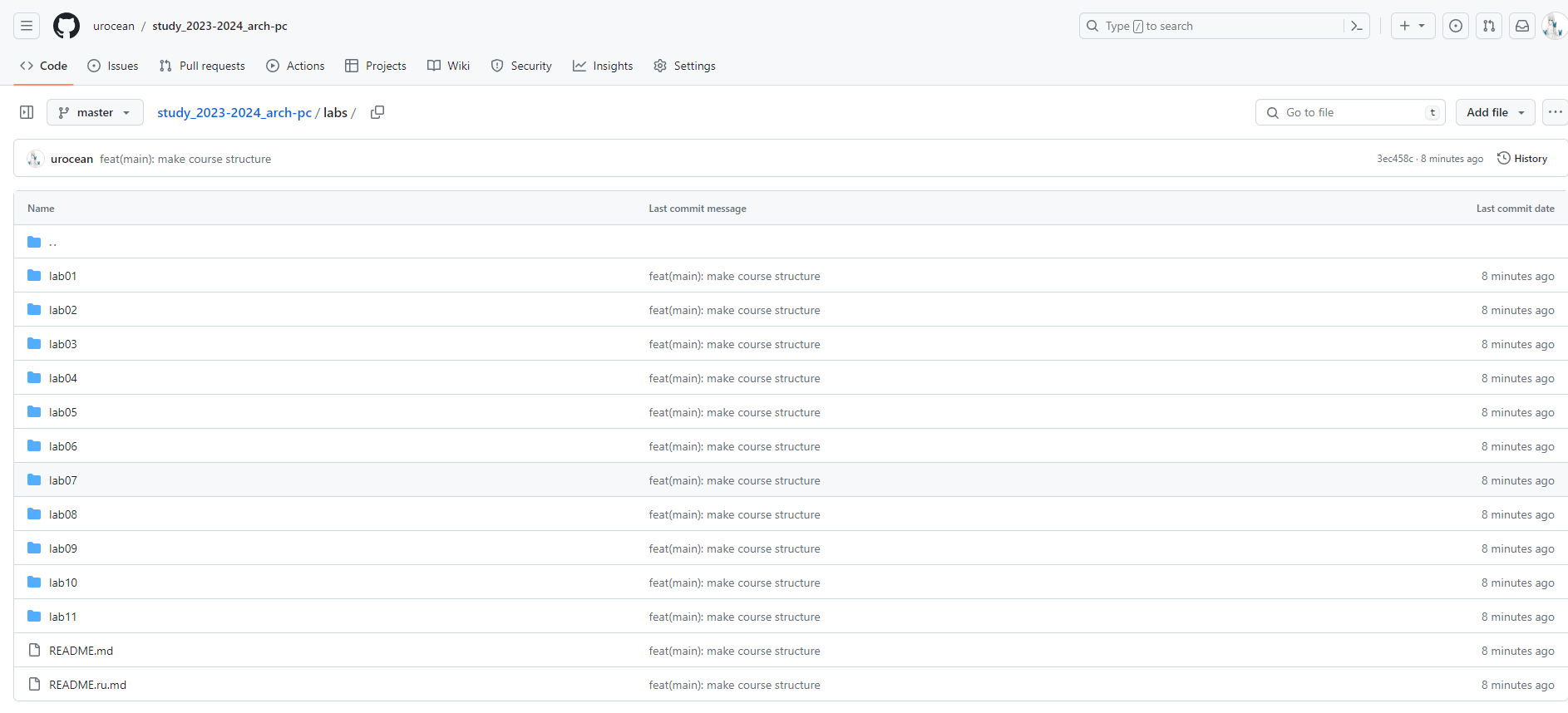


Figure 23: Рис. 23. Удаление каталога

Создаю необходимые каталоги. (рис. [24](#fig:024))

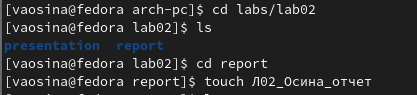


Figure 24: Рис. 24. Создание каталога

Отправляю файлы на сервер. (рис. [25](#fig:025)) и (рис. [26](#fig:026))

Figure 25: Рис. 25. Добавление изменения для сохранения и сохранение изменения

Figure 25: Рис. 25. Добавление изменения для сохранения и сохранение изменения

Figure 26: Рис. 26. Отправление файла на сервер

Figure 26: Рис. 26. Отправление файла на сервер

Проверяю правильность создания иерархии рабочего пространства. (рис. [27](#fig:027)) и (рис. [28](#fig:028))

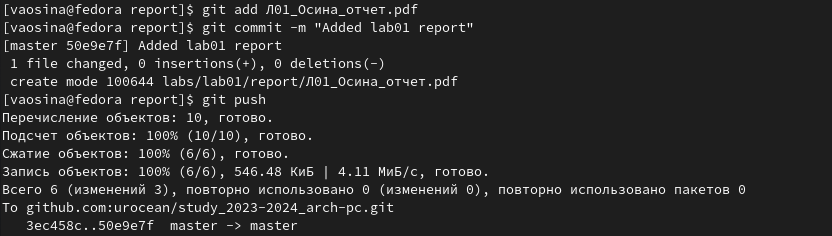


Figure 27: Рис.27. Просмотр иерархии рабочего пространства в локальном репозитории через терминал

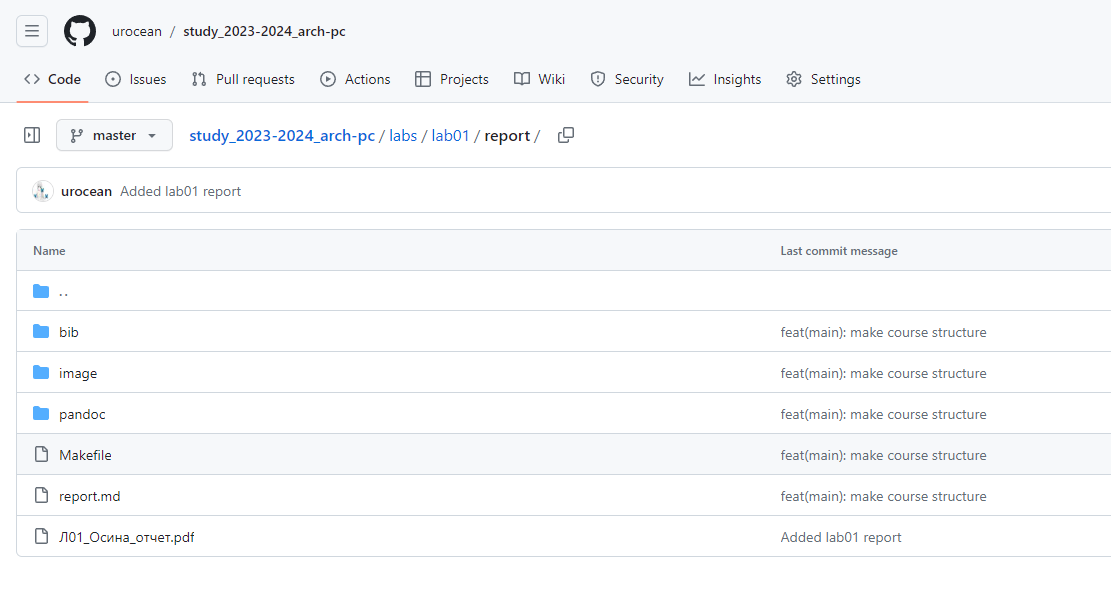


Figure 28: Рис. 28. Просмотр файлов на github

# 5 5. Задание для самостоятельной работы

##5.1. Создаю отчет по выполнению лабораторной работы в соответствующем каталоге рабочего пространства (labs>lab02>report). (рис. ??) Рис. 29. Создание файла с отчетом

Отчет я выполняю в текстовом редакторе Word и т. к. на текущий момент я всё ещё взаимодействую с отчетом по данной (второй) лабораторной работе, то на github файл с отчетом я загружу позднее по тому же алгоритму, по которому сейчас буду добавлять отчет о первой лабораторной работе

##5.2. Проверяю местонахождение файла с отчетом о первой лабораторной работе и копирую его в соответствующий каталог (labs>lab01>report). (рис. [29](#fig:030)) и (рис. [30](#fig:031))

Figure 29: Рис. 30. Проверка местонахождения отчета

Figure 30: Рис. 31. Копирование файла с отчетом в соответствующий каталог

##5.3. Загружаю файл на github (рис. [1](#fig:001))

Figure 31: Рис. 32. Отправление файла на сервер

Проверяю github и убеждаюсь в том, что файл с отчетом загружен: (рис. ??)

Figure 32: Рис. 33. Проверка на наличие файла с отчетом

# 6 Выводы

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

# 7 Список литературы

1. ТУИС – Архитектура ЭВМ ](https://esystem.rudn.ru/pluginfile.php/2089082/mod\_resource/content/0/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%962.%20%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0%20%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%8F%20%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D0%B9%20Git.pdf)