

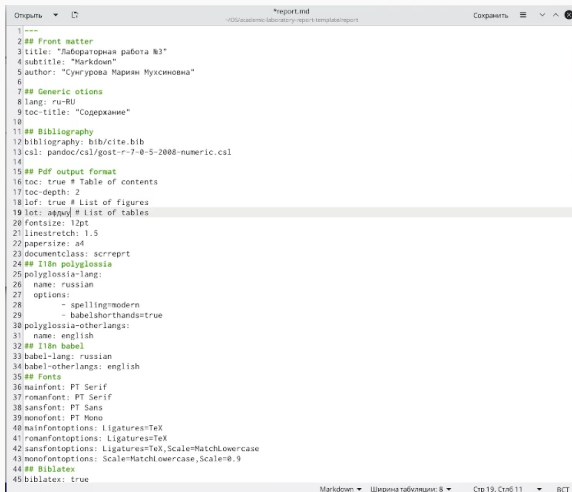
Лабораторная работа №3

Сунгурова Мариян Мухсиновна, НКНбд-01-21

1. Цель работы
2. Процесс выполнения
3. Вывод
4. Источники

Научиться оформлять отчёты с помощью легковесного языка разметки Markdown.

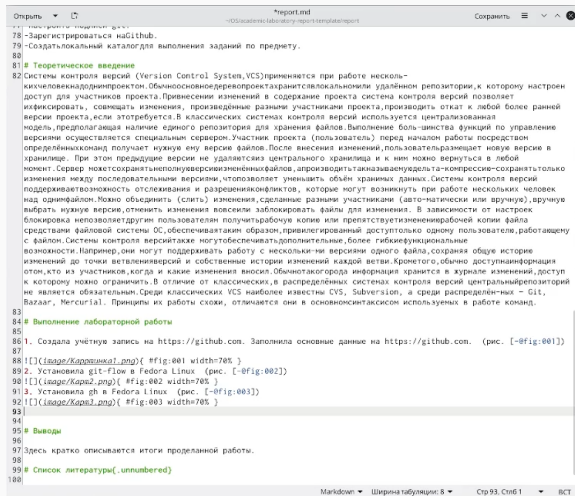
1. Приступила к редактированию файла report.md (рис. 1)



```
1 ---
2 ## Front matter
3 title: "Лабораторная работа №3"
4 subtitle: "Markdown"
5 author: "Сунгурова Мариан Мухсиновна"
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: false # 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
35 ## Fonts
36 mainfont: PT Serif
37 romanfont: PT Serif
38 sansfont: PT Sans
39 monofont: PT Mono
40 mainfontoptions: Ligatures=TeX
41 romanfontoptions: Ligatures=TeX
42 sansfontoptions: Ligatures=TeX,Scale=MatchLowercase
43 monofontoptions: Scale=MatchLowercase,Scale=0.9
44 ## Biblatex
45 biblatex: true
```

Figure 1: Окно текстового редактора с открытым report.md файлом

2. Теоретическое введение (рис. 2)



```
Открыть  ▾  ~\OS\academic\laboratory-report-template\report  Сохранить  ▾  ×  ↻

78 -Зарегистрироваться наGitHub.
79 -Создатьлокальный каталогдля выполнения заданий по предмету.
80
81 # Теоретическое введение
82 Системы контроля версий (Version Control System,VCS)применяется при работе несколь-
кихчеловекнадоднимпроектом.Обычноосновное деревопроектахранитсвалочноили удалённом репозитории,к которому настроен
доступ для участников проекта.Привнесении изменений в содержание проекта система контроля версий позволяет
ихфиксируют, совмещать изменения, произведённые разными участниками проекта,производить откат к любой более ранней
версии проекта,если этотребуется.В классических системах контроля версий используется централизованная
модель,предполагающая наличие единого репозитория для хранения файлов.Выполнение больш-инства функций по управлению
версиями осуществляется специальным сервером.Участник проекта (пользователь) перед началом работы посредством
определённыхкоманд получает нужную ему версию файлов.После внесения изменений,пользовательразмещает новую версию в
хранилище. При этом предыдущие версии не удаляютсяиз центрального хранилища и к ним можно вернуться в любой
момент.Сервер можетсохранятьнеполнуюверсиюизменённыхфайлов,апроизводитьтакназываемуюдельта-компрессию-сохранятьтолько
изменения между последовательными версиями,чтопозволяет уменьшить объём хранимых данных.Системы контроля версий
поддерживаютвозможность отслеживания и разрешенияконфликтов, которые могут возникнуть при работе нескольких человек
над однимфайлом.Можно объединить (слить) изменения,сделанные разными участниками (авто-матически или вручную),вручную
выбрать нужную версию,отменить изменения повсейли заблокировать файлы для изменения. В зависимости от настроек
блокировка не позволяетдругим пользователям получитьрабочую копию или препятствуетизменениирабочей копии файла
средствами файловой системы ОС,обеспечиваятаким образом,привилегированный доступтолько одному пользователю,работающему
с файлом.Системы контроля версийтакже могутобеспечиватьдополнительные,более гибкиефункциональные
возможности.Например,они могут поддерживать работу с несколькими версиями одного файла,сохраняя общую историю
изменений до точки ветвленияверсий и собственные истории изменений каждой ветви.Кроме того,обычно доступнаяинформация
отом,кто из участников,когда и какие изменения вносил.Обычнотакого рода информация хранится в журнале изменений,доступ
к которому можно ограничить.В отличие от классических,в распределённых системах контроля версий центральныйрепозиторий
не является обязательным.Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых – Git,
Bazaar, Mercurial. Принципы их работы схожи, отличается они в основномсинтаксисом используемых в работе команд.

83
84 # Выполнение лабораторной работы
85
86 1. Создала учётную запись на https://github.com. Заполнила основные данные на https://github.com. (рис. [-@fig:001])
87
88 ![image/Кадринка1.png]{ #fig:001 width=70% }
89 2. Установила git-flow в Fedora Linux (рис. [-@fig:002])
90 ![image/Кадр2.png]{ #fig:002 width=70% }
91 3. Установила gh в Fedora Linux (рис. [-@fig:003])
92 ![image/Кадр3.png]{ #fig:003 width=70% }
93
94
95 # Выводы
96
97 Здесь кратко описываются итоги проделанной работы.
98
99 # Список литературы(.unnumbered)
100
```

Figure 2: Окно текстового редактора с открытым report.md файлом

3. Создание ссылок на фото из папки images (рис. 3)

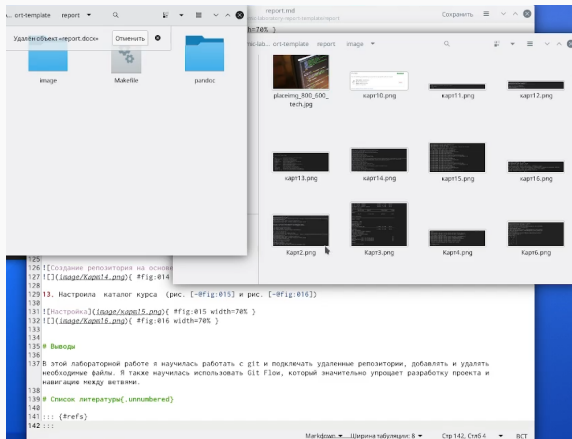
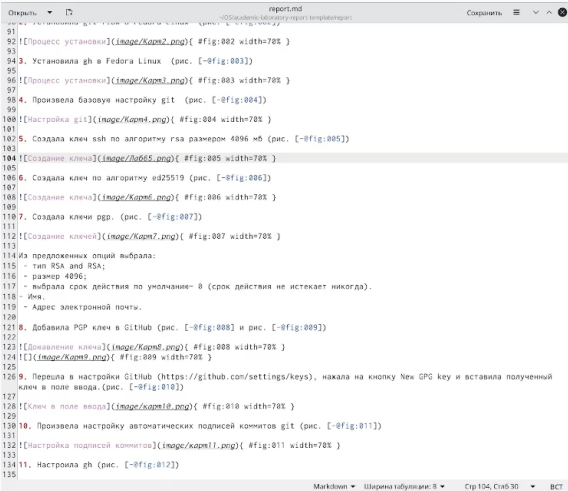


Figure 3: Папка с фото

4. Описала шаги, прикрепив фото (рис. 4)



```
report.md
Сохранить

91
92![[Процесс установки](image/Карт2.png){ #fig:002 width=70% }
93
943. Установила gh в Fedora Linux (рис. [-@fig:003])
95
96![[Процесс установки](image/Карт3.png){ #fig:003 width=70% }
97
984. Произвела базовую настройку git (рис. [-@fig:004])
99
100![[Настройка git](image/Карт4.png){ #fig:004 width=70% }
101
1025. Создала ключ ssh по алгоритму rsa размером 4096 мб (рис. [-@fig:005])
103
104![[Создание ключа](image/Карт5.png){ #fig:005 width=70% }
105
1066. Создала ключ по алгоритму ed25519 (рис. [-@fig:006])
107
108![[Создание ключа](image/Карт6.png){ #fig:006 width=70% }
109
1107. Создала ключи pgp. (рис. [-@fig:007])
111
112![[Создание ключей](image/Карт7.png){ #fig:007 width=70% }
113
114Из предложенных опций выбрала:
115- тип RSA and RSA;
116- размер 4096;
117- выбрала срок действия по умолчанию- 0 (срок действия не истекает никогда).
118- имя.
119- Адрес электронной почты.
120
1218. Добавила PGP ключ в GitHub (рис. [-@fig:008] и рис. [-@fig:009])
122
123![[Добавление ключа](image/Карт8.png){ #fig:008 width=70% }
124![[image/Карт9.png){ #fig:009 width=70% }
125
1269. Перешла в настройки GitHub (https://github.com/settings/keys), нажала на кнопку New GPG key и вставила полученный
    ключ в поле ввода.(рис. [-@fig:010])
127
128![[Ключ в поле ввода](image/Карт10.png){ #fig:010 width=70% }
129
13010. Произвела настройку автоматических подписей коммитов git (рис. [-@fig:011])
131
132![[Настройка подписей коммитов](image/Карт11.png){ #fig:011 width=70% }
133
13411. Настроила gh (рис. [-@fig:012])
135

Markdown Шрифт: 14pt Стр 104, Стр 50 BCT
```

Figure 4: Этапы выполнения

5. Описала источники (рис. 5)

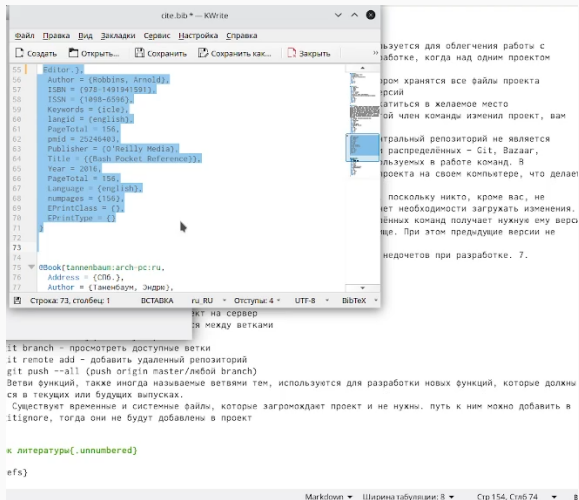


Figure 5: Файл bib с источниками

В этой лабораторной работе я научилась оформлять отчеты при помощи легковесного языка разметки Markdown, работать с фото и bib файлами для источников.

1. КулябовД. С. Введение в операционные системы –Лекция;