Отчёт по лабораторной работе №2

Дисциплина: архитектура компьютера

Кайнова Алина Андреевна

Содержание

# Цель работы

Изучить идеологию систем контроля версий и научиться применять их, в частности, приобрести навыки по работе с Git.

# Задание

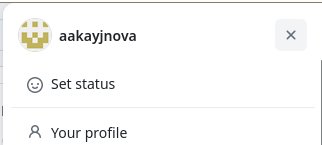
1. Настройка Github
2. Базовая настройка Git
3. Создание SSH ключа
4. Создание рабочего пространства и репозитория курса на основе шаблона
5. Создание репозитрия курса на основе шаблона
6. Настройка каталога курса
7. Задания для самостоятельной работы

# 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 необходимо. В классических системах контроля версий используется централизованная модель, пред- полагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Участник проекта 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 заблоки-ровать 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.

# Выполнение лабораторной работы

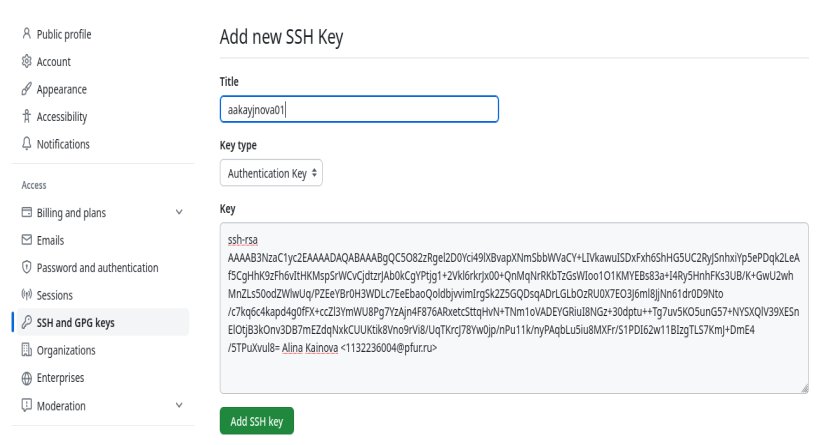
## 0.1 Настройка Github

* Создадим учётную запись на сайте GitHub 

## 0.2 Базовая настройка Git

* Сделаем предварительную конфигурацию Git рис. 2: Предварительная конфигурация Git
* Настроим utf-8 рис. 3: Настройка кодировки
* Зададим имя начальной ветви рис. 4: Создание имени для начальной ветки
* Зададим параметр autocrlf рис. 5: Параметр autocrlf
* Зададим параметр safecrlf рис. 6: Параметр safecrlf

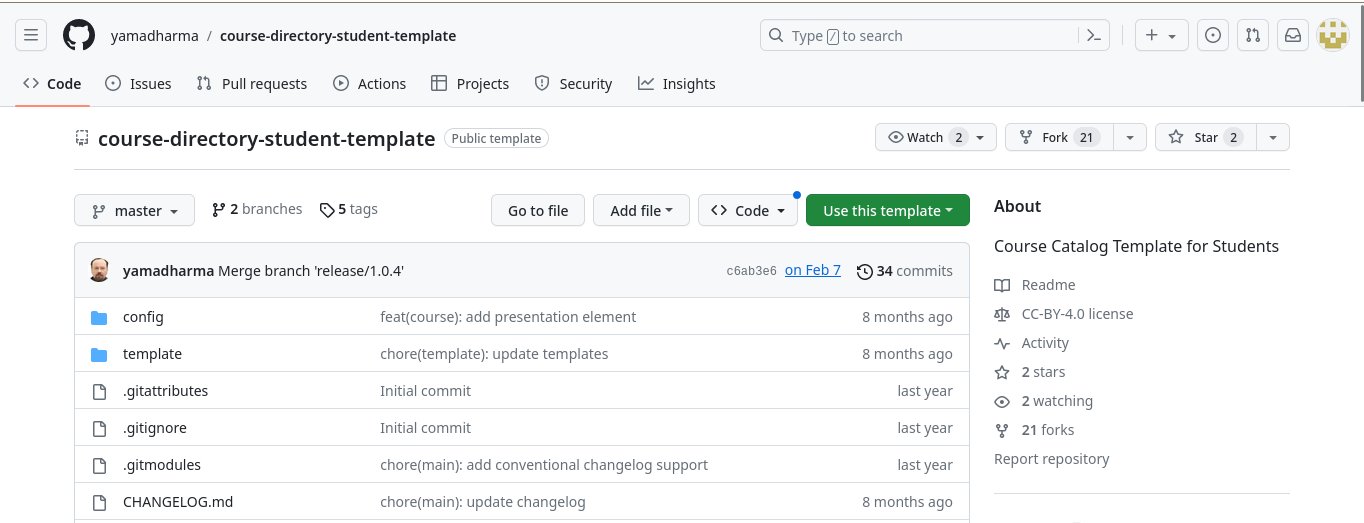
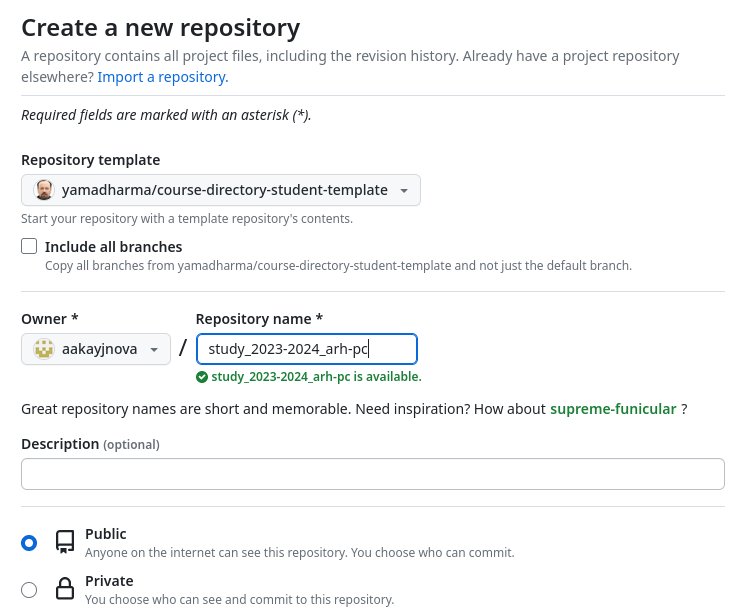
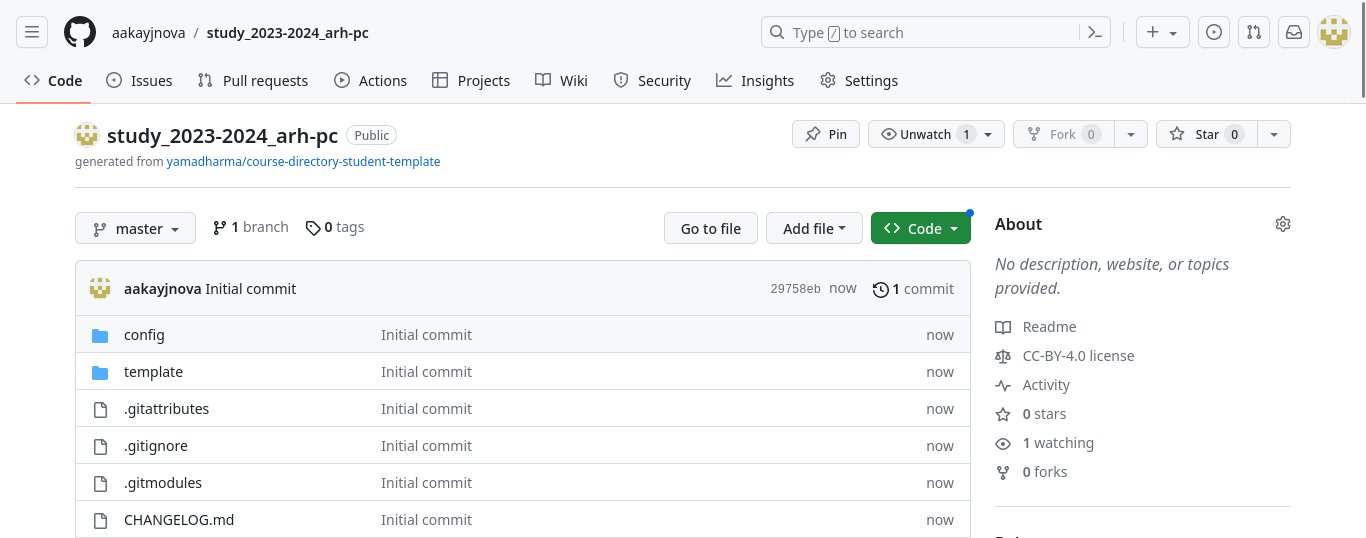
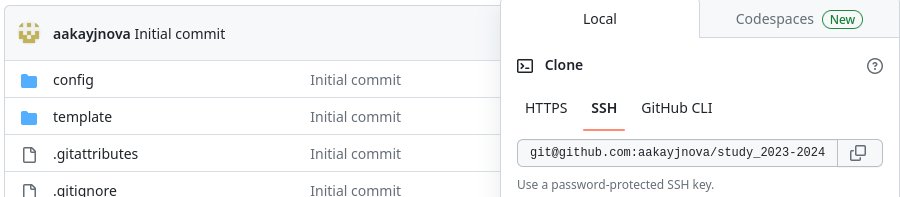
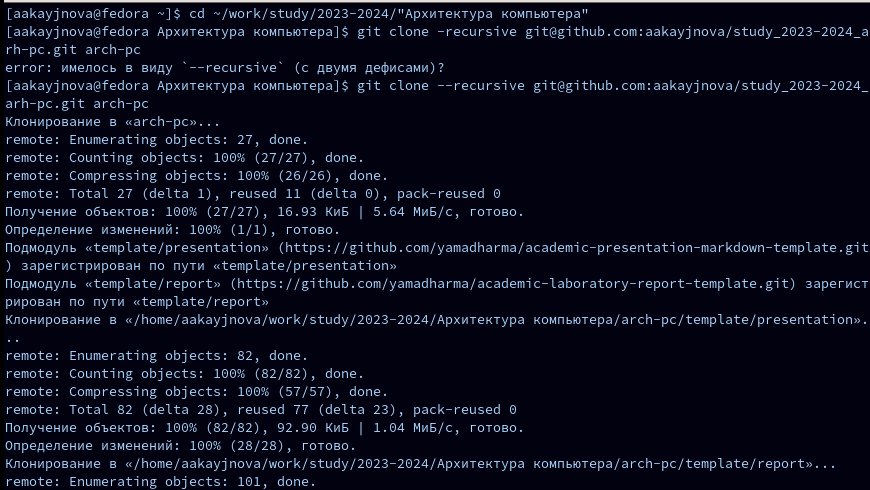
## 0.3 Создание SSH ключа

* Генерируем пару ключей рис. 7: Генерация SSH ключей
* Копируем ключ из локальной консоли рис. 8: Копирование содержимого файла
* Добавляем новый ключ 

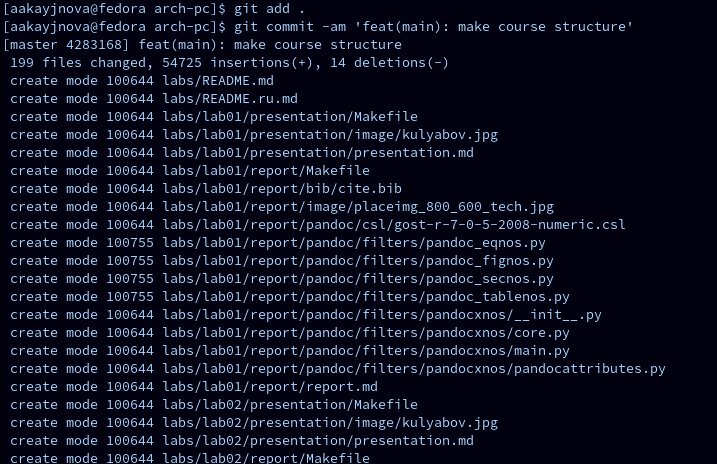
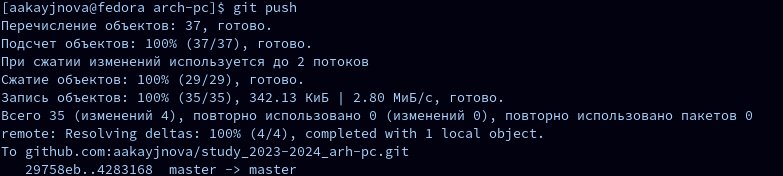
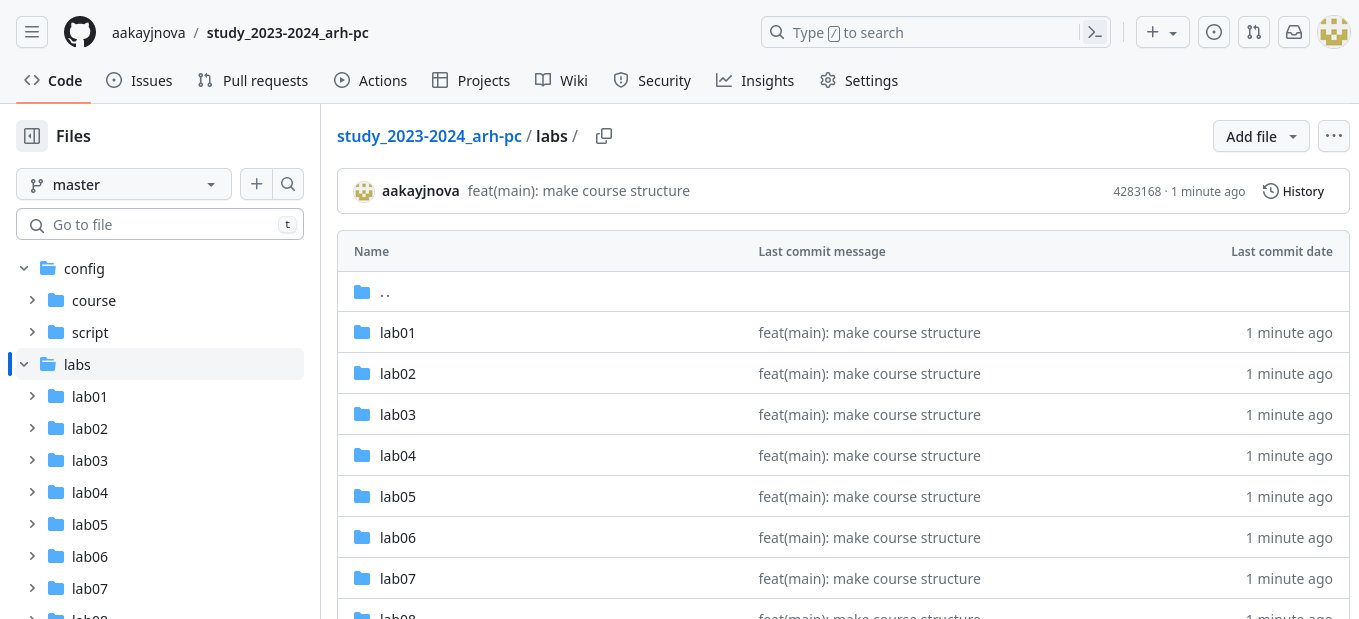
## 0.4 Создание рабочего пространства и репозитория курса на основе шаблона

* Создадим каталог для данного предмета через терминал и проверим правильность выполнения команды рис. 10: Создание рабочего пространства

## 0.5 Создание репозитория курса на основе шаблона

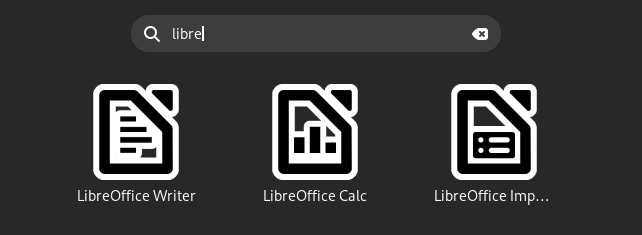
* В браузере переходим на страницу репозитория с шаблоном курса 
* Зададим имя репозитория и создадим его 
* Репозиторий создан 
* Переходим в созданный каталог курса через терминал рис. 14: Перемещение между директориями
* Копируем ссылку репозитория 
* Клонируем репозиторий, используя ссылку 

## 0.6 Настройка каталога курса

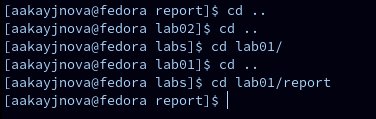
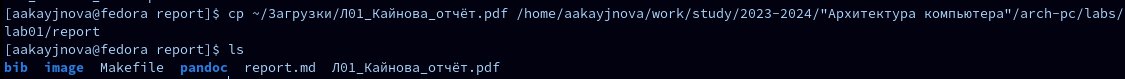
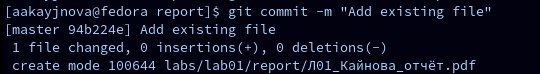
* Переходим в каталог arch-pc рис. 17: Перемещение между директориями
* Удаляем лишние файлы рис. 18: Удаление файлов
* Создаём необходимые каталоги рис. 19: Создание каталогов
* Добавляем все созданные в ходе работы каталоги и сохраняем изменения на сервере 
* Отправляем всё на сервер 
* Проверяем правильность выполнения работы на Github 

## 0.7 Выполнение заданий для самостоятельной работы

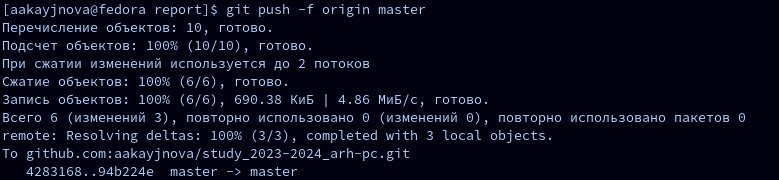
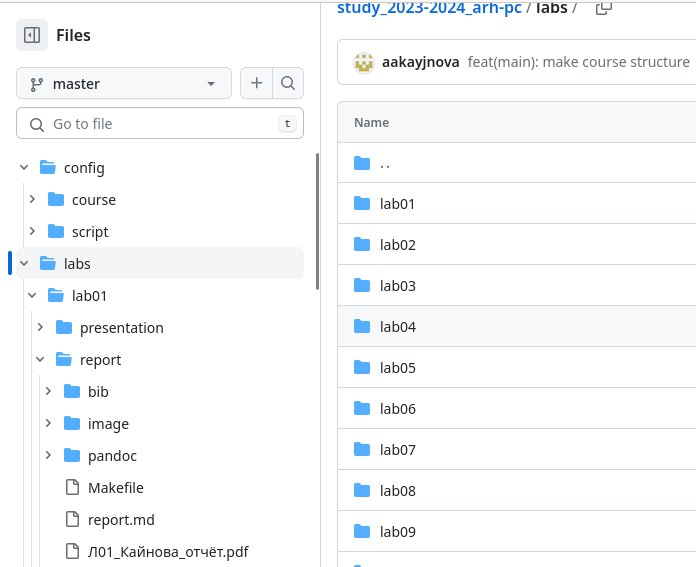
1. Отчёт по выполнению лабораторной работы в соответствующем каталоге рабочего пространства

* Переходим в директорию labs/lab02/report и создаём файл для отчёта по второй лабораторной работе рис. 23: Создание файла
* Оформляем отчёт в текстовом редакторе LibreOffice Writer, найдя его в меню приложений 

1. Копирование отчёта предыдущей лабораторной работы в соответствующие каталоги рабочего пространства

* Переходим в подкаталог lab01/report 
* Проверяем местонахождение файла с первой лабораторной работой рис. 26: Проверка местонахождения файла
* Копируем первую лабораторную работу и проверяем правильность выполнения команды 
* Добавляем созданные файлы на сервер рис. 28: Добавление файлов на сервер
* Сохраняем изменения на сервере 

1. Загрузка файлов на Github

* Отправляем в центральный репозиторий сохранённые изменения 
* Проверяем правильность выполнения работы на Github 

# 1 Выводы

В ходе данной лабораторной работы мы изучили структуру систем контроля версий и научились применять средства контроля версий по работе с ситемой Git.

# Список литературы

1. [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)