Лабораторная работа №2. Система контроля версий Git

Павленко Сергей

Содержание

# 1 Цель работы

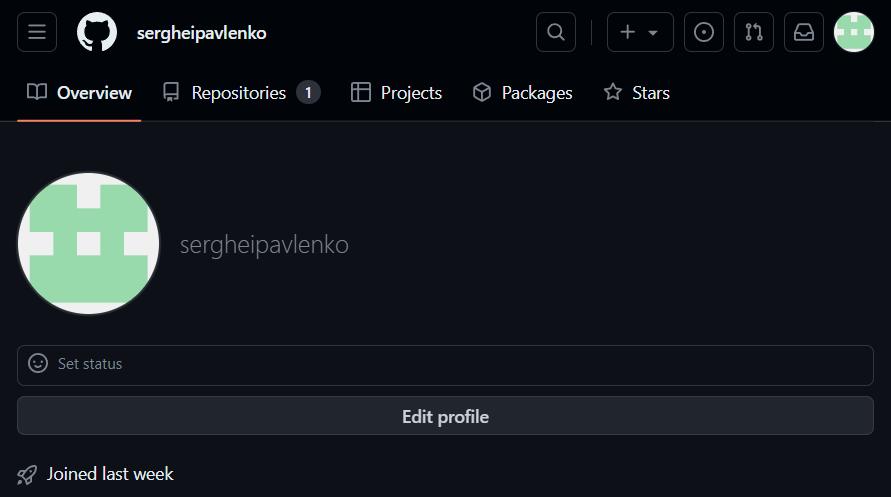
Целью работы является изучить идеологию и применение средств контроля версий. Приобрести практические навыки по работе с системой git.

# 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 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. Демидова А. В. 14 Архитектура ЭВМ 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.

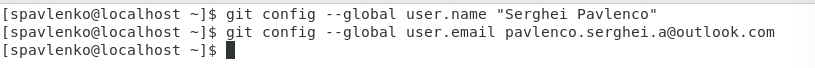
# 3 Выполнение лабораторной работы

1. Создайте учётную запись на github.com



1

2. Базовая настройка git

Сделаем предварительную конфигурацию git. 

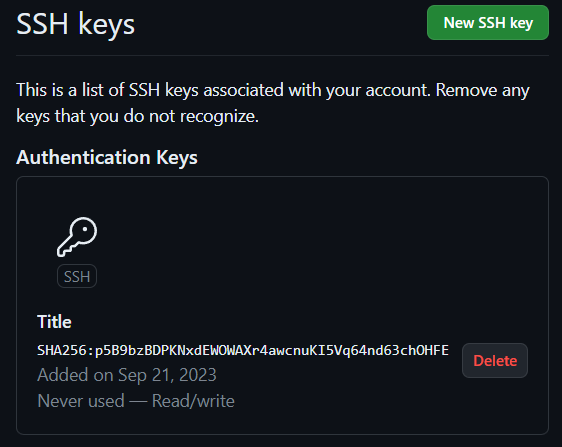
Настроим utf-8 в выводе сообщений git: 3

Зададим имя начальной ветки (будем называть её master): 4

Параметр autocrlf: 5

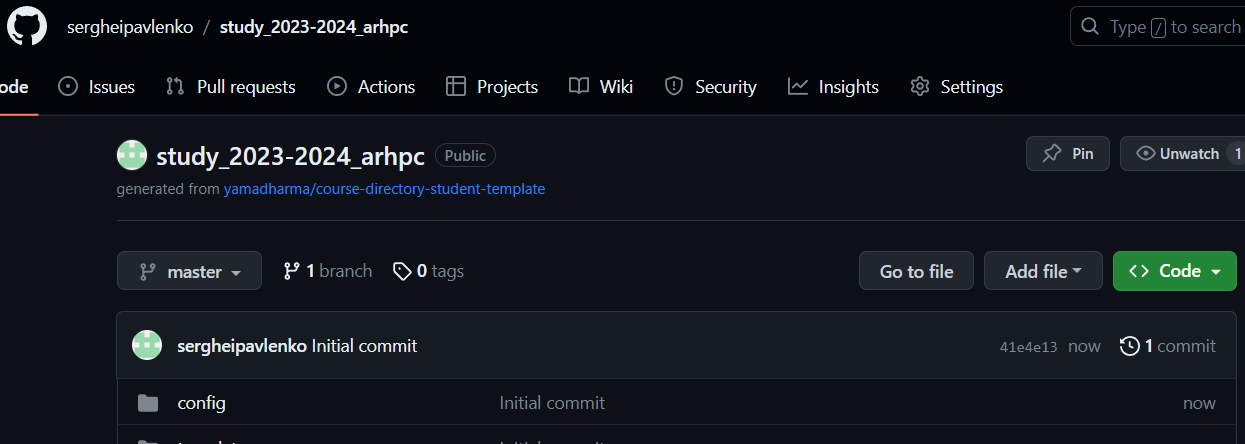
Параметр safecrlf: 6

Создание SSH ключа Для последующей идентификации пользователя на сервере репозиториев необходимо сгенерировать пару ключей (приватный и открытый): 

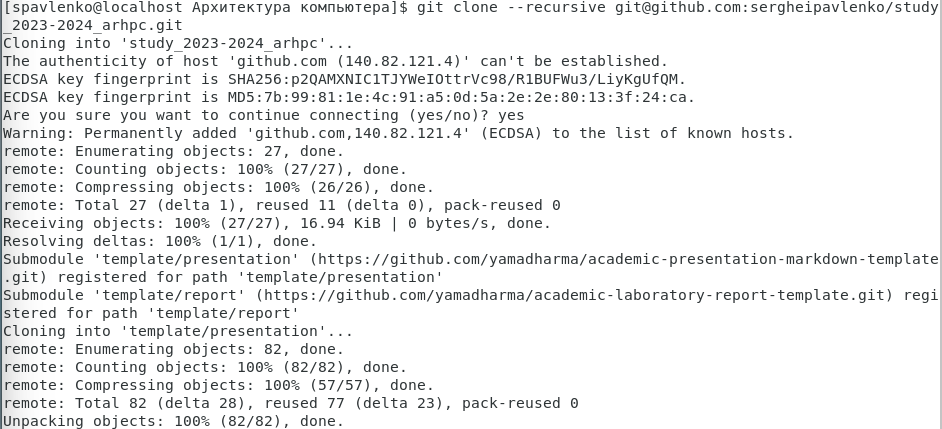
Создали SSH ключ 

Сознание рабочего пространства и репозитория курса на основе шаблона Откройте терминал и создайте каталог для предмета «Архитектура компьютера»: 9

Создали каталог для предмета «Архитектура компьютера»

Сознание репозитория курса на основе шаблона Репозиторий на основе шаблона можно создать через web-интерфейс github. Перейдём на станицу репозитория с шаблоном курса https://github.com/yamadharma/cour se-directory-student-template. Далее выберите Use this template. 

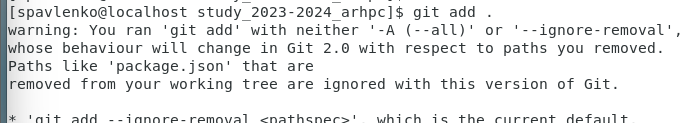
Создали репозиторий по шаблону В открывшемся окне задали имя репозитория study\_2023–2024\_arhpc и создали репозиторий (from template). Откроем терминал и перейдём в каталог курса: 11

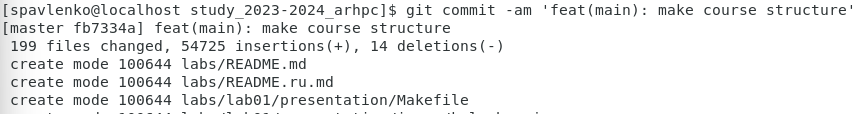
Клонируем созданный репозиторий: 

Настройка каталога курса Перейдём в каталог курса: 13

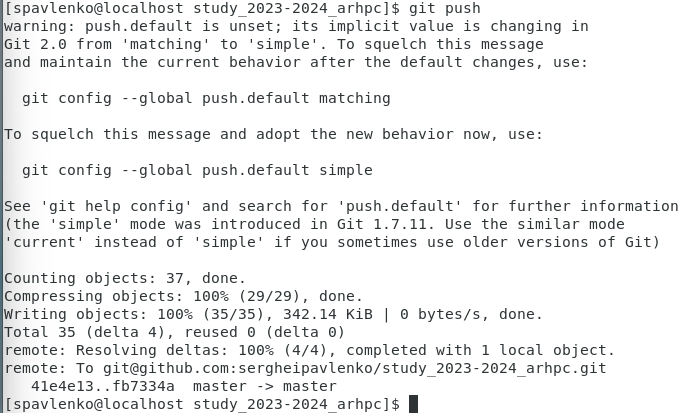
Удалим лишние файлы: 14

Создадим необходимые каталоги: 15

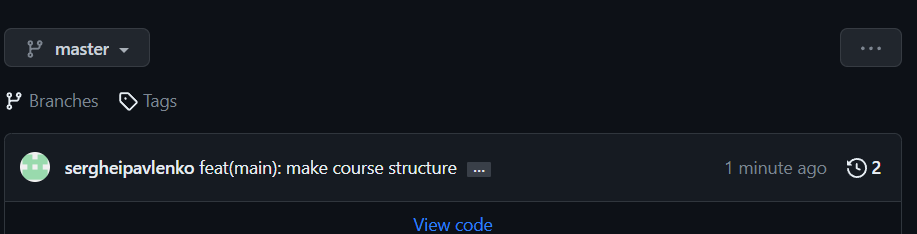
Отправим файлы на сервер: 



17



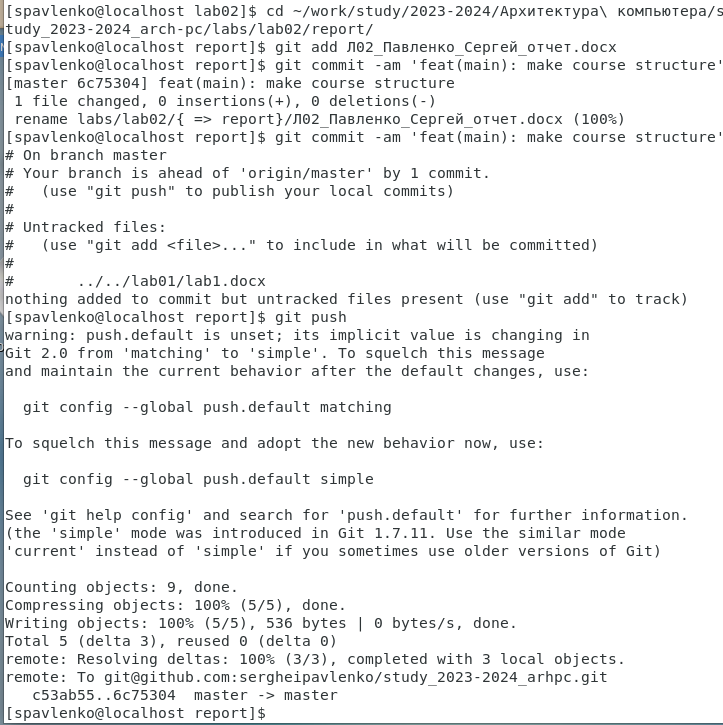
18



19

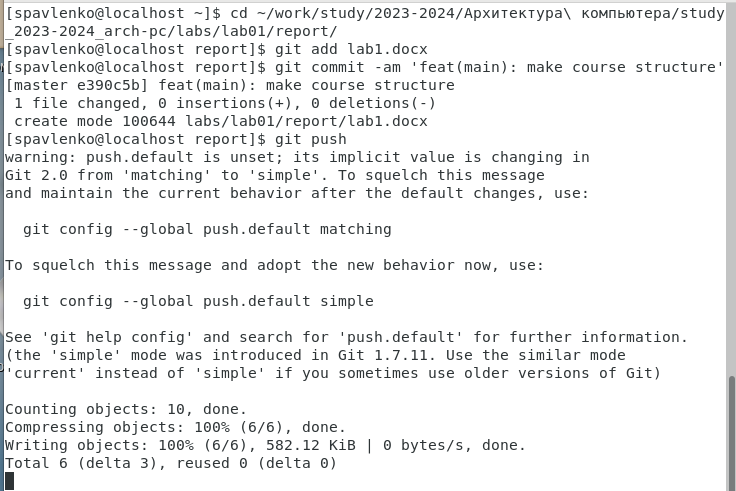
# 4 Задание для самостоятельной работы

1. Создайте отчет по выполнению лабораторной работы в соответствующем каталоге рабочего пространства (labs>lab02>report).



20

2. Скопируйте отчеты по выполнению предыдущих лабораторных работ в

соответствующие каталоги созданного рабочего пространства и загрузите файлы на github. 

Отчёты по лабораторным работам, находящийхся по адресу ~/labs/lab02/report и ~/labs/lab01/report, отправили в репозиторий и сохранили их в соответствующим им директориям

# 5 Выводы

В самостоятельной работе мы выполнили работу по сохранению отчетов в репозиторий github.com, с помощью соответствующих команд в Terminal

# Список литературы