

Лабораторная работа №3

Отчёт

Сергеев Даниил Олегович

Содержание

1	Цель работы	5
2	Задание	6
3	Ход выполнения лабораторной работы	7
3.1	Создание отчёта	7
3.2	Создание файлов отчёта в формате pdf и docx	11
3.3	Отправка файлов на github	12
4	Вывод	15
	Список литературы	16

Список иллюстраций

3.1	Каталог рабочего пространства лабораторной работы.	7
3.2	Файл отчёта report.md.	8
3.3	Замена Front matter.	8
3.4	Цель лабораторной работы.	8
3.5	Задание лабораторной работы.	9
3.6	Ход выполнения лабораторной работы.	9
3.7	Определение пути для рисунка и его добавление.	9
3.8	Изменение номера следующего изображения.	9
3.9	Ответы на контрольные вопросы.	10
3.10	Вывод.	10
3.11	Добавление ссылок к рисункам.	10
3.12	Редактирование пути к рисункам.	11
3.13	Работа команды make.	11
3.14	Вид готового отчёта.	12
3.15	Сохранение файлов в git.	13
3.16	Выполнение git push.	14
3.17	Файлы отчёта на github.	14

Список таблиц

1 Цель работы

Научиться оформлять отчёты с помощью легковесного языка разметки Markdown. [1]

2 Задание

- Сделайте отчёт по предыдущей лабораторной работе в формате Markdown.
- В качестве отчёта просьба предоставить отчёты в 3 форматах: pdf, docx и md (в архиве, поскольку он должен содержать скриншоты, Makefile и т.д.)

3 Ход выполнения лабораторной работы

3.1 Создание отчёта

Перейдем каталог с файлами отчёта и откроем report.md. (рис. 3.2)

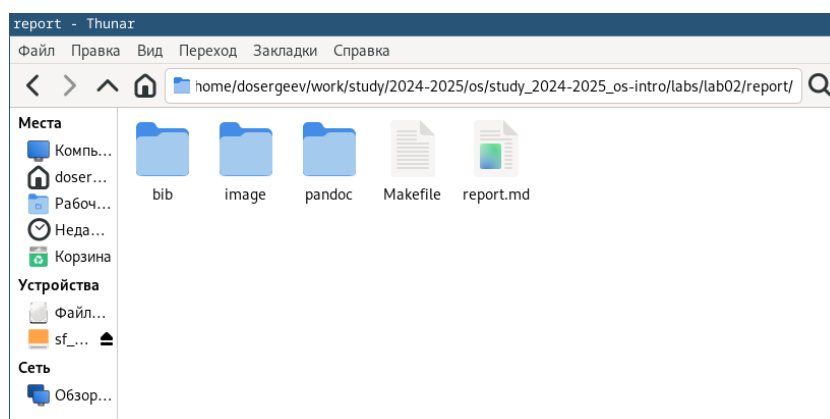
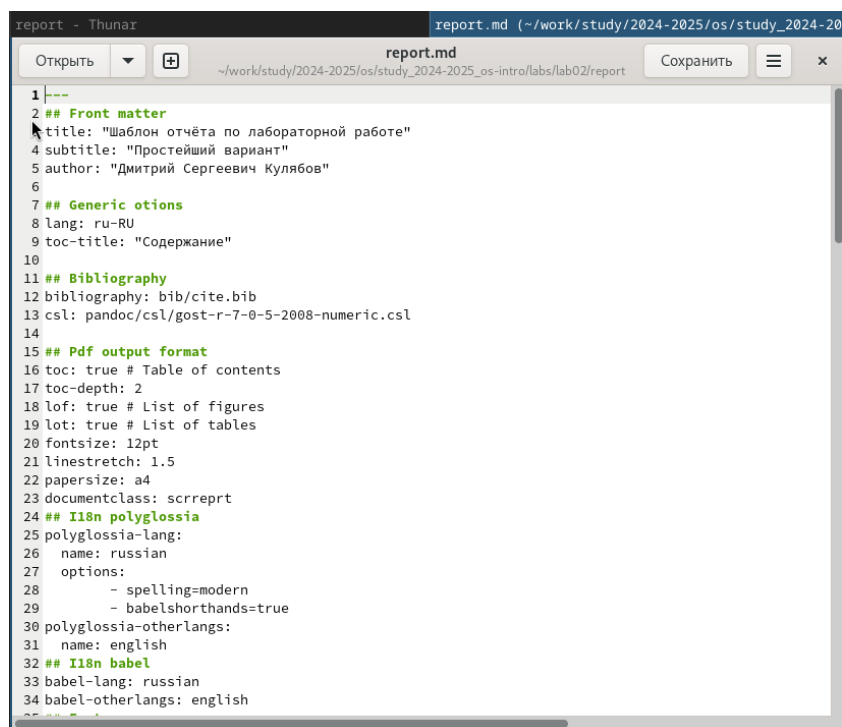


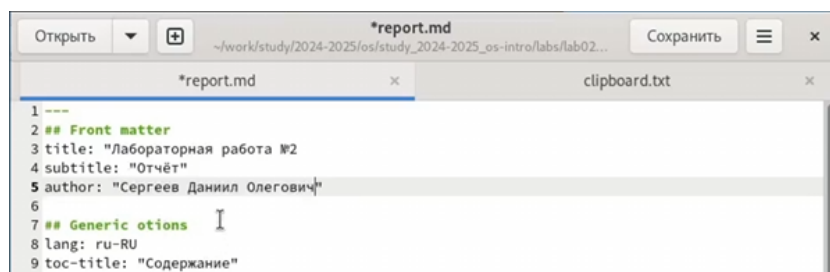
Рис. 3.1: Каталог рабочего пространства лабораторной работы.



```
1 ---
2 ## Front matter
3 title: "Шаблон отчёта по лабораторной работе"
4 subtitle: "Простейший вариант"
5 author: "Дмитрий Сергеевич Кулябов"
6
7 ## Generic options
8 lang: ru-RU
9 toc-title: "Содержание"
10
11 ## Bibliography
12 bibliography: bib/cite.bib
13 csl: pandoc/csl/gost-r-7-0-5-2008-numeric.csl
14
15 ## Pdf output format
16 toc: true # Table of contents
17 toc-depth: 2
18 lof: true # List of figures
19 lot: true # List of tables
20 fontsize: 12pt
21 linestretch: 1.5
22 papersize: a4
23 documentclass: scrreprt
24 ## I18n polyglossia
25 polyglossia-lang:
26   name: russian
27   options:
28     - spelling=modern
29     - babelshorthands=true
30 polyglossia-otherlangs:
31   name: english
32 ## I18n babel
33 babel-lang: russian
34 babel-otherlangs: english
```

Рис. 3.2: Файл отчёта report.md.

Заменим титульный лист.



```
1 ---
2 ## Front matter
3 title: "Лабораторная работа №2"
4 subtitle: "Отчёт"
5 author: "Сергеев Даниил Олегович"
6
7 ## Generic options
8 lang: ru-RU
9 toc-title: "Содержание"
```

Рис. 3.3: Замена Front matter.

Добавим цель, задание, ход выполнения лабораторной работы.



```
71 # Цель работы
72
73 Изучить идеологию и применение средств контроля версий. Освоить умения по работе с git.
74
```

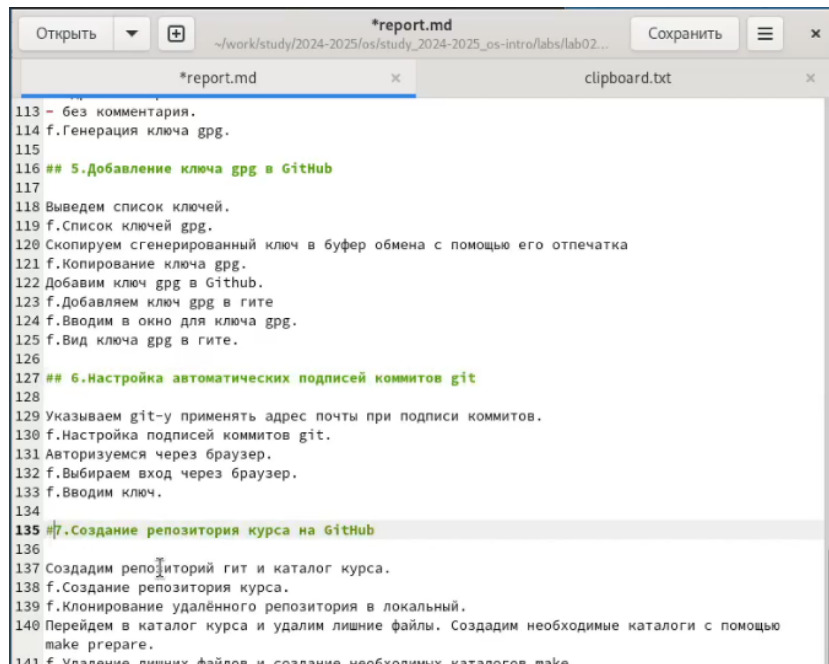
Рис. 3.4: Цель лабораторной работы.


```

75 # Задание
76
77 - Создать базовую конфигурацию для работы с git.
78 - Создать ключ SSH.
79 - Создать ключ PGP.
80 - Настроить подписи git.
81 - Зарегистрироваться на Github.
82 - Создать локальный каталог для выполнения заданий по предмету.
83

```

Рис. 3.5: Задание лабораторной работы.



```

113 - без комментария.
114 f.Генерация ключа gpg.
115
116 ## 5.Добавление ключа gpg в GitHub
117
118 Выведем список ключей.
119 f.Список ключей gpg.
120 Скопируем сгенерированный ключ в буфер обмена с помощью его отпечатка
121 f.Копирование ключа gpg.
122 Добавим ключ gpg в Github.
123 f.Добавляем ключ gpg в гите
124 f.Вводим в окно для ключа gpg.
125 f.Вид ключа gpg в гите.
126
127 ## 6.Настройка автоматических подписей коммитов git
128
129 Указываем git-у применять адрес почты при подписи коммитов.
130 f.Настройка подписей коммитов git.
131 Авторизуемся через браузер.
132 f.Выбираем вход через браузер.
133 f.Вводим ключ.
134
135 ## 7.Создание репозитория курса на GitHub
136
137 Создадим репозиторий гит и каталог курса.
138 f.Создание репозитория курса.
139 f.Клонирование удалённого репозитория в локальный.
140 Перейдем в каталог курса и удалим лишние файлы. Создадим необходимые каталоги с помощью
    make prepare.
141 f.Удаление лишних файлов и создание необходимых каталогов make

```

Рис. 3.6: Ход выполнения лабораторной работы.

Начнем оформлять рисунки, используемые в лабораторной работе.

```

84 # Выполнение лабораторной работы
85
86 ## 1.Установка необходимого ПО (git, gh)
87 Установим git и gh.
88 ![Процесс установки git и gh.](1.PNG){#fig:001 width=70%}
89 Необходимые пакеты уже установлены, поэтому продолжим.
90

```

Рис. 3.7: Определение пути для рисунка и его добавление.

Для каждого нового изображения поменяем номер, чтобы небыло ошибок при генерации pdf. (рис. 3.8)

```

## 2.Базовая настройка git
Зададим имя и почту владельца репозитория, настроим utf-8 в выводе сообщений git, зададим
имя начальной ветки master, параметр autocrlf и safecrlf.
![Настройка параметров git.](2.PNG){#fig:002 width=70%}

```

Рис. 3.8: Изменение номера следующего изображения.

Добавим ответы на контрольные вопросы.

```
145
146 # Ответы на контрольные вопросы
147
148 1. Системы контроля версий – это инструменты, позволяющие организовать работу над проектом
    разработки, выкладывать его обновления и контролировать релизы и изменения кода. Они
    предназначены для отслеживания изменений, защиты исходного кода от удаления и изменения,
    возможности отката изменений и командной работы для 10 и более человек.
149 2. Хранилище – репозиторий в котором хранятся файлы проекта в различных версиях.
150 Commit – комментарий внесённых изменений в репозитории.
151 История – история изменений файлов проекта.
152 Рабочая копия – копия, созданная из определенной версии репозитория, которую модифицирует
    разработчик.
153 3. Централизованные системы контроля версий имеют единый сервер под хранение проекта, для
    изменения которых необходимо скачать необходимые файлы, изменить и вернуть обратно на
    сервер. Пример централизованной VCS: Subversion. Децентрализованные системы полностью
    копируют удалённый репозиторий в локальный. При этом внесённые изменения отправляются на
    сервер в качестве новой версии. Пример: git.
154 4. Создается репозиторий для работы с проектом, при необходимости файлы обновляются локально
    и отправляются на сервер в качестве новой версии.
155 5. Репозиторий копируется локально. После внесения изменений файлы загружаются на сервер в
    качестве отдельной версии. После этого изменённая ветка может быть объединена с текущей и
    отправлена в релиз.
156 6. Хранение файлов проекта, отслеживание версий, защита от изменений, работа в команде.
157 7. git clone – клонирование проекта с сервера в указанный локальный репозиторий.
```

Рис. 3.9: Ответы на контрольные вопросы.

Добавим вывод лабораторной работы.

```
178
179 # Вывод
180
181 В результате выполнения лабораторной работы я изучил как применять средства контроля версий
    для работы с удалённым репозиторием и освоил умения по работе с git.
```

Рис. 3.10: Вывод.

К некоторым рисункам добавим ссылки (рис. 3.11)

```
84 # Выполнение лабораторной работы
85
86 ## 1. Установка необходимого ПО (git, gh)
87 Установим git и gh. (рис. [-@fig:001])
88 ![Процесс установки git и gh.](1.PNG){#fig:001 width=70%}
89 Необходимые пакеты уже установлены, поэтому продолжим.
90
```

Рис. 3.11: Добавление ссылок к рисункам.

Отредактируем относительный путь к рисункам, расставим строки через одну для корректного отображения отчёта в сгенерированных файлах.

```

99 ## Создание ключа ssh
100
101 Создадим ключ ssh по алгоритму ed25519. (рис. [-@fig:003])
102
103 ![Генерация ключа ssh.](image/3.PNG){#fig:003 width=70%}
104
105 Выведем ключ, скопируем и вставим в гит.
106
107 ![Копирование ключа ssh.](image/4.PNG){#fig:004 width=70%}
108
109 ![Добавляем ключ ssh в гите.](image/5.PNG){#fig:005 width=70%}
110
111 ![Вид ключа ssh в гите.](image/6.PNG){#fig:006 width=70%}
112
113

```

Рис. 3.12: Редактирование пути к рисункам.

3.2 Создание файлов отчёта в формате pdf и docx

Пропишем команду make для создания файлов отчёта в pdf и docx. (рис. 3.13)

```

[dosergeev@vbox report]$ make
pandoc "report.md" --filter pandoc-crossref --number-sections
--citeproc -o "report.docx"
pandoc "report.md" --filter pandoc-crossref --pdf-engine=lualat
tex --pdf-engine-opt=--shell-escape --citeproc --number-sect
ions -o "report.pdf"
[WARNING] [makePDF] LaTeX Warning: Empty bibliography on input
line 500.
[dosergeev@vbox report]$

```

Рис. 3.13: Работа команды make.

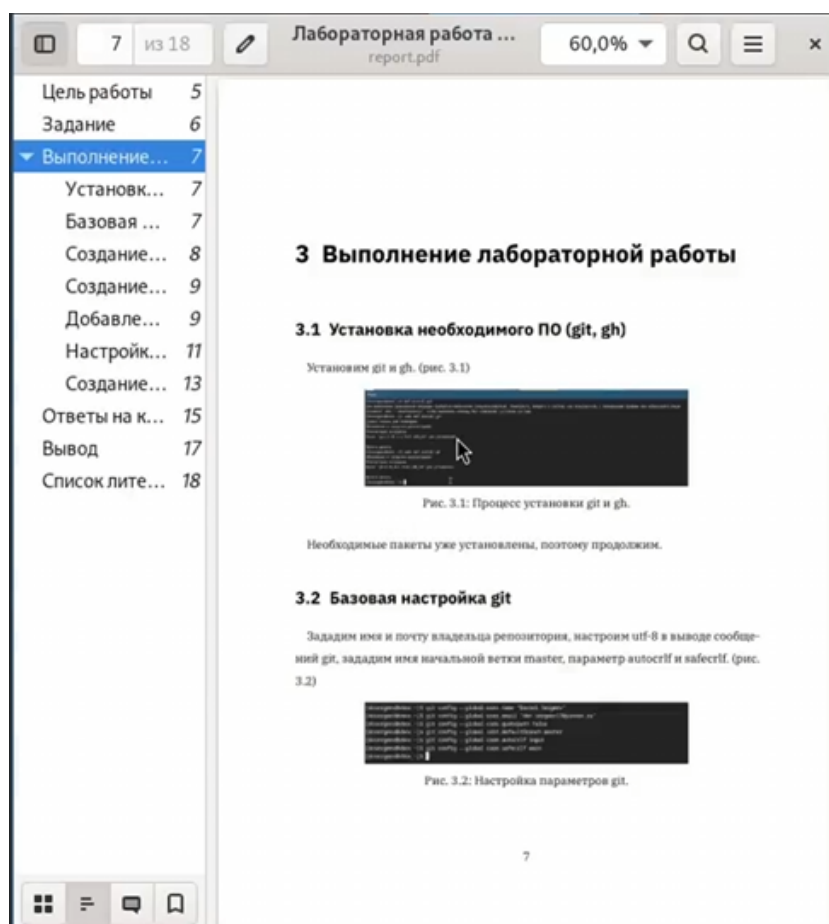


Рис. 3.14: Вид готового отчёта.

3.3 Отправка файлов на github

Проиндексируем все изменения и проведем коммит.

```

[dosergeev@vbox report]$ cd ~/work/study/2024-2025/os/study_20
24-2025_os-intro/
[dosergeev@vbox study_2024-2025_os-intro]$ git add .
[dosergeev@vbox study_2024-2025_os-intro]$ git commit -am "fea
t(main): add lab02"
[master d643458] feat(main): add lab02
27 files changed, 140 insertions(+), 152 deletions(-)
create mode 100755 labs/lab02/report/image/1.PNG
create mode 100755 labs/lab02/report/image/10.PNG
create mode 100755 labs/lab02/report/image/11.PNG
create mode 100755 labs/lab02/report/image/12.PNG
create mode 100755 labs/lab02/report/image/13.PNG
create mode 100755 labs/lab02/report/image/14.PNG
create mode 100755 labs/lab02/report/image/15.PNG
create mode 100755 labs/lab02/report/image/16.PNG
create mode 100755 labs/lab02/report/image/17.PNG
create mode 100755 labs/lab02/report/image/18.PNG
create mode 100755 labs/lab02/report/image/19.PNG
create mode 100755 labs/lab02/report/image/2.PNG
create mode 100755 labs/lab02/report/image/20.PNG
create mode 100755 labs/lab02/report/image/3.PNG
create mode 100755 labs/lab02/report/image/4.PNG
create mode 100755 labs/lab02/report/image/5.PNG
create mode 100755 labs/lab02/report/image/6.PNG
create mode 100755 labs/lab02/report/image/7.PNG
create mode 100755 labs/lab02/report/image/8.PNG
create mode 100755 labs/lab02/report/image/9.PNG
delete mode 100644 labs/lab02/report/image/placement_800_600_t
ech.jpg
create mode 100644 labs/lab02/report/report.docx
create mode 100644 labs/lab02/report/report.pdf
[dosergeev@vbox study_2024-2025_os-intro]$

```

Рис. 3.15: Сохранение файлов в git.

Отправим файлы на сервер github. (рис. 3.16)

```
[dosergeev@vbox study_2024-2025_os-intro]$ git push
Перечисление объектов: 47, готово.
Подсчет объектов: 100% (44/44), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (34/34), готово.
Запись объектов: 100% (36/36), 3.34 МБ | 3.94 МБ/с, готово.
Total 36 (delta 6), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (6/6), completed with 4 local o
bjects.
To github.com:FrigatZero/study_2024-2025_os-intro.git
   1642e1f..d643458  master -> master
[dosergeev@vbox study_2024-2025_os-intro]$
```

Рис. 3.16: Выполнение git push.

study_2024-2025_os-intro / labs / lab02 / report /

FrigatZero feat(main): add lab02 d643458 · 4 days ago History

Name	Last commit message	Last commit date
..		
bib	feat(main): add lab02	4 days ago
image	feat(main): add lab02	4 days ago
pandoc	feat(main): make course stucture	last week
Makefile	feat(main): make course stucture	last week
report.docx	feat(main): add lab02	4 days ago
report.md	feat(main): add lab02	4 days ago
report.pdf	feat(main): add lab02	4 days ago

Рис. 3.17: Файлы отчёта на github.

4 Вывод

В результате выполнения лабораторной работы я научился оформлять отчёты с помощью легковесного языка разметки Markdown и создавать файлы отчёта в разных форматах.

Список литературы

1. Kulyabov. Лабораторная работа № 3. Markdown. RUDN.