Отчёт по лабораторной работе №3

Дисциплина: Архитектура компьютера

Буриева Шахзода Акмаловна

Содержание

# 1 Цель работы

Целью работы является изучить идеологию и применение средств контроля версий. Приобрести практические навыки по работе с системой git.

# 2 Теоретическое введение

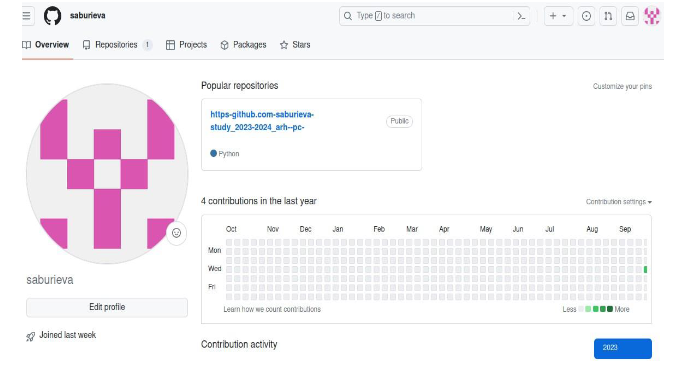
Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен доступ для участников про- екта. При внесении изменений в содержание проекта система контроля 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, пред- полагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Участник про- екта (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 хранили- ща 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 кон- фликтов, 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 функ- циональные возможности. Например, они могут поддерживать работу с несколькими вер- сиями 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.

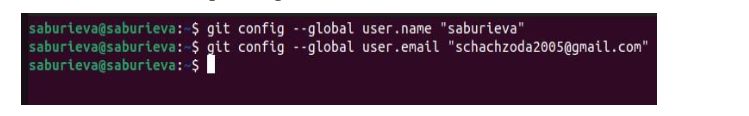
# 3 Выполнение лабораторной работы

Я создала учётную запись на github и заполнила основные данные.



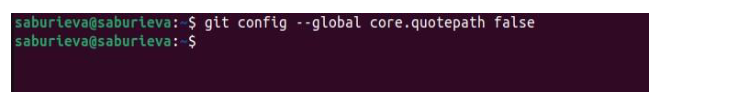
Создание учетной записи

Я указала имя и e-mail владельца репозитория.



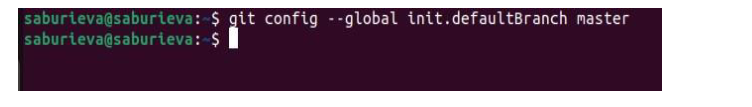
Указание имени и t-mail

Я настроила uf-8 в выводе сообщений git.



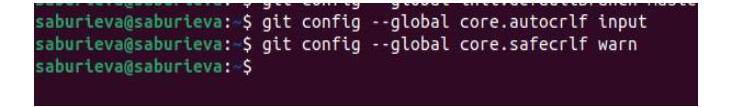
Настройка uf-8

Задала имя «master» для начальной ветки.



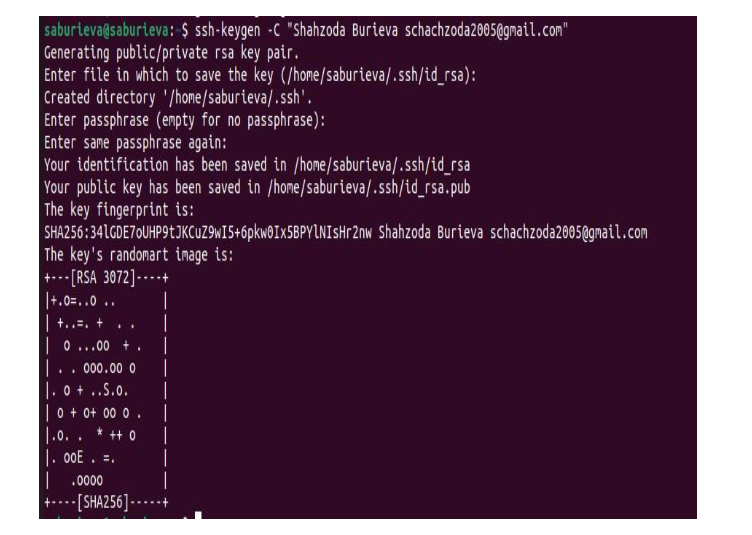
Имя master

Настроила параметры autocrlf and safecrlf.



Параметры autocrlf и safecrlf

Сгенерировала пару ключей(открытый и приватный).



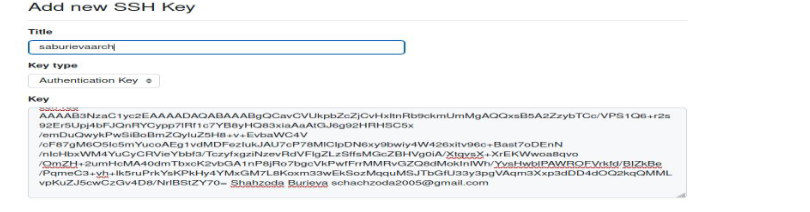
Гененрация ключей

Скопировала из локальной консоли ключ в буфер обмена



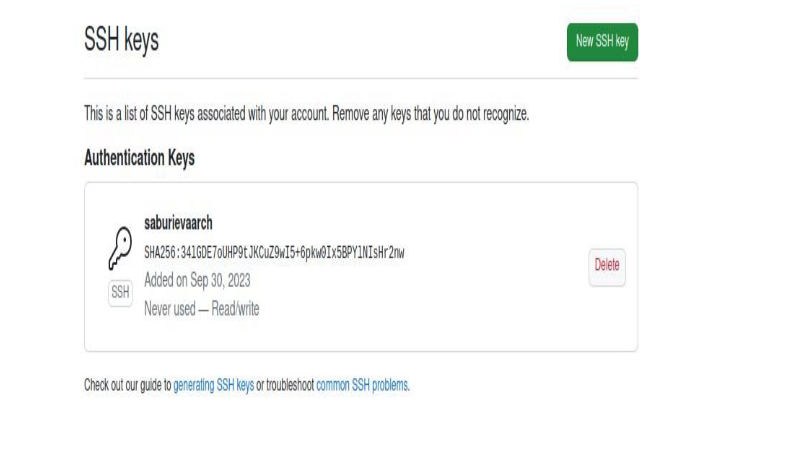
Копировка в буфер обмена

Вставила ключ в появившееся на сайте поле и указала для ключа название saburievaarch.



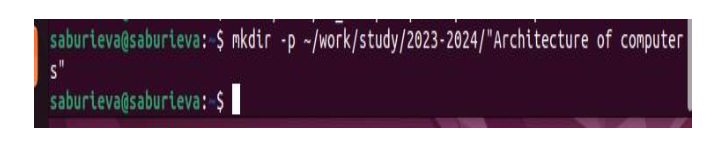
Указание имени ключа

Я создала SSH ключ.



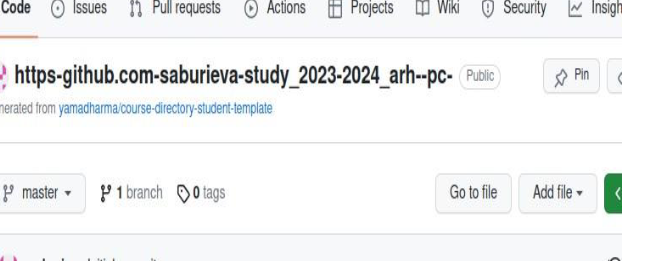
Создание SSH ключа

Создала каталог для предмета «Архитектура компьютера».



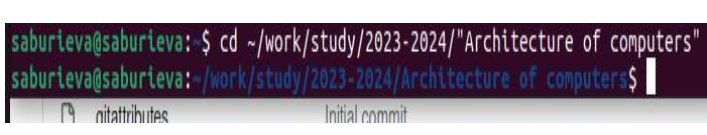
Создание каталога для предмета

Создала репозиторий курса на основе шаблона через web- интерфейс github.



Создание репозитория на github

Я перешла в каталог курса.



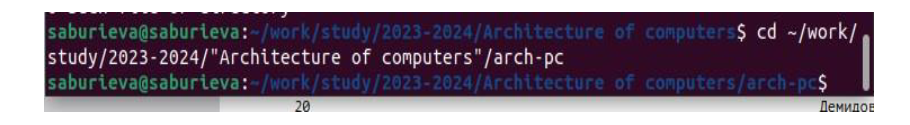
Переход в каталог курса

Клонировала созданный репозиторий.



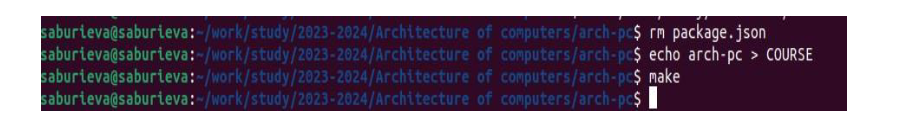
Клонирование репозитория

Перешла в каталог курса.



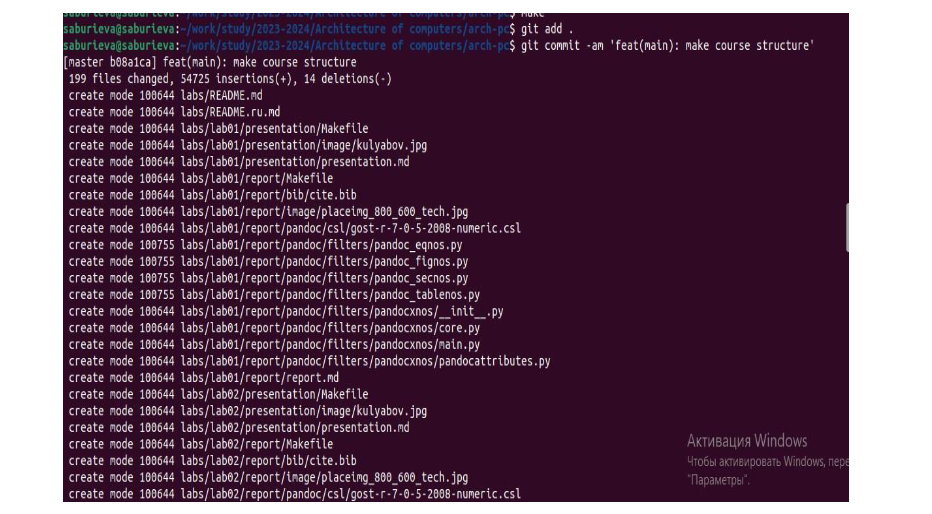
Переход в каталог курса

Удалила лишние файлы и создала необходимые каталоги.



Удаление файлов и создание каталогов

Ввела команды git add . and git commit -am.



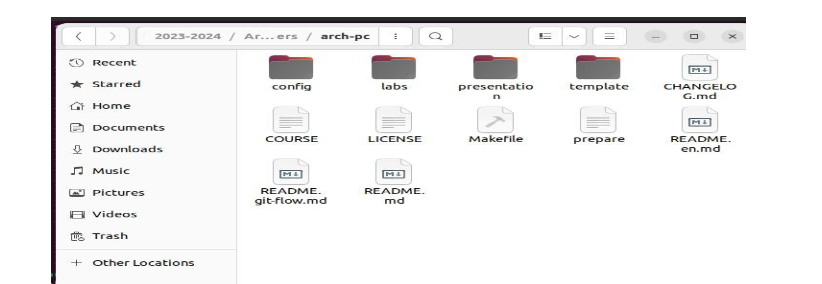
Команды git add . и got commit -am

Ввела команду git push и отправила файлы на сервер.



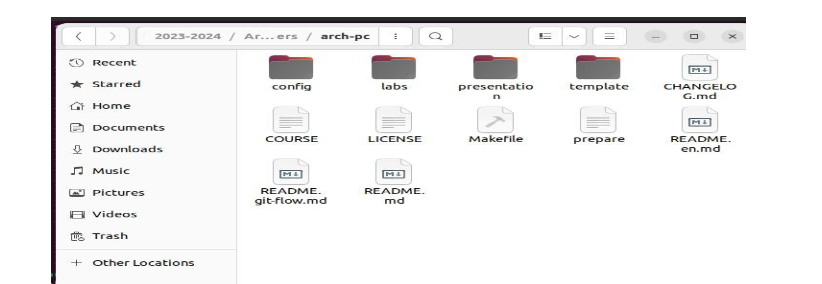
Команда git push и отправка файлов на сервер

Проверила правильность создания иерархии рабочего пространства в локальном репозитории.



Проверка на терминале

Проверила правильность создания иерархии рабочего пространства на сайте github.



Проверка на github

# 4 Выводы

Идеология и применение средств контроля изучены. После базовой настройки git создали иерархию рабочего пространства в локальном репозитории на странице github.

# Список литературы