Отчёт по лабораторной работе №2

Операционные системы

Трусова Алина Александровна

Содержание

Список иллюстраций

Список таблиц

# Цель работы

1. Изучить идеологию и применение средств контроля версий.
2. Освоить умения по работе с git.

# Задание

1. Установка программного обеспечения
2. Базовая настройка git
3. Создание ключей ssh
4. Создание ключей pgp
5. Добавление pgp ключа в github
6. Настройка автоматических подписей коммитов git
7. Настройка gh
8. Создание шаблона для рабочего пространства
9. Настройка каталога курса
10. Контрольные вопросы

# 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.

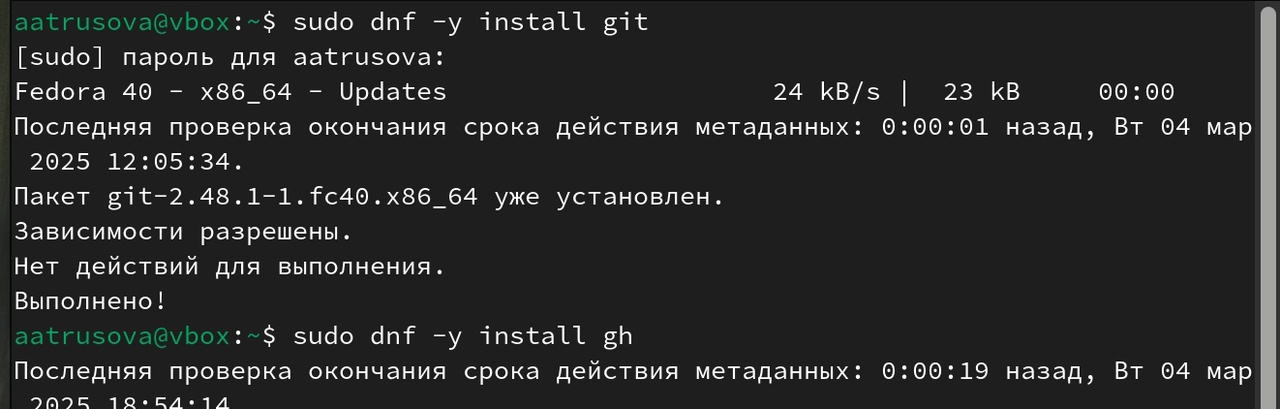
##Примеры использования git

1. Система контроля версий Git представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды git с различными опциями.
2. Благодаря тому, что Git является распределённой системой контроля версий, резервную копию локального хранилища можно сделать простым копированием или архивацией.

# Выполнение лабораторной работы

##1. Установка программного обеспечения

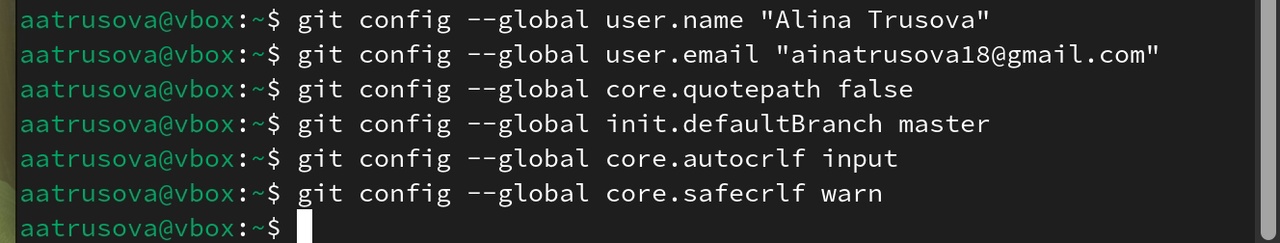
Установила git и gh (рис. [-@fig:001]).



Установка git и gh

##2. Базовая настройка git

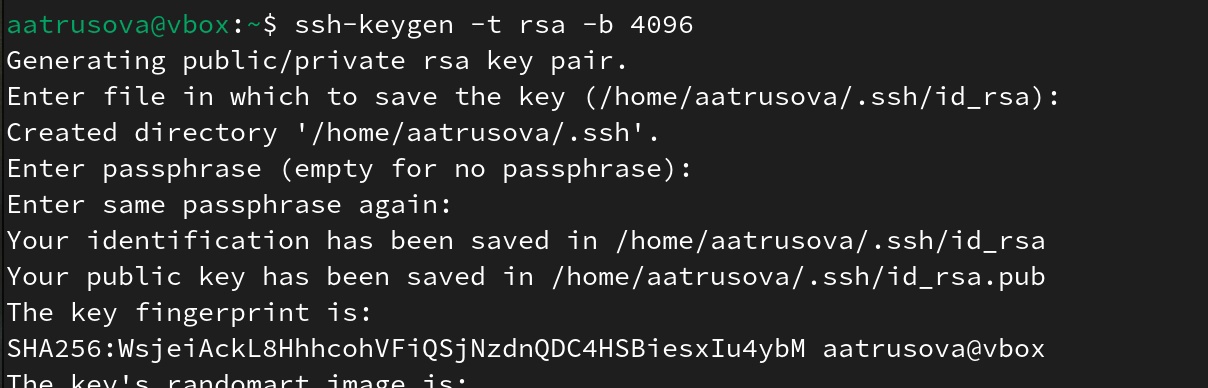
Задала имя и email владельца репозитория, настроила utf-8 в выводе сообщений git, задала имя начальной ветки и назвала её master, добавила параметры autocrlf и safecrlf (рис. [-@fig:002]).



Настройка git

##3. Создание ключа ssh

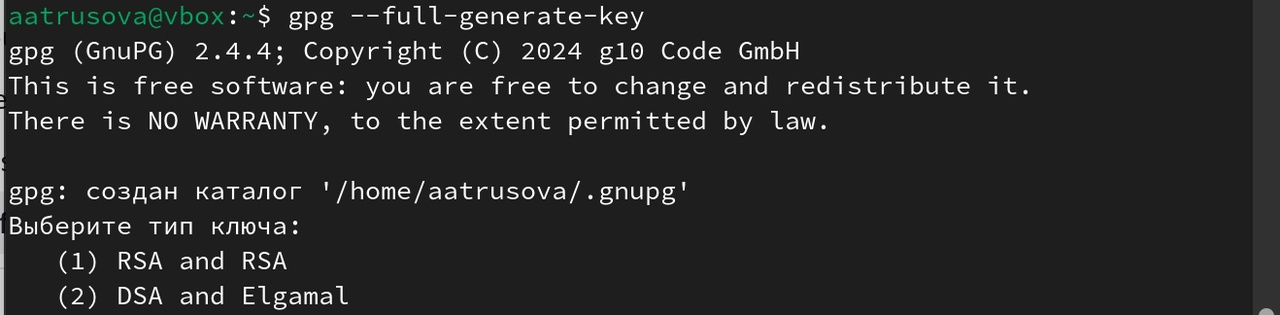
Создала ключ по алгоритму rsa с ключом размером 4096 бит (рис. [-@fig:003]).



Ключ rsa

##4. Создание ключа pgp

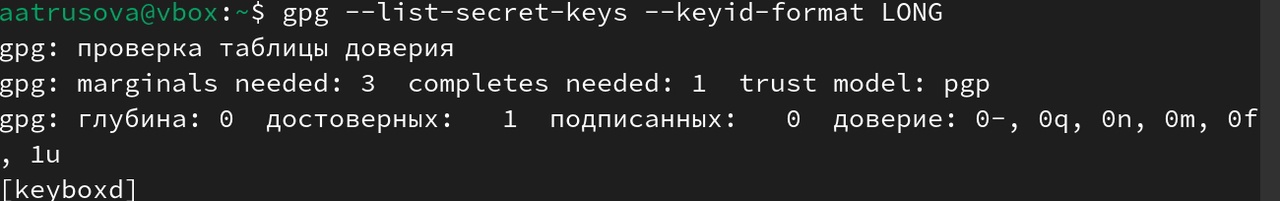
Создала ключ с нужными параметрами (рис. [-@fig:004]).



Ключ pgp

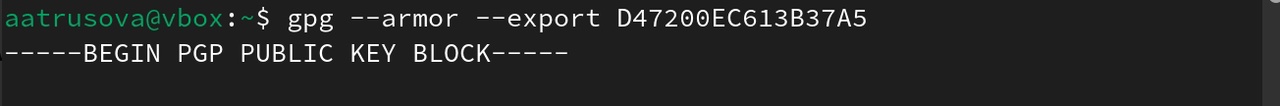
##5. Добавление pgp ключа в github

Вывела список ключей и скопировала отпечаток приватного ключа (рис. [-@fig:005]).



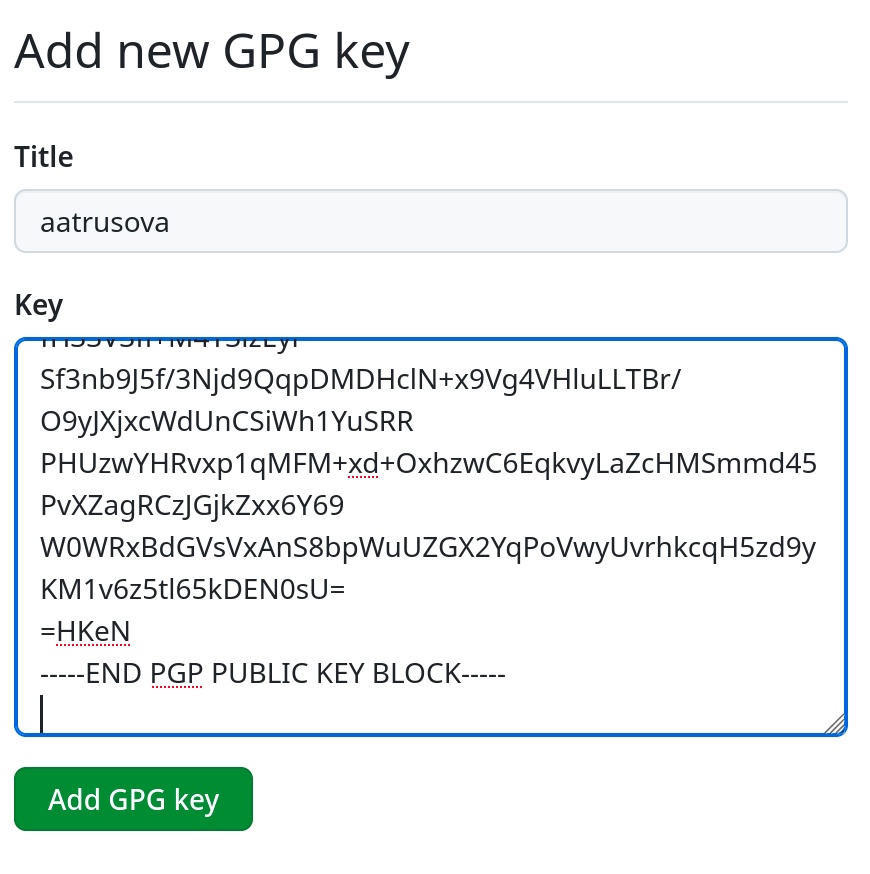
Ключ pgp

Вывела сгенерированный ключ и скопировала его в буфер обмена (рис. [-@fig:006]).



Ключ pgp

Добавила новый ключ gpg на github (рис. [-@fig:007]).



Добавление ключа на github

##6. Настройка автоматических подписей коммитов git

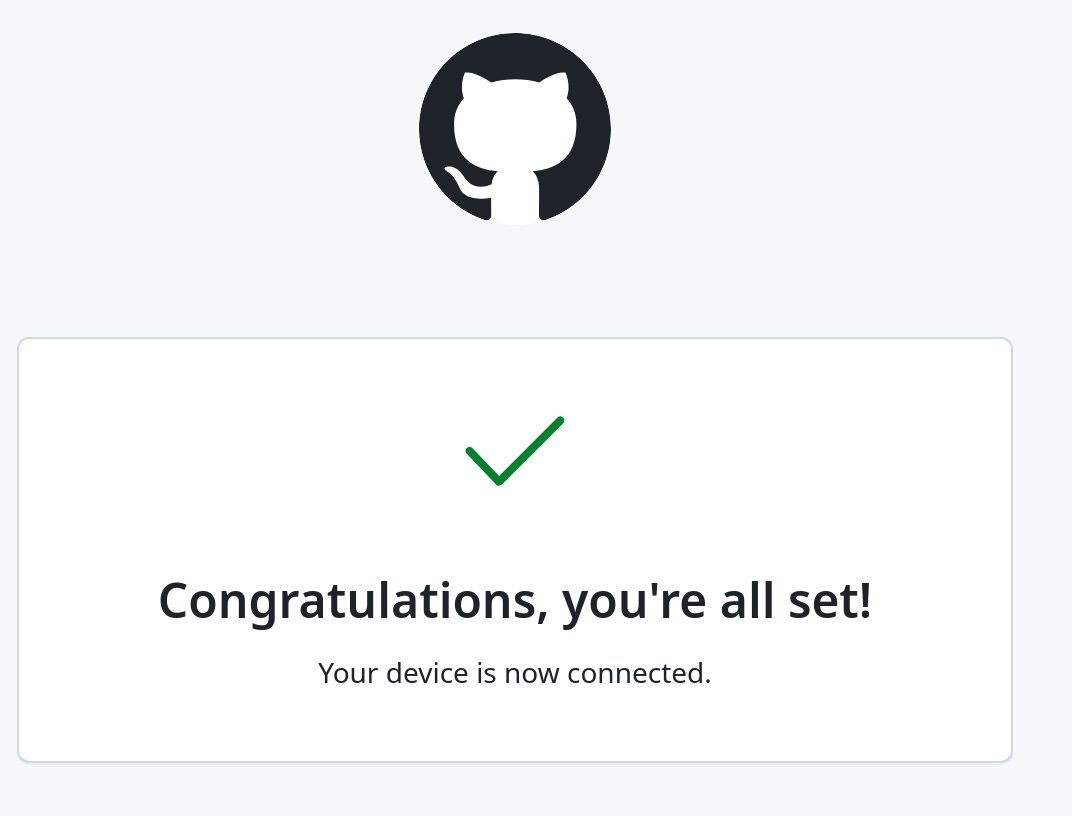
Указала git применять введённый email для подписи коммитов (рис. [-@fig:008]).



Настройка подписей коммитов git

##7. Настройка gh

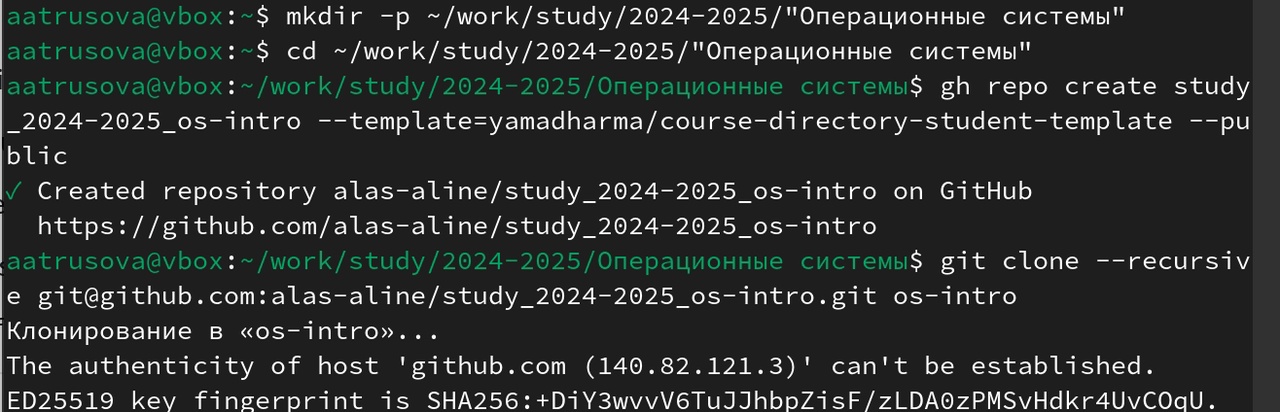
Авторизовалась на гитхабе (рис. [-@fig:009]).



Авторизация

##8. Создание шаблона рабочего пространства

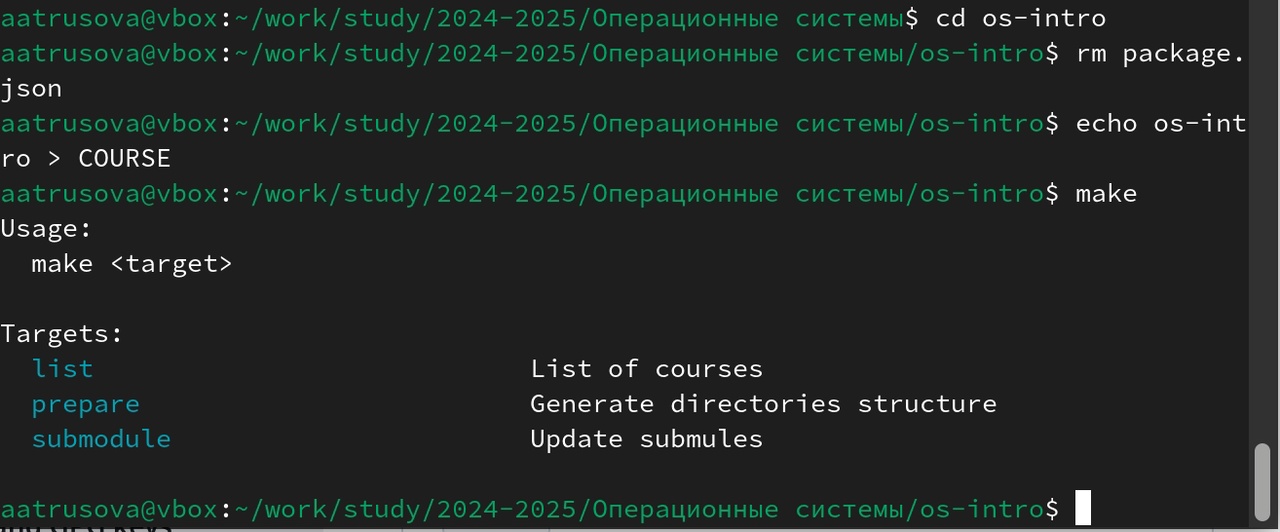
Создала рабочее пространство по шаблону (рис. [-@fig:010]).



Шаблон репозитория

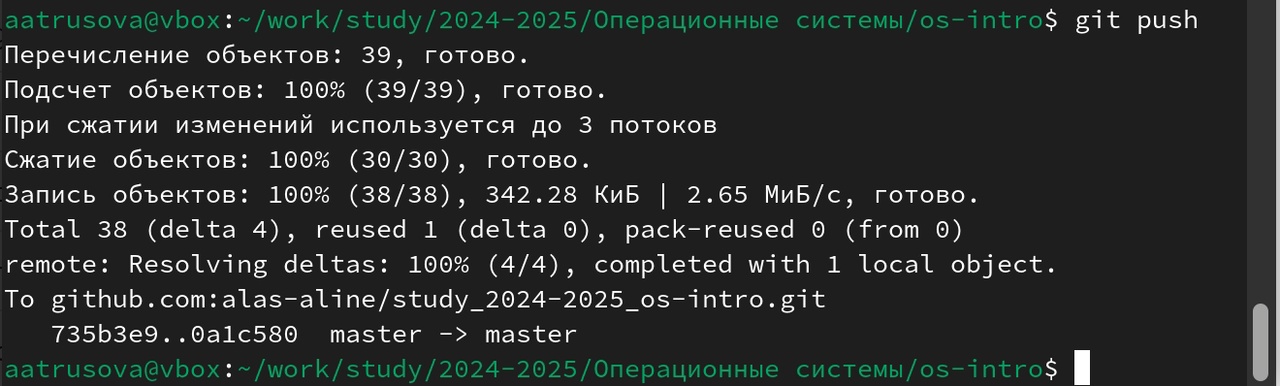
##9. Настройка каталога курса

Перешла в каталог курса, удалила лишние файлы и создала новые (рис. [-@fig:011]).



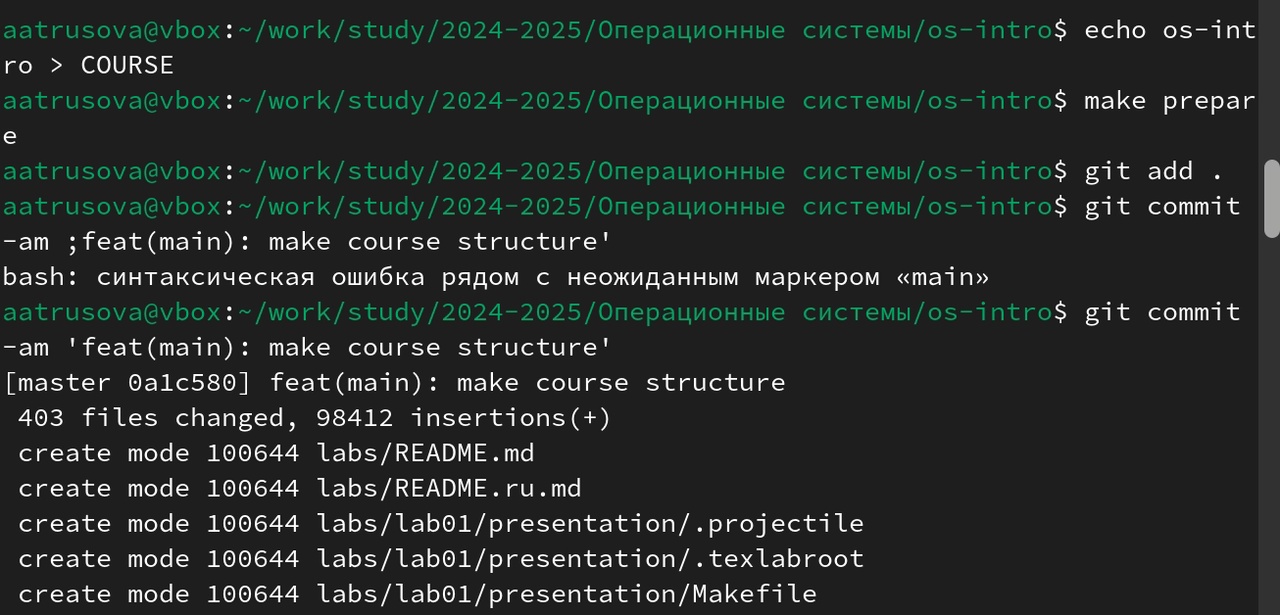
Настройка каталога курса

Прописала коммит (рис. [-@fig:012]).



Коммит

Отправила изменения на репозиторий (рис. [-@fig:013]).



Коммит

# Выводы

Я изучила идеологию и применение средств контроля версий и освоила умения по работе с git.

# Список литературы

1. Архитектура компьютеров и операционные системы. Раздел “Операционные системы”. “Лабораторная работа №2” https://esystem.rudn.ru/mod/page/view.php?id=1224371