Отчёт по лабораторной работе №2

дисциплина: архитектура компьютеров

Ведьмина Александра Сергеевна

Table of Contents

# 1 Цель работы

Изучить идеологию и применение средств контроля версий, научиться работать с системой git.

# 2 Задание

1. Зарегистрироваться на сайте github.
2. Создать репозиторий курса.
3. Изучить основные команды git.
4. Выполнить задания для самостоятельной работы.
5. Создать отчёт по выполнению лабораторной работы.

# 3 Теоретическое введение

Системы контроля версий нужны для работы нескольких человек над одним проектом. Основное дерево проекта хранится в локальном или удалённом репозитории. Участники проекта имеют к нему доступ и могут видеть изменения, вносимые каждым из участников в содержимое проекта. Можно делать откат к любой более ранней версии проекта. В классических системах контроля версий используется централизованная модель (единый репозиторий для хранения файлов). Большинство функций выполняет специальный сервер. Перед началом работы пользователь получает нужную ему версию файлов и может разместить новую версию в хранилище, имея возможность вернуться к предыдущим версиям в любой момент. Более того, системы контроля версий способны 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.

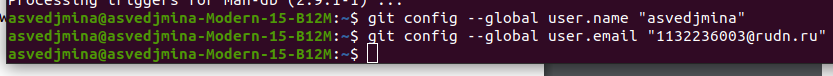
Система контроля версий Git - это набор программ командной строки, доступ к которым можно получить через терминал с помощью команды git и её опций.

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 изменения 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 - сохранить все добавленные изменения и все изменённые файлы git checkout -b - создание новой ветки, базирующейся на текущей git checkout - переключение на некоторую ветку (при переключении на ветку, которой ещё нет в локальном репозитории, она будет создана и связана с удалённой) git push origin - отправка изменений конкретной ветки в центральный репозиторий git merge –no-ff - слияние ветки с текущим деревом git branch -d - удаление локальной уже слитой с основным деревом ветки git branch -D - принудительное удаление локальной ветки git push origin - удаление ветки с центрального репозитория

Работать со своей веткой следует начинать с проверки и получения изменений из центрального репозитория. Это можно осуществить с помощью команд git checkout master, git pull, git checkout -b . После этого можно вносить изменения в локальном дереве. Если нужно сохранить все изменения в текущем каталоге, то используем git add . Отправляем в центральный репозиторий с помощью команды git push origin или git push.

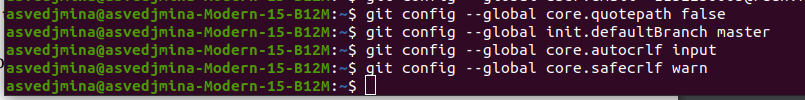
# 4 Выполнение лабораторной работы

Создаю учётную запись на сайте https://github.com/ , после чего делаю предварительную конфигурацию git.



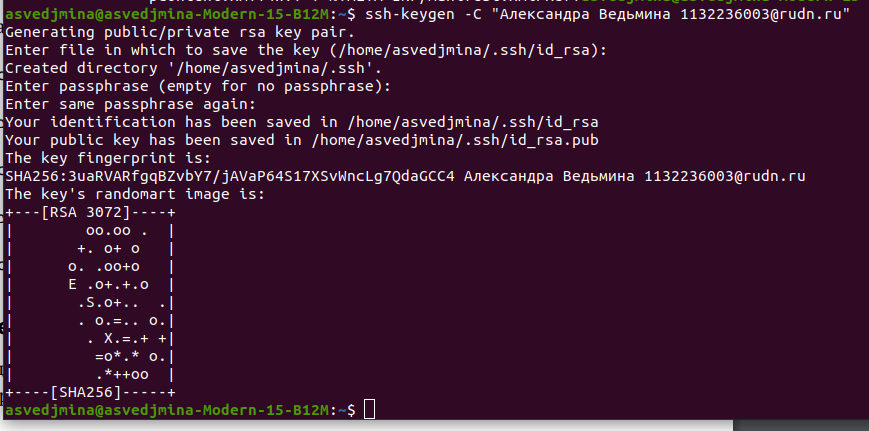
конфигурация git

Настраиваю utf-8 в выводе сообщений git, задаю имя master для начальной ветки, параметр autocrlf и safecrlf.



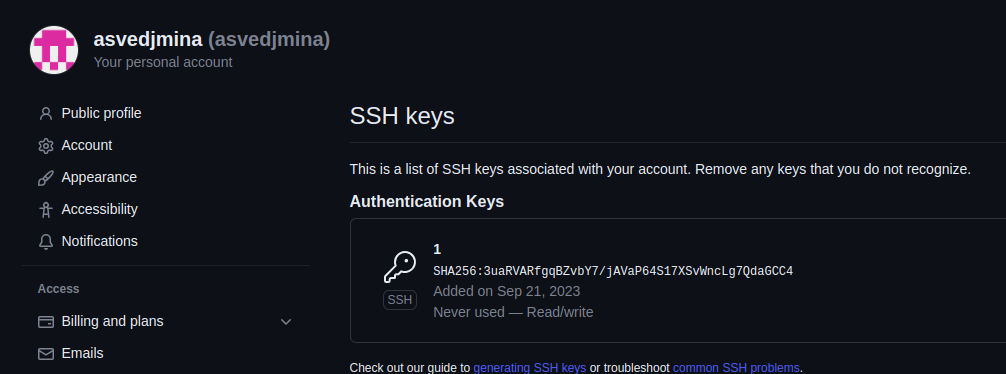
задание начальной ветки и параметров

Генерирую SSH-ключ.



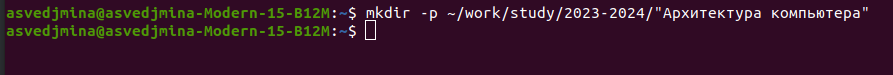
генерация ssh-ключа

Загружаю сгенерённый открытый ключ, зайдя на сайт http: //github.org/ под своей учётной записью и скопировав из локальной консоли ключ в буфер обмена.



добавление ключа

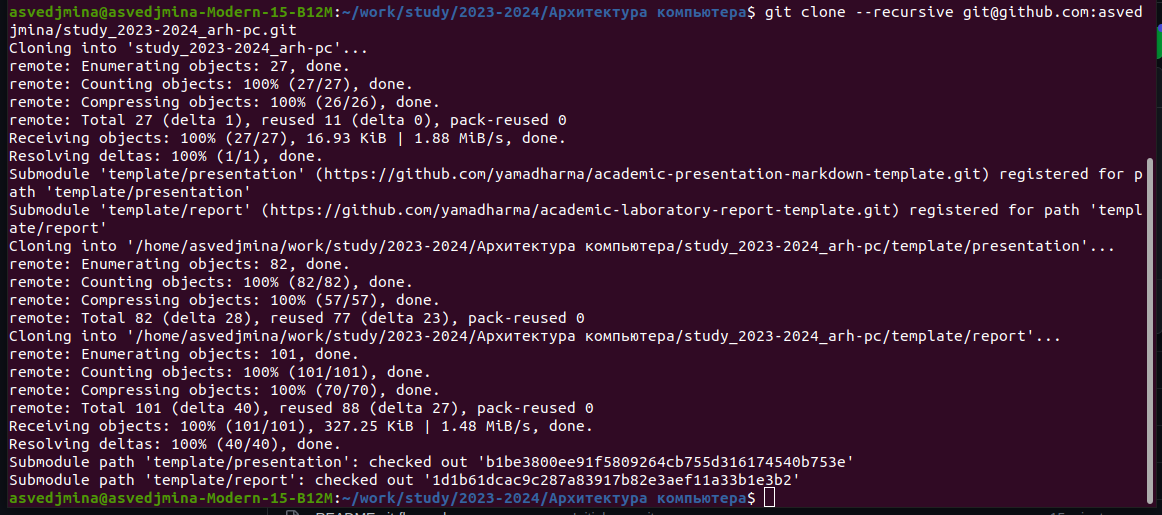
В терминале и создаю каталог для предмета «Архитектура компьютера» согласно указанному в инструкции шаблону.



создание каталога курса

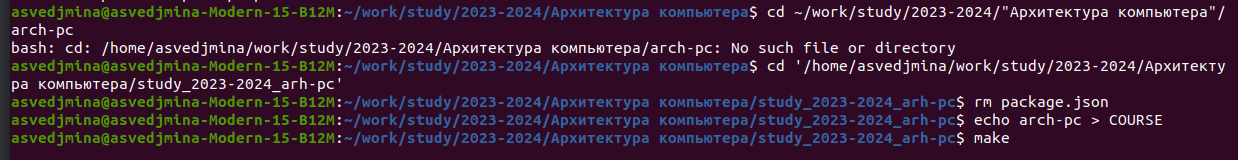
Для создания репозитория использую web-интерфейс github. Перехожу на страницу репозитория с шаблоном курса https://github.com/yamadharma/cour se-directory-student-template и выбираю “use this template”, называю репозиторий study\_2023-2024\_arh-pc.

В терминале перехожу в каталог курса и клонирую созданный репозиторий.



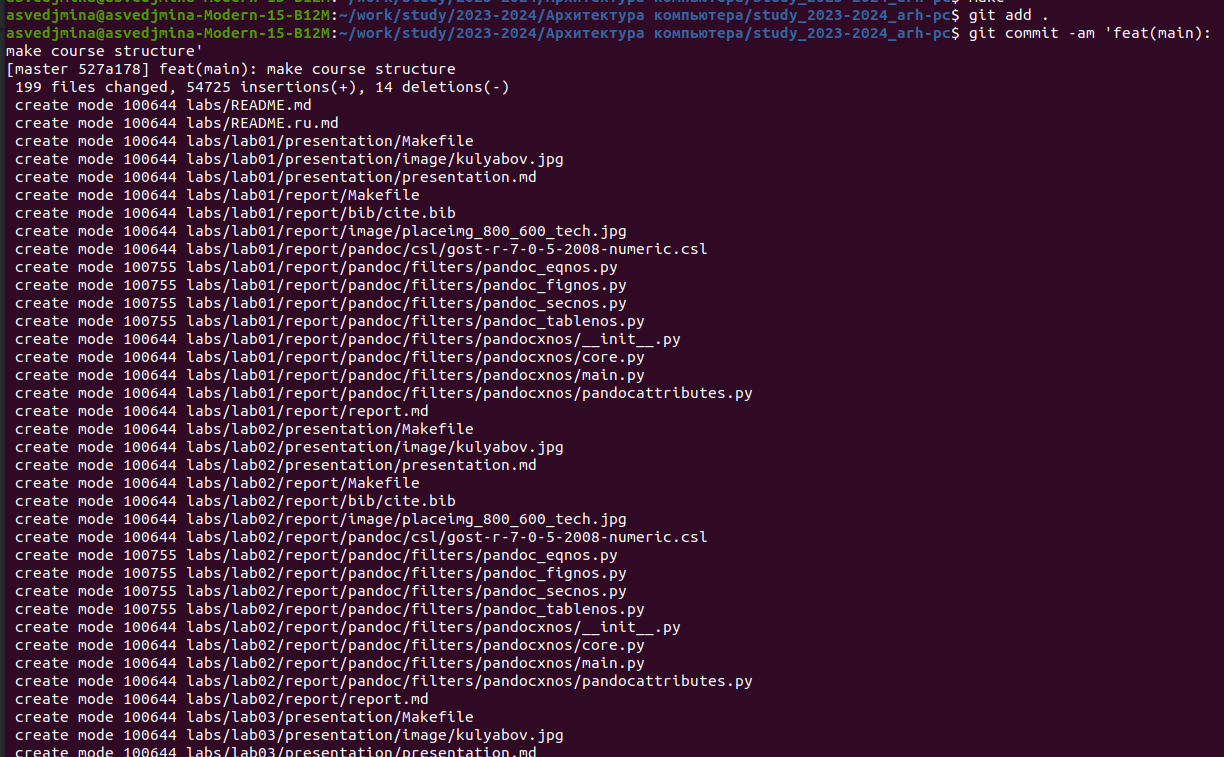
клонирование репозитория

Перехожу в каталог курса, удаляю лишние файлы и создаю необходимые каталоги.



удаление файлов и создание каталогов

Затем отправляю файлы на сервер.



отправка файлов на сервер

# 5 Задания для самостоятельной работы

1. Создайте отчет по выполнению лабораторной работы в соответствующем каталоге рабочего пространства (labs>lab02>report).
2. Скопируйте отчеты по выполнению предыдущих лабораторных работ в соответствующие каталоги созданного рабочего пространства.

Выполнение:

Перемещаю отчёт по первой лабораторной работе в ~/lab01/report.

перемещение отчёта

перемещение отчёта

Далее загружаю файлы на гитхаб.

# 6 Выводы

В ходе лабораторной работы я изучали, как работают средства контроля версий, система git.