

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
“РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ”

Факультет физико-математических и естественных наук

ОТЧЕТ

По лабораторной работе №3
“Markdown”

Выполнил:

Студент группы: НПИбд-02-21

Студенческий билет: №1032217060

ФИО студента: Королев Адам Маратович

Дата выполнения: 25.04.2022

Москва 2022

1 Цель работы:

- Научиться оформлять отчеты с помощью легковесного языка разметки Markdown.

2 Задание:

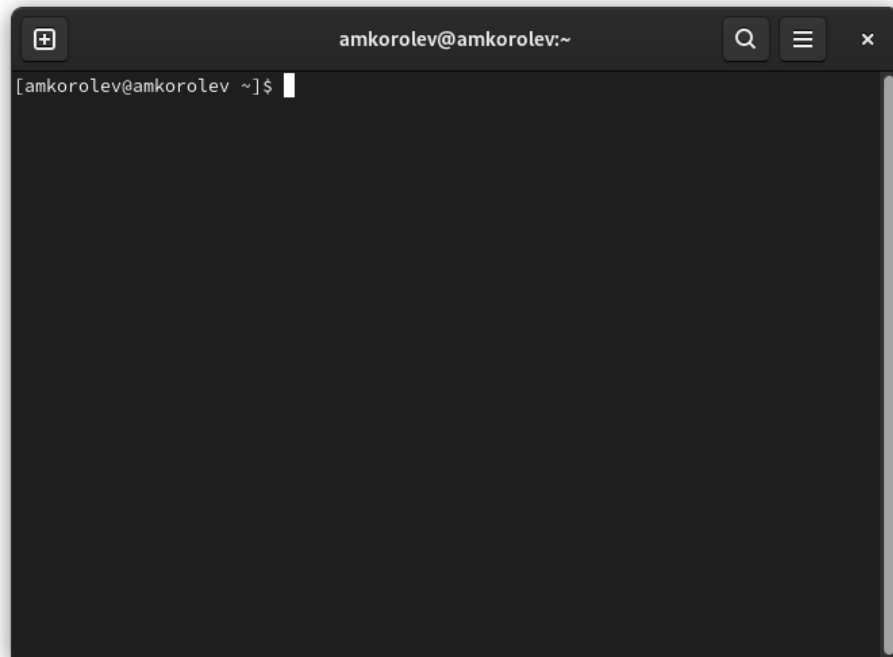
- Сделайте отчет по предыдущей лабораторной работе в формате Markdown.
- В качестве отчета просьба предоставить отчеты в 3 форматах: pdf, docx и md.

3 Теоретическое введение:

Markdown – это облегченный язык текстовой разметки документов. Его придумали в 2004 году блогер Джон Грубер и интернет-активист Аарон Шварц, чтобы быстро форматировать статьи. Требования к языку были следующими: - Чем проще - тем лучше. - Документы с этой разметкой можно перевести в красивый отформатированный вид, как на веб-странице. - Исходный текст материала должен оставаться читаемым даже без преобразования в веб-страницу.

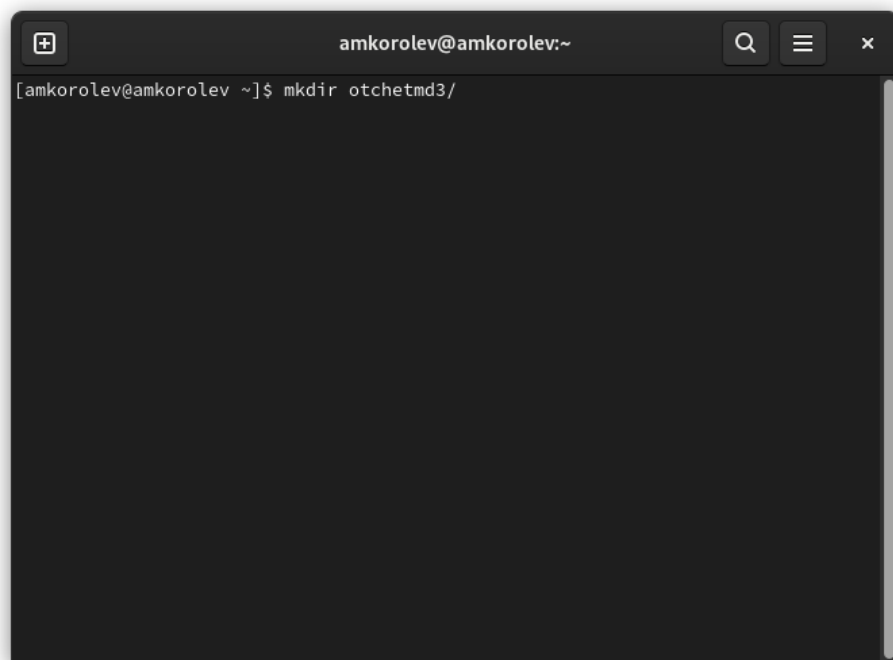
4 Выполнение лабораторной работы:

1. Необходимо запустить терминал Linux



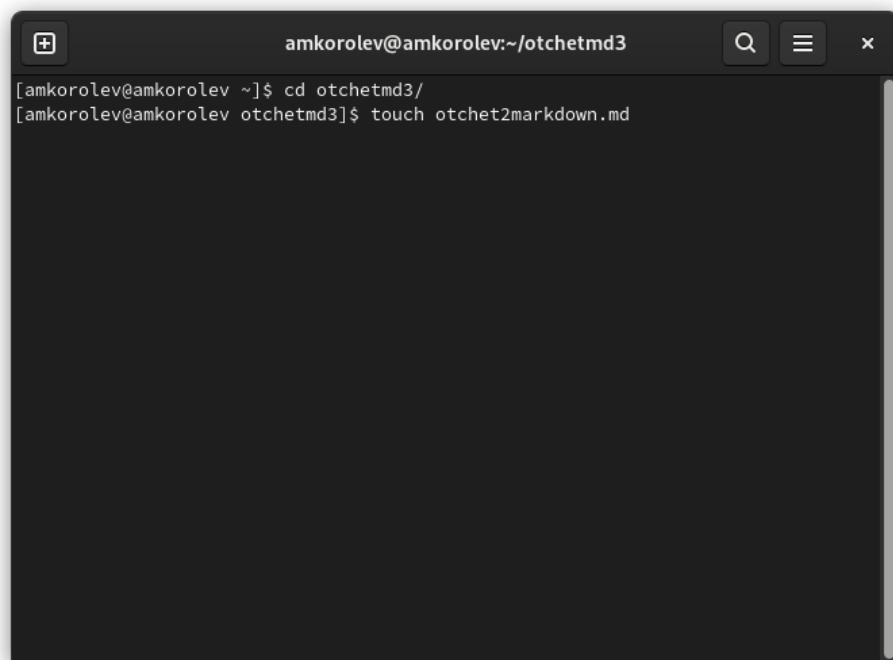
Запущенный терминал

2. Создать папку, в которой в дальнейшем будет создан файл формата .md, а также изображения.



Создаем папку

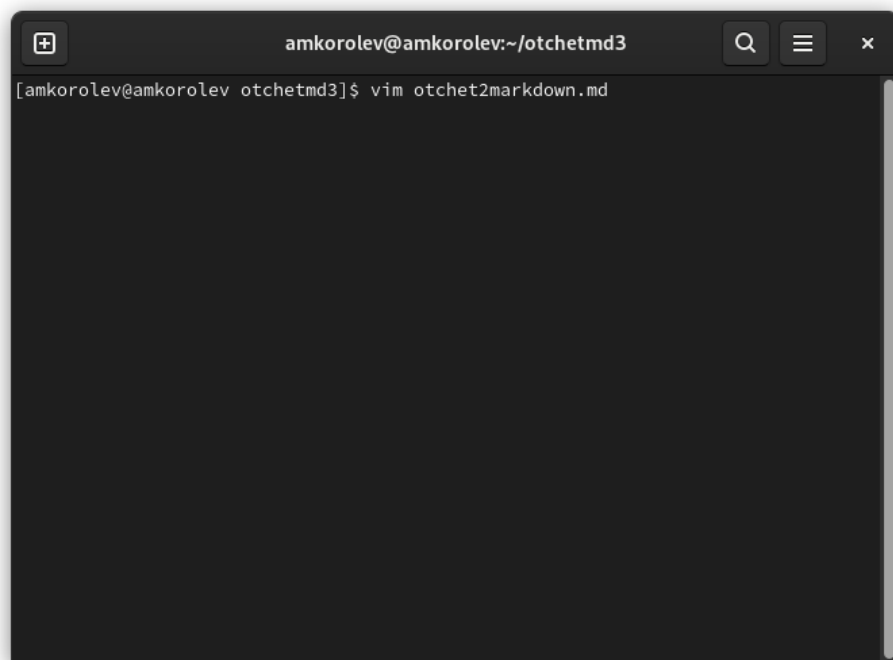
3. Перейдем в созданную папку. В ней создадим файл формата .md, в котором мы будем работать.

A terminal window with a dark background. The title bar at the top shows 'amkorolev@amkorolev:~/otchetmd3' and standard window controls (search, menu, close). The terminal content shows two commands: first, 'cd otchetmd3/' which changes the current directory, and second, 'touch otchet2markdown.md' which creates a new empty file. The prompt changes from '~' to 'otchetmd3' after the first command.

```
amkorolev@amkorolev:~/otchetmd3
[amkorolev@amkorolev ~]$ cd otchetmd3/
[amkorolev@amkorolev otchetmd3]$ touch otchet2markdown.md
```

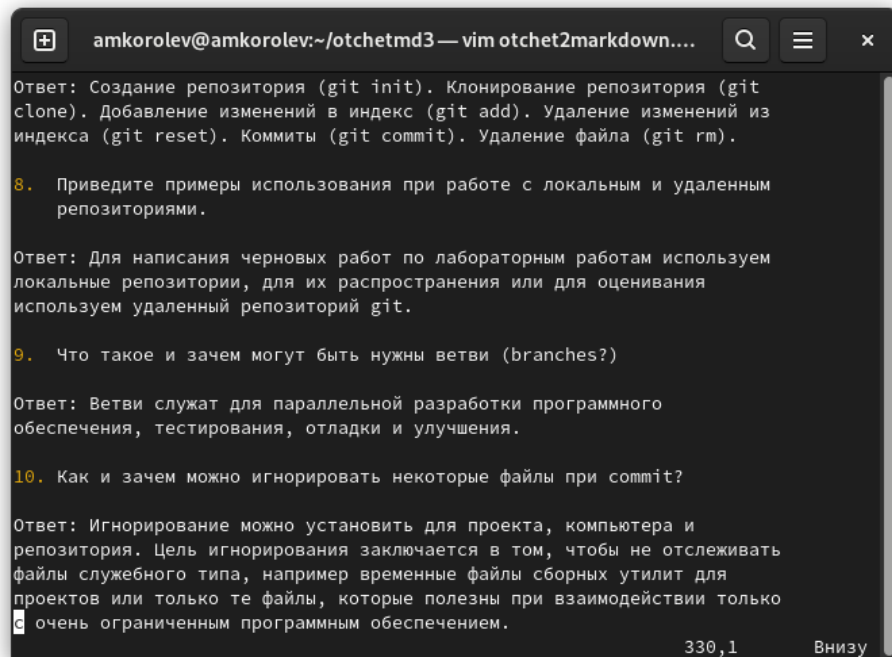
Переходим в папку. Создаем файл формата .md

4. Используя редактор vim, отредактируем файл формата .md.



Используем редактор vim

5. Отредактируем файл формата .md.



```
amkorolev@amkorolev:~/otchetmd3 — vim otchet2markdown....
Ответ: Создание репозитория (git init). Клонирование репозитория (git
clone). Добавление изменений в индекс (git add). Удаление изменений из
индекса (git reset). Коммиты (git commit). Удаление файла (git rm).

8. Приведите примеры использования при работе с локальным и удаленным
репозиториями.

Ответ: Для написания черновых работ по лабораторным работам используем
локальные репозитории, для их распространения или для оценивания
используем удаленный репозиторий git.

9. Что такое и зачем могут быть нужны ветви (branches?)

Ответ: Ветви служат для параллельной разработки программного
обеспечения, тестирования, отладки и улучшения.

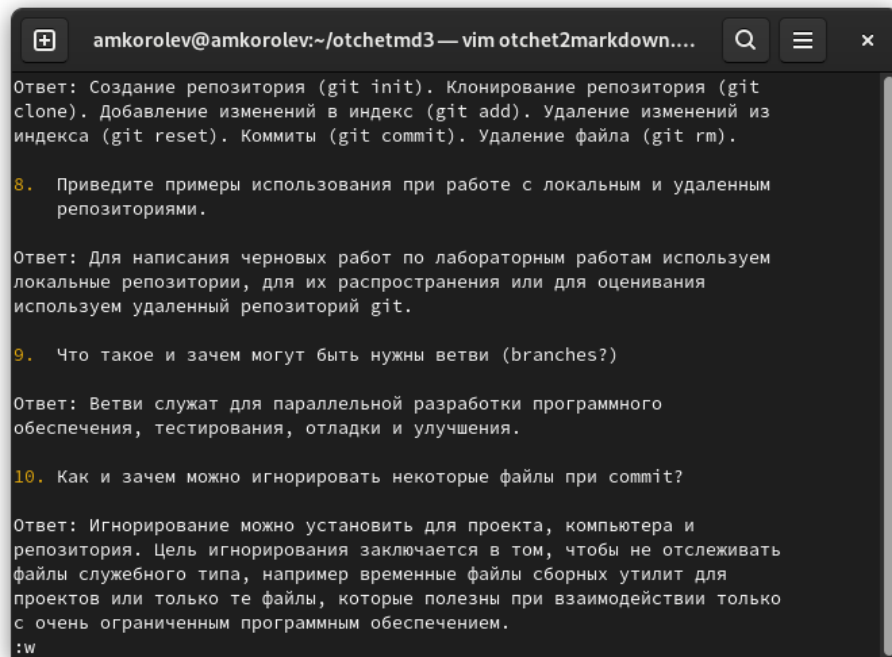
10. Как и зачем можно игнорировать некоторые файлы при commit?

Ответ: Игнорирование можно установить для проекта, компьютера и
репозитория. Цель игнорирования заключается в том, чтобы не отслеживать
файлы служебного типа, например временные файлы сборных утилит для
проектов или только те файлы, которые полезны при взаимодействии только
с очень ограниченным программным обеспечением.

330,1 Внизу
```

Редактируем файл

6. Сохраним файл формата .md, используя команду :w.



```
amkorolev@amkorolev:~/otchetmd3 — vim otchet2markdown....
Ответ: Создание репозитория (git init). Клонирование репозитория (git
clone). Добавление изменений в индекс (git add). Удаление изменений из
индекса (git reset). Коммиты (git commit). Удаление файла (git rm).

8. Приведите примеры использования при работе с локальным и удаленным
репозиториями.

Ответ: Для написания черновых работ по лабораторным работам используем
локальные репозитории, для их распространения или для оценивания
используем удаленный репозиторий git.

9. Что такое и зачем могут быть нужны ветви (branches?)

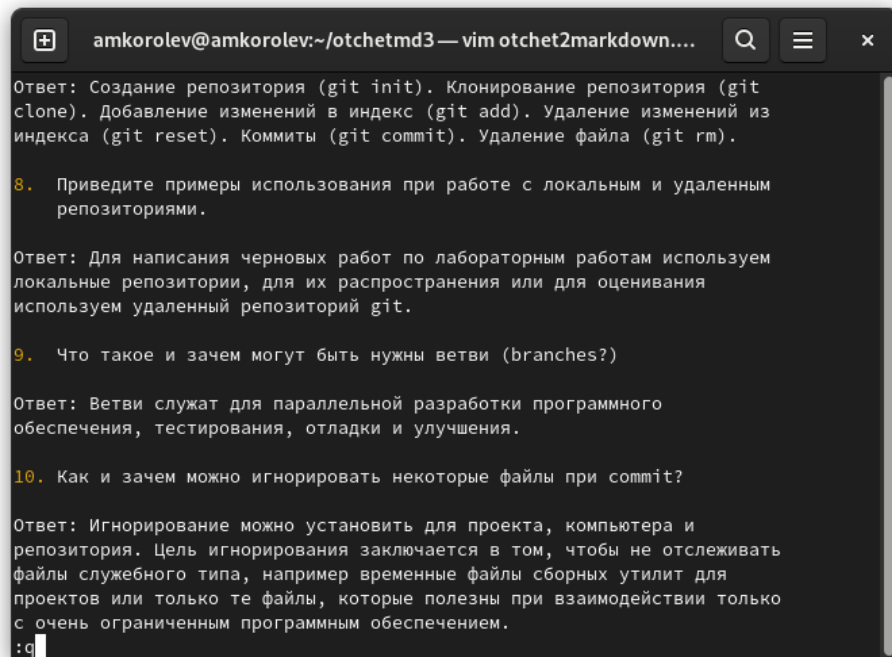
Ответ: Ветви служат для параллельной разработки программного
обеспечения, тестирования, отладки и улучшения.

10. Как и зачем можно игнорировать некоторые файлы при commit?

Ответ: Игнорирование можно установить для проекта, компьютера и
репозитория. Цель игнорирования заключается в том, чтобы не отслеживать
файлы служебного типа, например временные файлы сборных утилит для
проектов или только те файлы, которые полезны при взаимодействии только
с очень ограниченным программным обеспечением.
:w
```

Используем команду :w

7. Выйдем из редактора vim, используя команду :q.



```
amkorolev@amkorolev:~/otchetmd3 — vim otchet2markdown....
Ответ: Создание репозитория (git init). Клонирование репозитория (git
clone). Добавление изменений в индекс (git add). Удаление изменений из
индекса (git reset). Коммиты (git commit). Удаление файла (git rm).

8. Приведите примеры использования при работе с локальным и удаленным
репозиториями.

Ответ: Для написания черновых работ по лабораторным работам используем
локальные репозитории, для их распространения или для оценивания
используем удаленный репозиторий git.

9. Что такое и зачем могут быть нужны ветви (branches?)

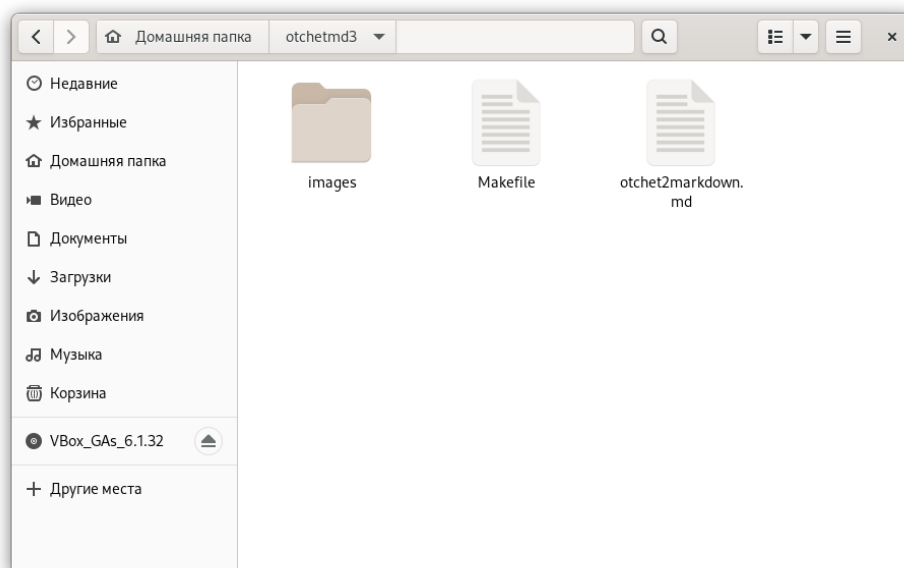
Ответ: Ветви служат для параллельной разработки программного
обеспечения, тестирования, отладки и улучшения.

10. Как и зачем можно игнорировать некоторые файлы при commit?

Ответ: Игнорирование можно установить для проекта, компьютера и
репозитория. Цель игнорирования заключается в том, чтобы не отслеживать
файлы служебного типа, например временные файлы сборных утилит для
проектов или только те файлы, которые полезны при взаимодействии только
с очень ограниченным программным обеспечением.
:q
```

Выходим из редактора, команда :q

8. Перенесем в папку, где располагается файл формата .md, наши изображения и makefile для создания файлов формата docx. и pdf.



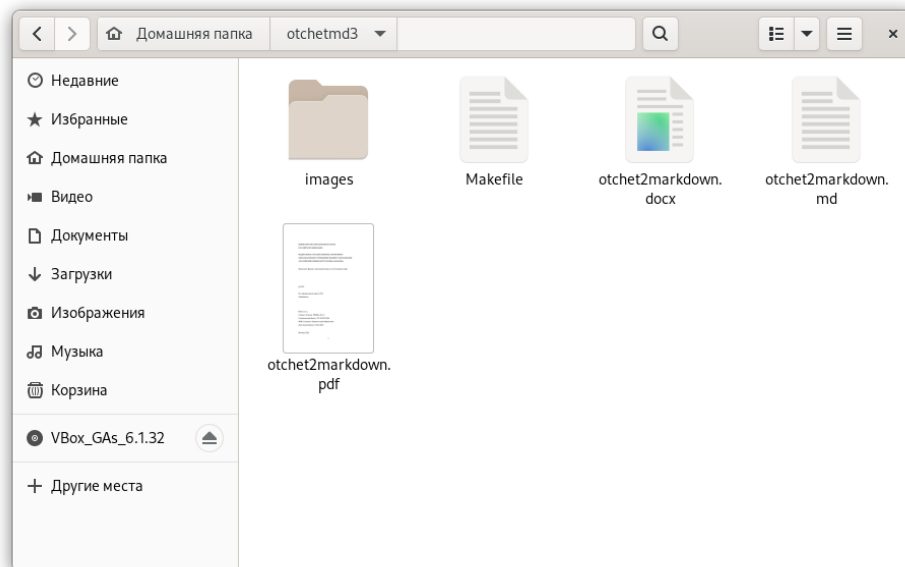
Переносим makefile и изображения

9. Используя имеющийся Makefile, используя команду make создадим файлы форматов docx. и pdf.

```
amkorolev@amkorolev:~/otchetmd3 — make
[amkorolev@amkorolev otchetmd3]$ make
pandoc "otchet2markdown.md" -F pandoc-crossref --number-sections --citeproc -o "otchet2markdown.docx"
pandoc "otchet2markdown.md" -F pandoc-crossref --pdf-engine=lualatex --pdf-engine-opt=--shell-escape --citeproc --number-sections -o "otchet2markdown.pdf"
```

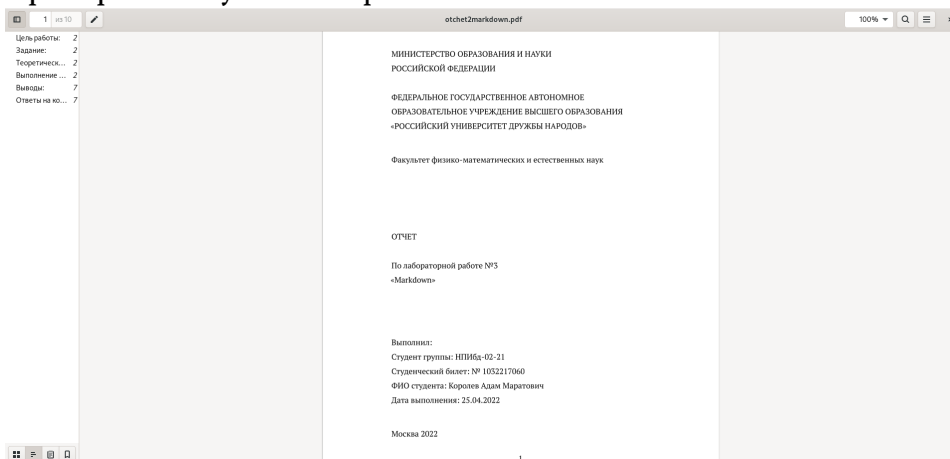
Создаем файлы форматов docx и pdf используя Makefile

10. Проверим, появились ли у нас файлы формата docx. и pdf.



Проверяем наличие файлов

11. Проверим полученные файлы на ошибки.



Проверяем работоспособность файлов

5 Выводы:

В процессе выполнения задания были приобретены навыки по оформлению отчетов с помощью легковесного языка разметки Markdown.