Отчёт по лабораторной работе 1

Настройка рабочего пространства. Система контроля версий Git. Язык разметки Markdown.

Чванова Ангелина Дмитриевна, НПИбд-02-21

Содержание

# Цель работы

Настроить рабочее пространство для лабораторной работы. Изучить идеологию и применение средств контроля версий, а также освоить умения по работе с git. Основить работу с Markdown.

# Задание

– Создать базовую конфигурацию для работы с git. – Создать локальный каталог для выполнения заданий по предмету. - Написание отчета и презентации с помощью Markdown. - Конвертация отчета и презентации во все необходимые форматы.

# Теоретическое введение

## Системы контроля версий. Общие понятия

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 дельтакомпрессию — сохранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных

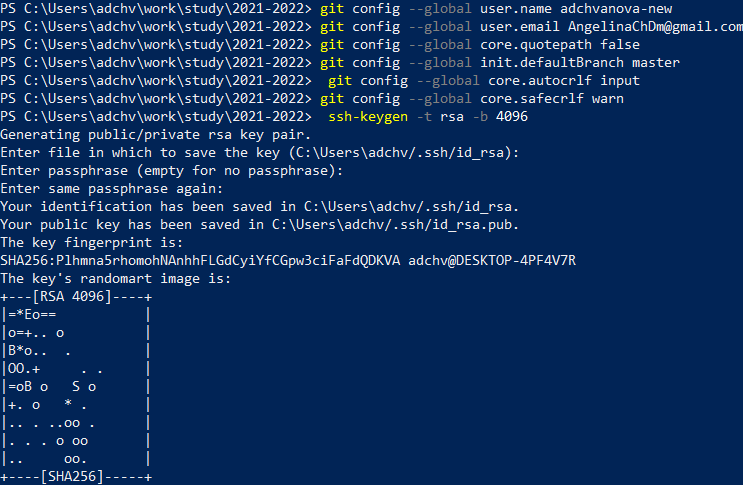
## GitHub

GitHub— крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки.Веб-сервис основан на системе контроля версий Git и разработан на Ruby on Rails[7] и Erlang компанией GitHub, Inc (ранее Logical Awesome). Сервис бесплатен для проектов с открытым исходным кодом и (с 2019 года) небольших частных проектов, предоставляя им все возможности (включая SSL), а для крупных корпоративных проектов предлагаются различные платные тарифные планы.

## Markdown Markdown — это облегченный язык разметки с синтаксисом форматирования обычного текста. Документация поддерживает разметку Markdown в соответствии с CommonMark и ее синтаксический анализ через подсистему Markdig. Документация также поддерживает пользовательские расширения Markdown, которые предоставляют более обширный контент на сайте документации.

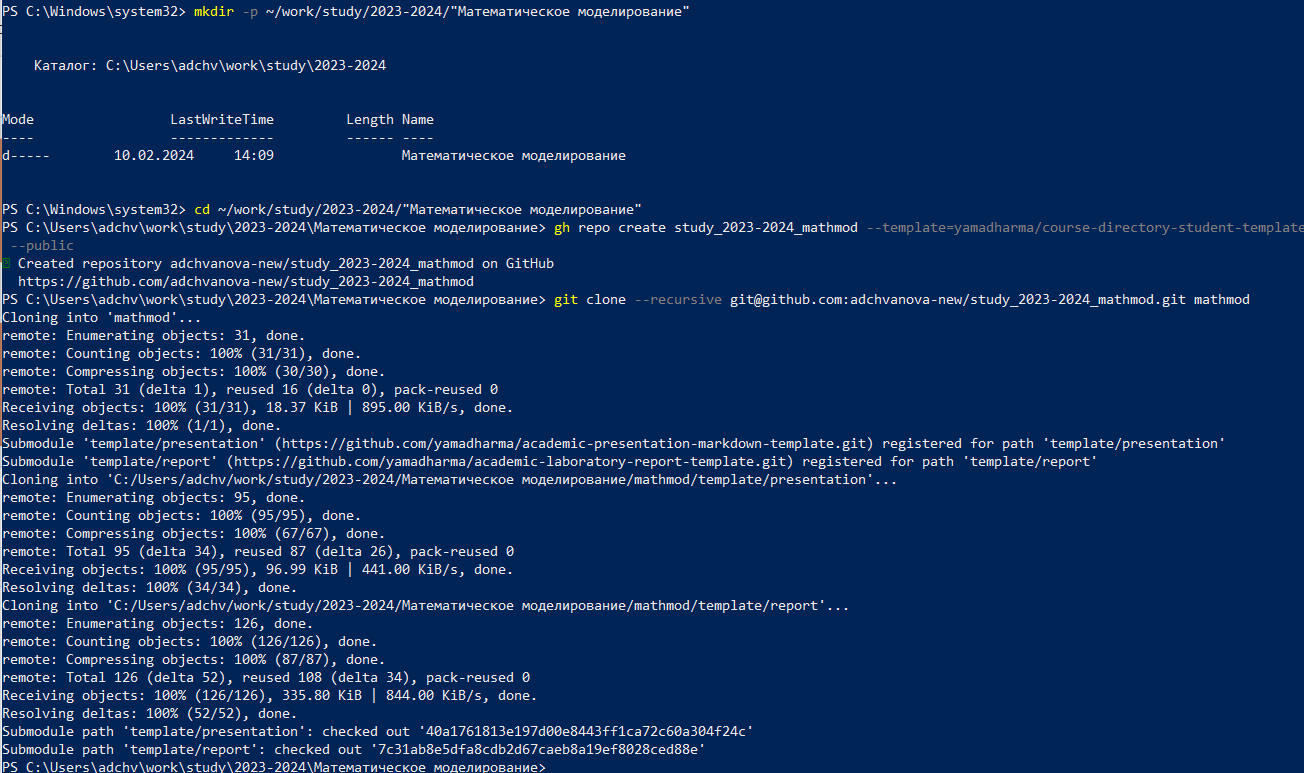
# Выполнение лабораторной работы

1. Подготовка GitHub. Установка имени и электронной почты. Получение ключа. Выполняем следующие команды, чтобы git узнал наше имя и электронную почту (рис. 1).



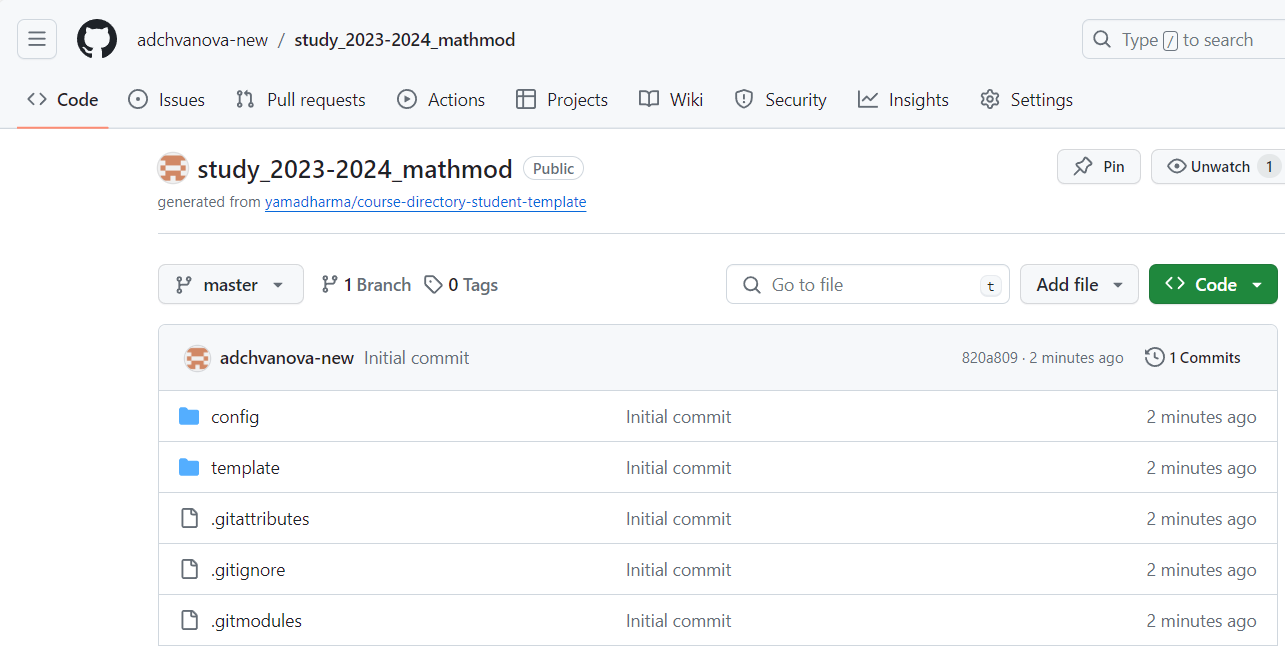
Установка имени и электронной почты. Получение ключа.

1. Создание рабочего пространства с помощью утилит: mkdir -p ~/work/study/2023-2024/“Математическое моделирование” cd ~/work/study/2023-2024/“Математическое моделирование” gh repo create study\_2023-2024\_mathmod –template=yamadharma/course-directory-student-template –public git clone –recursive git@github.com:adchvanova-new/study\_2023-2024\_mathmod.git mathmod (рис. 2).



Создание рабочего пространства с помощью утилит.

1. Проверка репозитория на GitHub Открываем гитхаб и видим, что репозиторий успешно клонирован (рис. 3).



Проверка репозитория на GitHub.

1. Настройка каталога курса Переход в каталог курса:

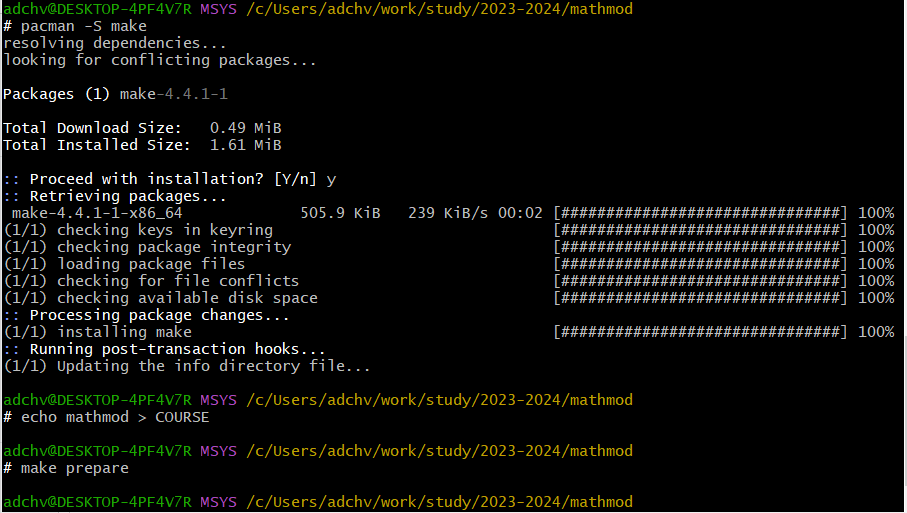
cd ~/work/study/2023-2024/“Математическое моделирование”/mathmod Удаление лишних файлов:

rm package.json (рис. 4).

Настройка каталога курса.

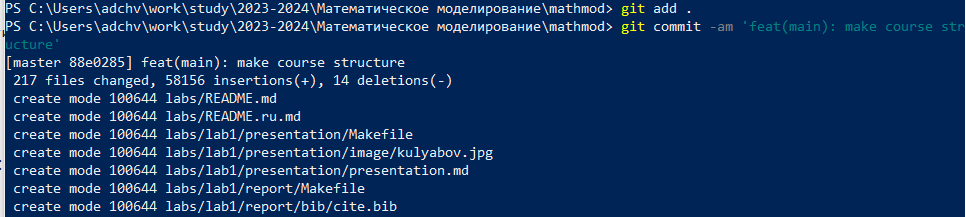
Настройка каталога курса.

Переход в msys2 и установка make. Затем мы прописываем команды для создания необходимых каталогов: echo mathmod > COURSE make prepare (рис. 5).

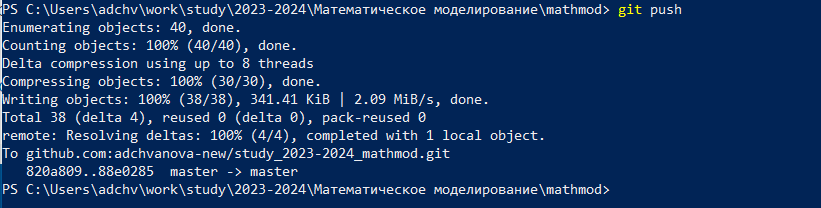


Работа в msys2.

1. Отправка файлов на сервер: git add . git commit -am ‘feat(main): make course structure’ git push (рис. 6-7).

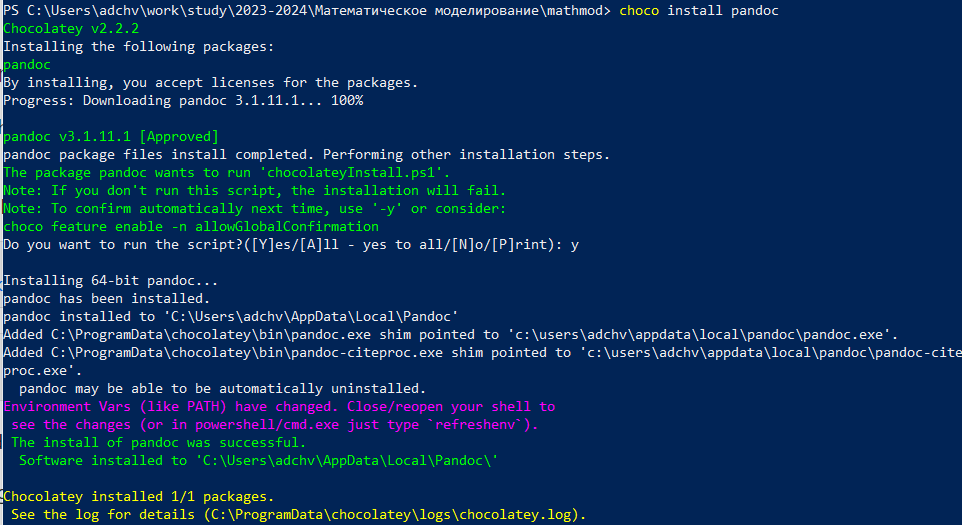


Отправка файлов на сервер.

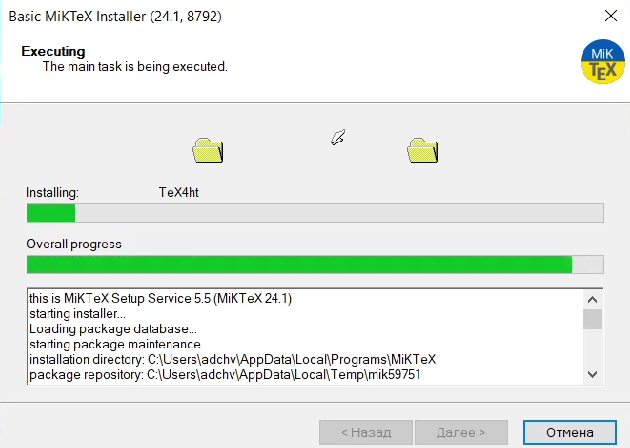


Отправка файлов на сервер.

1. Установка pandoc и MikTex (рис. 8-9).

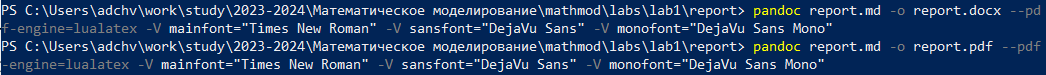


Установка pandoc.

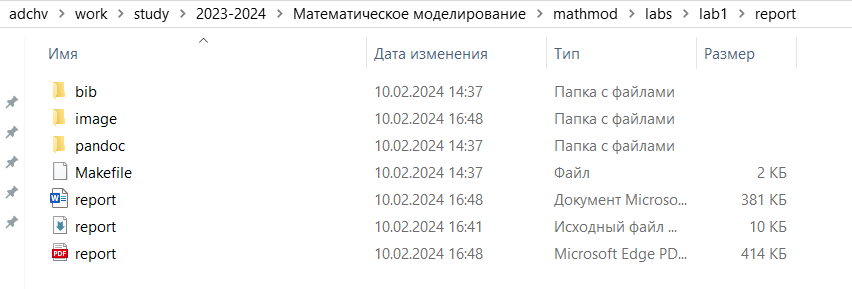


Установка MikTex.

1. Конвертация отчета В docx: pandoc report.md -o report.docx –pdf-engine=lualatex -V mainfont=“Times New Roman” -V sansfont=“DejaVu Sans” -V monofont=“DejaVu Sans Mono” В pdf: pandoc report.md -o report.pdf –pdf-engine=lualatex -V mainfont=“Times New Roman” -V sansfont=“DejaVu Sans” -V monofont=“DejaVu Sans Mono” (рис. 10- 11).



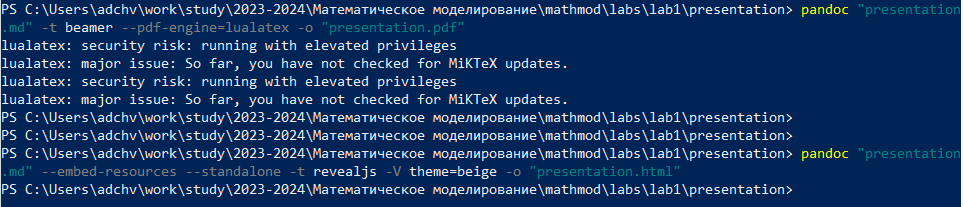
Конвертация.



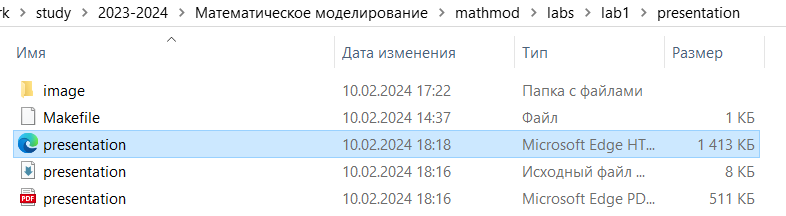
Папка с файлами.

1. Конвертация презентации В pdf: pandoc “presentation.md” -t beamer –pdf-engine=lualatex -o “presentation.pdf” В html: pandoc “presentation.md” –embed-resources –standalone -t revealjs -V theme=beige -o “presentation.html”

(рис. 12- 13).



Конвертация.



Папка с файлами.

# Выводы

Нами было настроено рабочее пространство для лабораторной работы и изучена идеология и применение средств контроля версий, а также освоены умения по работе с git и Markdown.

# Список литературы

• Документация по Git: https://git-scm.com/book/ru/v2

• Документация по Markdown: https://learn.microsoft.com/ru-ru/contribute/markdown-reference

• Документация по MiKTeX: https://kpfu.ru/staff\_files/F2077692752/Inst\_MiKTeX.pdf