Лабораторная работа №1

Введение в git и markdown

Лебедева Ольга Андреевна

Содержание

# Цель работы

1. Изучение основных команд Git для управления версиями кода, включая клонирование репозиториев, фиксацию изменений и совместную работу с другими разработчиками.
2. Освоение основных команд и концепций операционной системы Linux, таких как навигация по файловой системе, управление файлами и папками, а также работа с правами доступа.
3. Освоение синтаксиса и возможностей языка разметки Markdown для создания структурированных документов с форматированным текстом, изображениями и другими элементами.

# Теоретическое введение

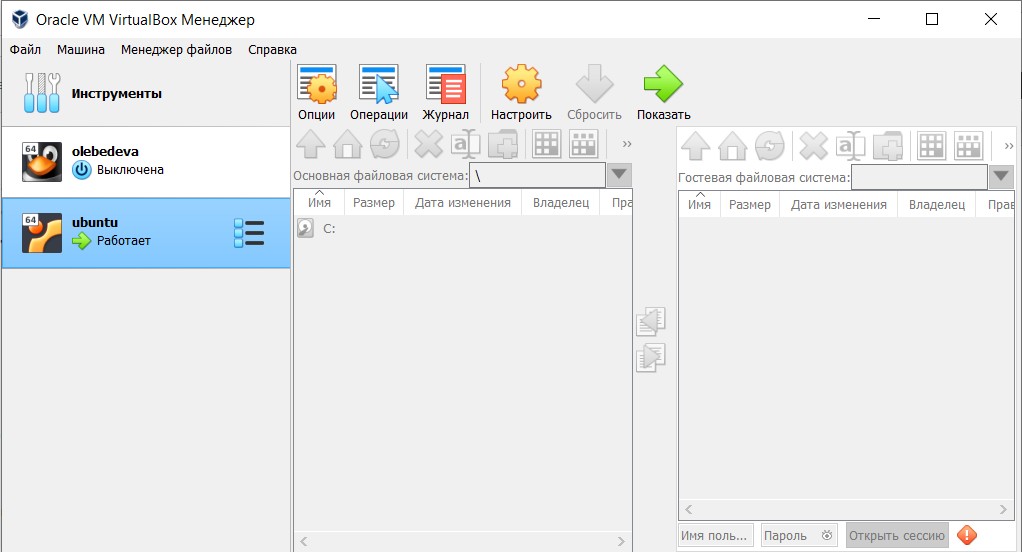
Git - это система контроля версий, разработанная Линусом Торвальдсом. Она предназначена для отслеживания изменений в исходном коде программного обеспечения и управления ими. Git позволяет разработчикам эффективно работать над проектами в команде, фиксировать изменения, создавать ветви разработки, сливать изменения и многое другое. Этот инструмент широко используется в индустрии разработки ПО и является стандартом для совместной работы над проектами.

Linux - это семейство операционных систем с открытым исходным кодом, основанных на ядре Linux. Linux предоставляет свободное программное обеспечение и позволяет пользователям полностью настраивать и настраивать свои операционные системы под свои нужды. Он широко используется в различных областях, включая серверы, персональные компьютеры, мобильные устройства и встроенные системы.

Markdown - это легкий язык разметки, который используется для форматирования текста. Он позволяет создавать структурированные документы с использованием простого и интуитивно понятного синтаксиса. Markdown широко применяется для написания README файлов в репозиториях Git, статей на блогах, форматирования сообщений в форумах и многое другое. Он позволяет быстро создавать читаемый и структурированный контент без необходимости в глубоком знании HTML или других языков разметки.

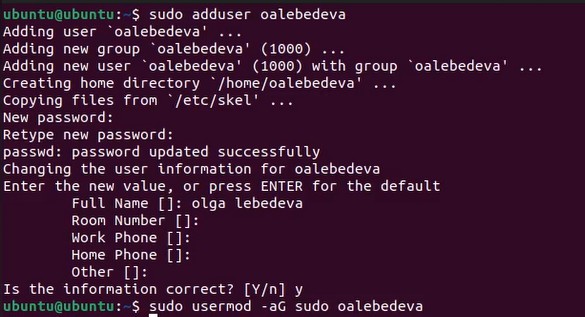
# Ход работы

Для работы будем использовать виртуальную машину с установленным дистрибутивом ubuntu. (рис. [-@fig:001])

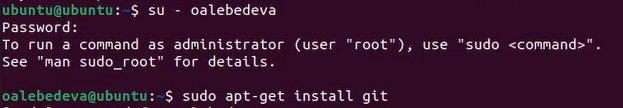


Запуск VB

При начальном запуске машины имя пользователя - ubuntu. Для соглашения об наименовании добавим нового пользователя и предоставим ему права администратора. Обозначим имя и пароль. (рис. [-@fig:002]) (рис. [-@fig:003])

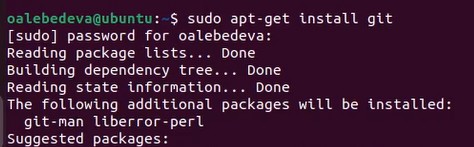


Соглашение об наименовании



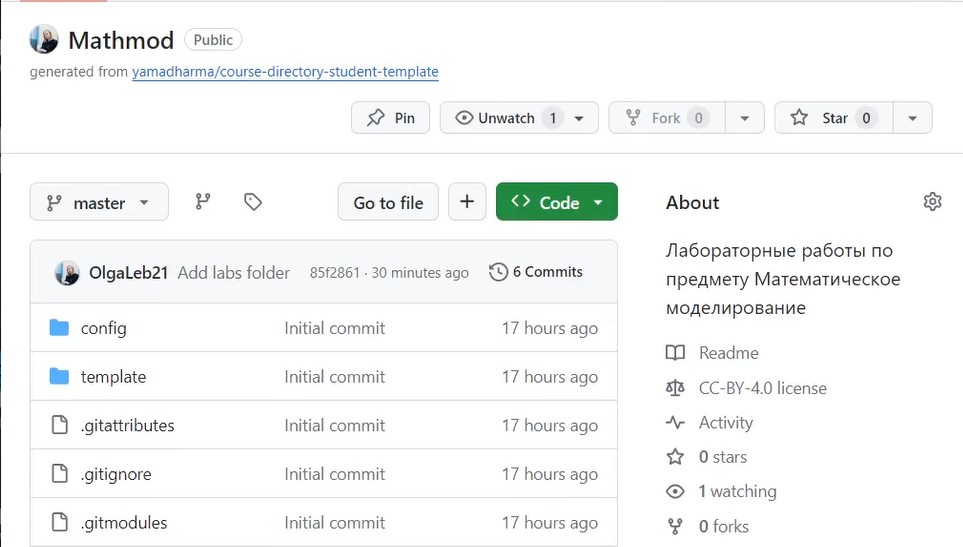
Соглашение об наименовании

Далее, устанавливаем Git на нашем компьютере Ubuntu, чтобы мы могли использовать его для управления версиями. (рис. [-@fig:004])



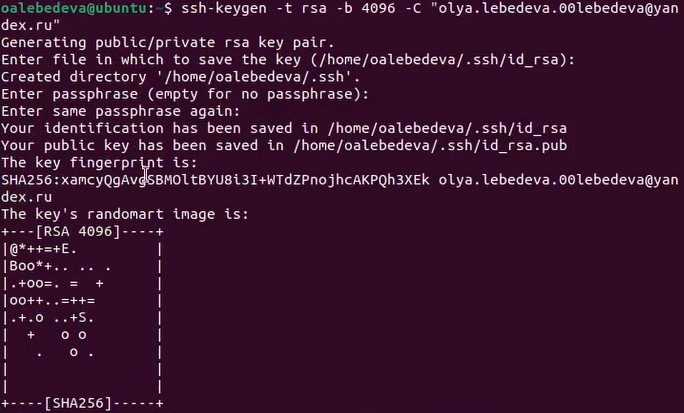
Установка Git

Клонируем репозиторий, предложенный в качестве шаблона для хранения и создания лабораторных работ и отчётов. (рис. [-@fig:005])



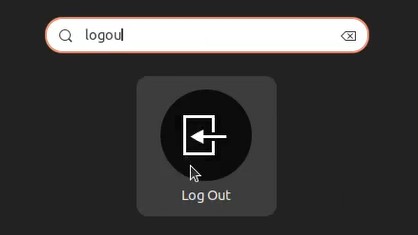
Клонирование репозитория

Создаём через консоль ssh-ключ, чтобы связать локальную папку и репозиторий на Гитхабе. (рис. [-@fig:006])

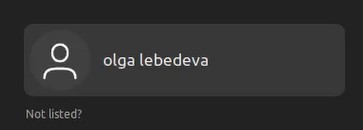


Создание ssh-ключа

Мы не учли, что необходимо было перезайти под именем нового пользователя, поэтому делаем это с помощью logout. (рис. [-@fig:007]) (рис. [-@fig:008])

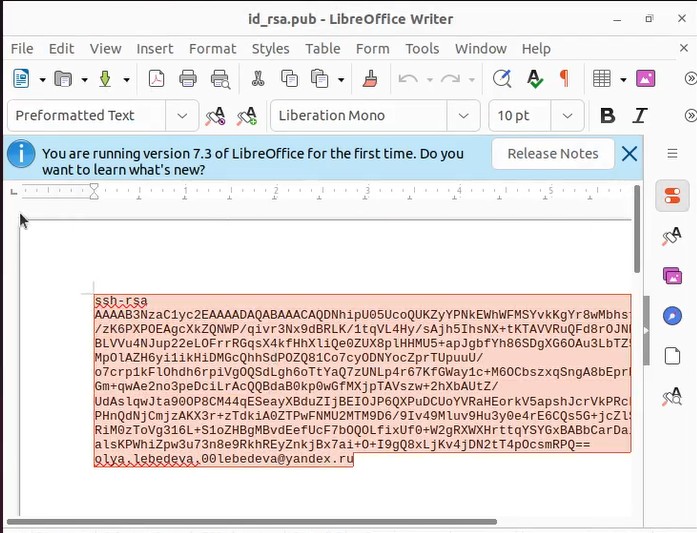


Повторная авторизация”



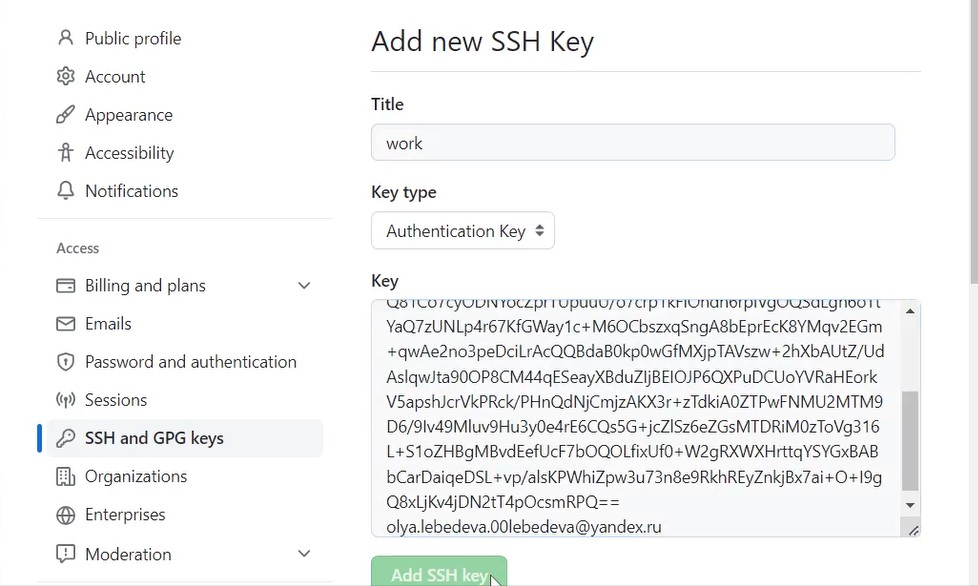
Вход в аккаунт

Открываем скрытые файлы и находим папку .ssh, где был сгенерирован ключ.(рис. [-@fig:009])



Запись ключа ssh

Заходим на Гитхаб в раздел настройки и создаём новый ключ, чтобы связать наш репозиторий и локальную папку. (рис. [-@fig:010])



Данные об ssh-ключе

Следующие команды в Git используются для настройки имени пользователя и адреса электронной почты, которые будут ассоциированы с коммитами. Первая команда устанавливает имя пользователя, а вторая - наш адрес электронной почты. Это помогает идентифицировать нас как автора коммитов в истории Git. (рис. [-@fig:011])

Идентификация

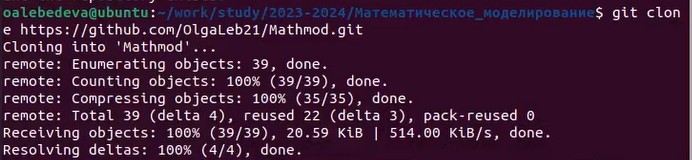
Идентификация

Создаём рабочее пространство по следующей иерархии.(рис. [-@fig:012])



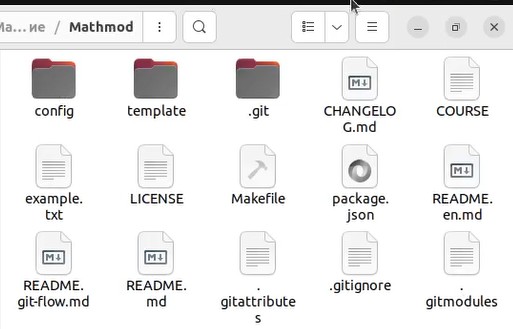
Создание каталогов

Переходим в каталог Математическое\_моделирование и клонируем репозиторий, который мы уже разместили на своём Гитхабе. (рис. [-@fig:013])



Клонирование репозитория

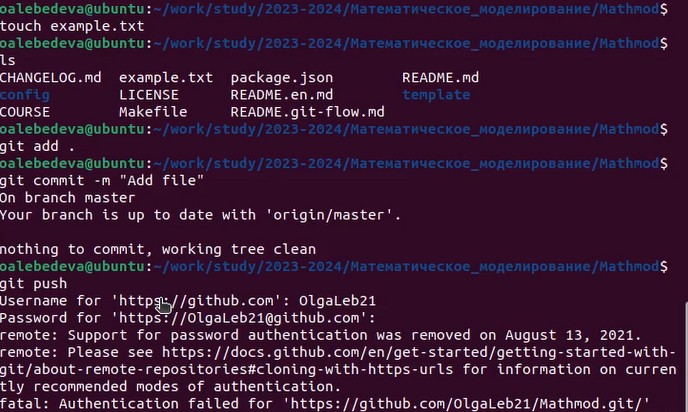
Переходим на компьютере в папку Mathmod и видим, что предложенный репозиторий был удачно клонирован. (рис. [-@fig:014])



Клонированные папки и файлы

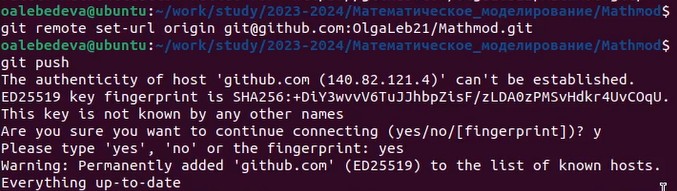
Для примера создадим через консоль текстовый файл с именем example.txt. Далее ввёдём три стандартных команды, которые используются для добавления изменений в репозиторий Git, фиксации изменений и отправки их на удаленный сервер: (рис. [-@fig:015])

git add .  
git commit -m "Message"  
git push



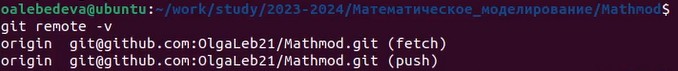
Отправка изменений в репозиторий

Однако возникает ошибка, так как мы используем URL-адрес клонирования HTTPS для репозитория. Чтобы исправить это, введём следующуй команду и повторим push, после чего все изменения в локальной папке уйдут в наш репозиторий. (рис. [-@fig:016])



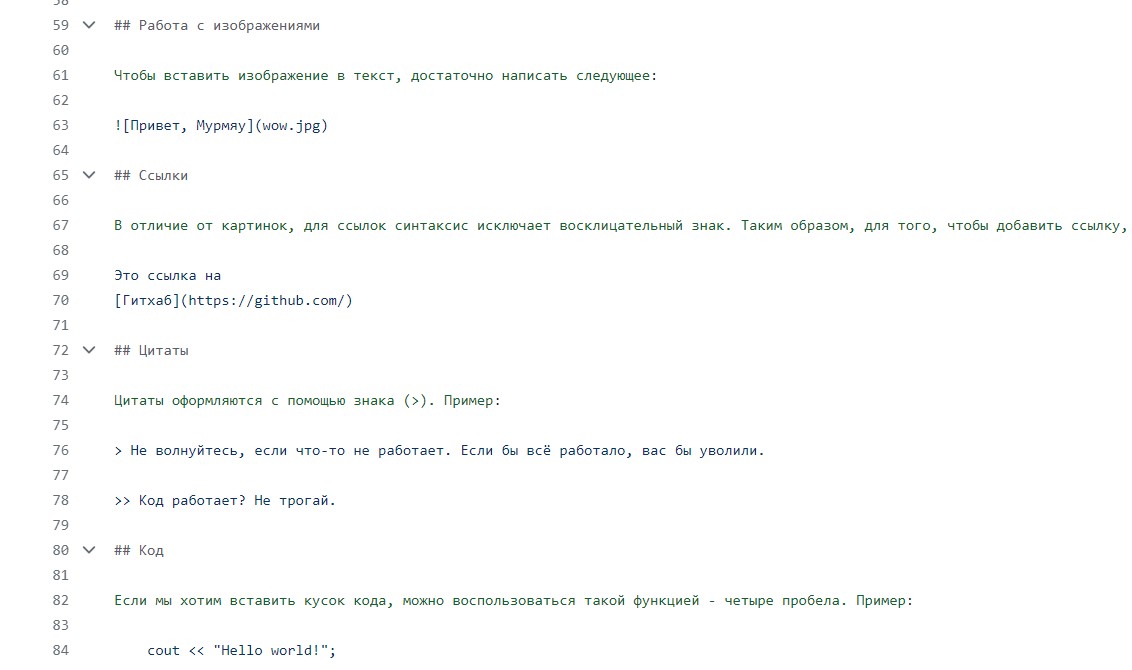
Исправление возникшей ошибки

Также мы в любой момент можем проверить, с какими репозиториями можем удалённо работать: (рис. [-@fig:017])

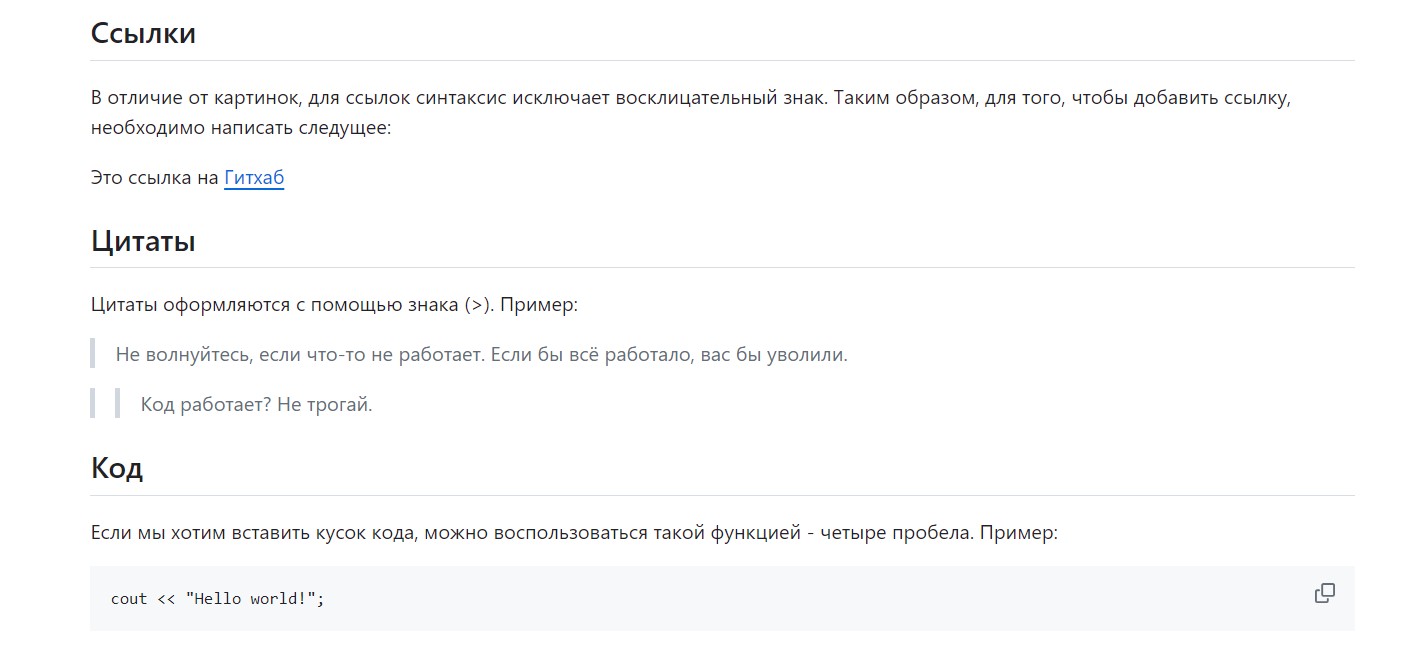


Список удаленных репозиториев

Во второй части работы вспомним markdown. Сделаем это на примере собственного файла, где указаны все необходимые особенности синтаксиса. (рис. [-@fig:018]) (рис. [-@fig:019])

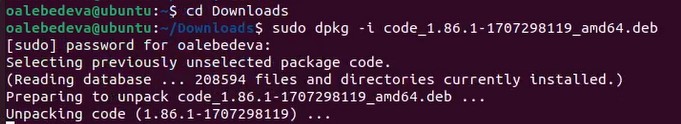


Синтаксис markdown



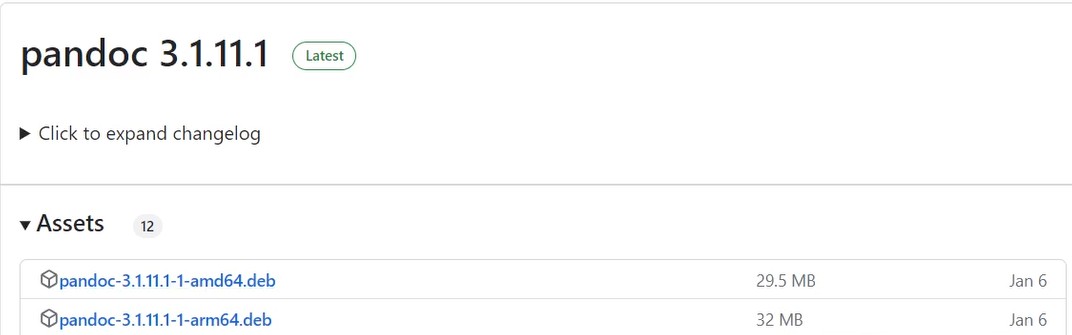
Пользовательский вид

Переходим к установке среды Visual Studio Code. Скачиваем файл с официального сайта и при помощи команды в консоли преобразуем в папку. (рис. [-@fig:020])



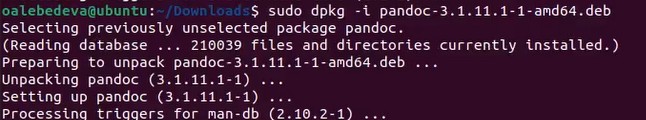
Установка Visual Studio Code

Для дальнейшей работы с файлам нам необходимо установить утилиту для работы с текстовыми форматами. Воспользуемся для этого утилитой pandoc. (рис. [-@fig:021])



Установка pandoc

Извлекаем из скаченного архива папки и файлы: (рис. [-@fig:022])



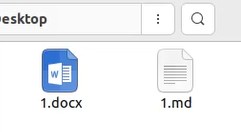
Установка pandoc

Для примера попробуем сконвертировать файл с расширением .md в другой текстовый формат. (рис. [-@fig:023])

Использование pandoc

Использование pandoc

В результате, получим новый файл с расширением .docx: (рис. [-@fig:024])



Конвертация файла

# Вывод

1. Изучили основные команды Git для управления версиями кода, включая клонирование репозиториев, фиксацию изменений и совместную работу с другими разработчиками.
2. Освоили основные команды и концепции операционной системы Linux, такие как навигация по файловой системе, управление файлами и папками, а также работа с правами доступа.
3. Освоили синтаксис и возможности языка разметки Markdown для создания структурированных документов с форматированным текстом, изображениями и другими элементами.