

Отчёт по лабораторной работе №3

дисциплина: Операционные системы

Латаева Гюзелия Андреевна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	18
6	Список литературы	19

Список иллюстраций

4.1	Проверка компиляции шаблона	8
4.2	Удаление файлов	8
4.3	Редактирование титульного листа	8
4.4	Редактирование цели работы	9
4.5	Выделяем “Задание” и “Теоретическое введение”	9
4.6	Удаляем “Задание” и “Теоретическое введение”	9
4.7	Редактирование последовательности выполнения работ	10
4.8	Редактирование заключения	10
4.9	Редактирование контрольных вопросов	10
4.10	Удаление списка литературы	11
4.11	Просмотр титульного листа	11
4.12	Просмотр содержания	12
4.13	Просмотр списка иллюстраций	13
4.14	Просмотр цели работы	14
4.15	Просмотр последовательности выполнения работ	15
4.16	Просмотр заключения	16
4.17	Просмотр контрольных вопросов	17

Список таблиц

1 Цель работы

Научиться оформлять отчёты с помощью легковесного языка разметки Markdown

2 Задание

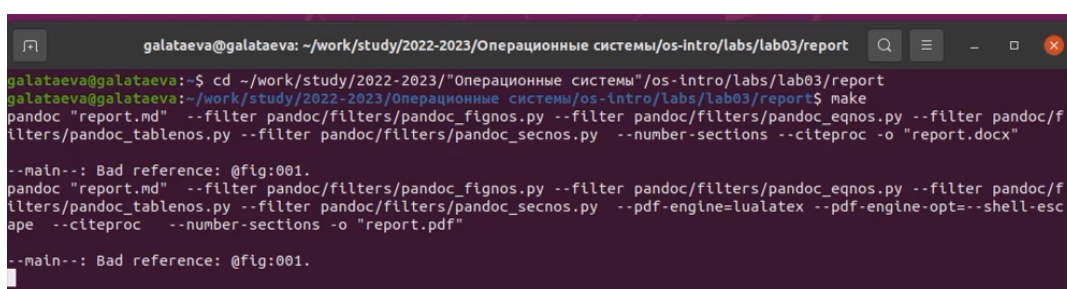
Сделать отчёт по предыдущей лабораторной работе в формате Markdown.

3 Теоретическое введение

Markdown - облегчённый язык разметки, созданный с целью обозначения форматирования в простом тексте, с максимальным сохранением его читаемости человеком, и пригодный для машинного преобразования в языки для продвинутых публикаций (HTML, Rich Text и других).

4 Выполнение лабораторной работы

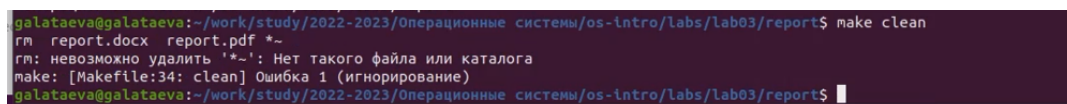
1) Переходим в каталог с шаблоном отчета по лабораторной работе № 3 и проведем компиляцию шаблона с использованием Makefile (рис. 4.1).



```
galataeva@galataeva: ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab03/report
galataeva@galataeva:~$ cd ~/work/study/2022-2023/Операционные системы/os-intro/labs/lab03/report
galataeva@galataeva:~/work/study/2022-2023/Операционные системы/os-intro/labs/lab03/report$ make
pandoc "report.md" --filter pandoc/filters/pandoc_fignos.py --filter pandoc/filters/pandoc_eqnos.py --filter pandoc/filters/pandoc_tablenos.py --filter pandoc/filters/pandoc_secnos.py --number-sections --citeproc -o "report.docx"
--main--: Bad reference: @fig:001.
pandoc "report.md" --filter pandoc/filters/pandoc_fignos.py --filter pandoc/filters/pandoc_eqnos.py --filter pandoc/filters/pandoc_tablenos.py --filter pandoc/filters/pandoc_secnos.py --pdf-engine=lualatex --pdf-engine-opt=-shell-escape --citeproc --number-sections -o "report.pdf"
--main--: Bad reference: @fig:001.
```

Рис. 4.1: Проверка компиляции шаблона

2) Удаляем полученные файлы с использованием Makefile (рис. 4.2).

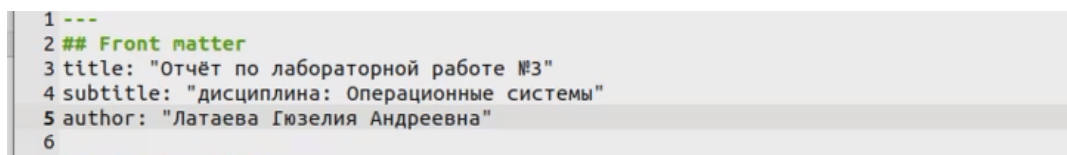


```
galataeva@galataeva:~/work/study/2022-2023/Операционные системы/os-intro/labs/lab03/report$ make clean
rm report.docx report.pdf *~
rm: невозможно удалить '*~': Нет такого файла или каталога
make: [Makefile:34: clean] Ошибка 1 (игнорирование)
galataeva@galataeva:~/work/study/2022-2023/Операционные системы/os-intro/labs/lab03/report$
```

Рис. 4.2: Удаление файлов

3) Редактируем отчет

- Титульный лист (рис. 4.3):



```
1 ---
2 ## Front matter
3 title: "Отчёт по лабораторной работе №3"
4 subtitle: "дисциплина: Операционные системы"
5 author: "Латаева Гюзелия Андреевна"
6
```

Рис. 4.3: Редактирование титульного листа

- Цель работы (рис. 4.4):

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

Рис. 4.4: Редактирование цели работы

- Выделяем и удаляем “Задание” и “Теоретическое введение” т.к. в отчете по лаб.2 они отсутствуют (рис. 4.5), (рис. 4.6):

```
3 # Задание
4
5 Здесь приводится описание задания в соответствии с рекомендациями
6 методического пособия и выданным вариантом.
7
8 # Теоретическое введение
9
10 Здесь описываются теоретические аспекты, связанные с выполнением работы.
11
12 Например, в табл. @tbl:std-dir приведено краткое описание стандартных каталогов Unix.
13
14 : Описание некоторых каталогов файловой системы GNU Linux {#tbl:std-dir}
15
16 | Имя каталога | Описание
17 |-----|-----
18 | '/' | Корневая директория, содержащая всю
19 | файловую
```

Рис. 4.5: Выделяем “Задание” и “Теоретическое введение”

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

# Выполнение лабораторной работы
Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. @fig:001).
! [Название рисунка](image/placeimg_800_600_tech.jpg){#fig:001 width=70%}
```

Рис. 4.6: Удаляем “Задание” и “Теоретическое введение”

- Переименовываем “Выполнение лабораторной работы” в “Последовательность выполнения работ” и редактируем этот пункт (рис. 4.7):

```
74 # Последовательность выполнения работы
75
76 1) Создаю учётную запись на https://github.com и заполняю данные.
77
78 ![Регистрация](image/1.png){#fig:001 width=70%}
79
80
81 2) Установка программного обеспечения
82 - Установка git-flow: сначала необходимо установить makecache а потом уже сам gitflow
83
84 ![Установка makecache](image/2.png){#fig:002 width=70%}
85
86 ![Установка epel_release](image/3.png){#fig:003 width=70%}
87
88 ![Установка gitflow](image/4.png){#fig:004 width=70%}
89
90 - Установка gh: на Centos 7 необходимо провести ряд манипуляций перед тем, как установить gh
91
92 ![Установка config-manager](image/5.png){#fig:005 width=70%}
93
94 ![Установка репозитория с gh](image/6.png){#fig:006 width=70%}
95
```

Рис. 4.7: Редактирование последовательности выполнения работ

- Заключение (рис. 4.8):

```
167 # Заключение
168
169 Я изучила применение средств контроля версий и в какой-то мере освоила навыки по работе с
git.
170
171
```

Рис. 4.8: Редактирование заключения

- Добавляем и заполняем пункт “Контрольные вопросы” (рис. 4.9):

```
171
172 # Контрольные вопросы
173
174 1. *Что такое системы контроля версий (VCS) и для решения каких задач они предназначены?*
175
176 Система контроля версий (от англ. Version Control System, VCS) – это, грубо говоря, место
хранения кода. Она создана для разработки продуктов: на хранение кода, синхронизацию работы
нескольких человек, создание релизов и т.д.
177
178
179 2. *Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая
копия.*
180 Хранилище (репозиторий) – это хранилище всех версий кода. Он бывает трех видов:
181 Локальный – расположен на одном компьютере, и работать с ним может только один человек.
182 Централизованный – расположен на сервере, куда имеют доступ сразу несколько программистов.
183 Распределенный – самый удобный вариант с облачным хранилищем. Главный репозиторий хранится
в облаке, а его локальные копии – у разработчиков на компьютерах.
184 Commit - запись изменений.
185 История - список предыдущих изменений.
186 Рабочая копия – текущее состояние файлов проекта, основанное на версии из хранилища (обычно
на последней).
```

Рис. 4.9: Редактирование контрольных вопросов

- Удаляем пункт “Список литературы”, т.к. в отчете по лаб.2 он отсутствует (рис. 4.10):

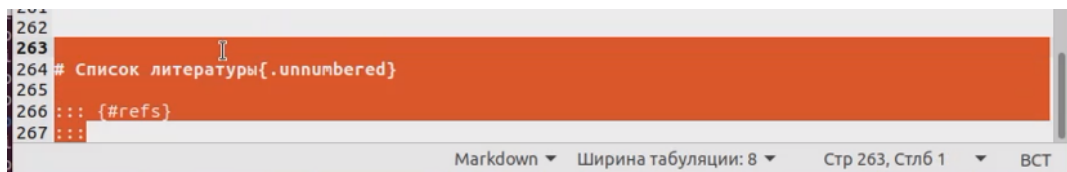


Рис. 4.10: Удаление списка литературы

3) Просмотр результата

- Титульный лист (рис. 4.11):

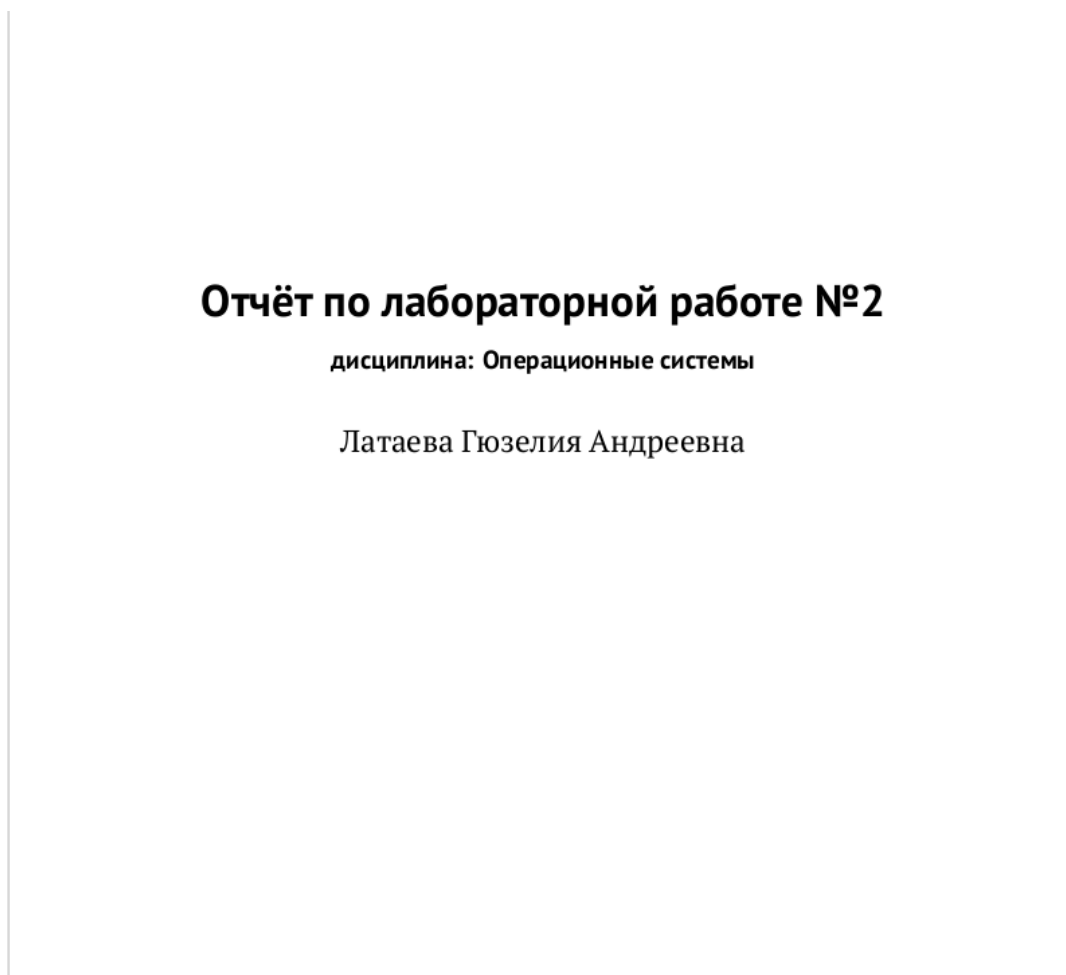


Рис. 4.11: Просмотр титульного листа

- Содержание (рис. 4.12):

Содержание

1	Цель работы	5
2	Последовательность выполнения работы	6
3	Заключение	15
4	Контрольные вопросы	16

Рис. 4.12: Просмотр содержания

- Список иллюстраций (рис. 4.13):

Список иллюстраций

2.1	Регистрация	6
2.2	Установка makescache	7
2.3	Установка epel_release	7
2.4	Установка gitflow	8
2.5	Установка config-manager	8
2.6	Установка репозитория с gh	8
2.7	Установка gh	8
2.8	Настройка git	9
2.9	Начальная ветка master	9
2.10	Параметры autocrlf и safecrlf	9
2.11	Алгоритм rsa	9
2.12	Алгоритм ed25519	10
2.13	Создание ключа	11
2.14	Список ключей	12
2.15	Список ключей	12
2.16	Добавление ключа GitHub	13
2.17	Настройка подписей	13
2.18	Создание репозитория	13
2.19	Копирование репозитория по шаблону	13
2.20	Переходим в каталог курса	14
2.21	Удаление лишних фалов и создание каталогов	14
2.22	Отправка данных на сервер	14

Рис. 4.13: Просмотр списка иллюстраций

- Цель работы (рис. 4.14):

1 Цель работы

Изучить идеологию и применение средств контроля версий. Освоить умения по работе с git.

Рис. 4.14: Просмотр цели работы

- Последовательность выполнения работ (рис. 4.15)

2 Последовательность выполнения работы

1) Создаю учётную запись на <https://github.com> и заполняю данные.

A screenshot of the GitHub registration interface. The background is dark blue. The text is white and light blue. The steps shown are: 1. 'Добро пожаловать на GitHub! Давайте начать приключение.' (Welcome to GitHub! Let's start the adventure.) 2. 'Введите свой адрес электронной почты,' (Enter your email address,) with a checkmark and 'lataeva-g@mail.ru'. 3. 'Создайте пароль' (Create a password) with a checkmark and '*****'. 4. 'Введите имя пользователя' (Enter your username) with a checkmark and 'galataeva'. 5. 'Хотите получать обновления и объявления о продуктах по электронной почте? Введите "у" для ответа "да" или "н" для ответа "нет"' (Do you want to receive updates and announcements about products by email? Enter 'u' for 'yes' or 'n' for 'no') with a checkmark and 'n'. 6. 'Подтвердите свою учетную запись' (Verify your account). Below this is a 'Проверка' (Verification) section: 'Пожалуйста, решите эту задачу, чтобы мы знали, что вы реальный человек.' (Please solve this task so we know you are a real person.) There is a text input field with the word 'Привет!' (Hello!) and a mouse cursor pointing at it.

Рис. 2.1: Регистрация

Рис. 4.15: Просмотр последовательности выполнения работ

- Заключение (рис. 4.16):

3 Заключение

Я изучила применение средств контроля версий и в какой-то мере освоила навыки по работе с git.

Рис. 4.16: Просмотр заключения

- Контрольные вопросы (рис. 4.17):

4 Контрольные вопросы

1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются?

Система контроля версий (от англ. Version Control System, VCS) — это, грубо говоря, место хранения кода. Она создана для разработки продуктов: на хранение кода, синхронизацию работы нескольких человек, создание релизов и т.д.

2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.

Хранилище (репозиторий) — это хранилище всех версий кода. Он бывает трех видов: Локальный — расположен на одном компьютере, и работать с ним может только один человек. Централизованный — расположен на сервере, куда имеют доступ сразу несколько программистов. Распределенный — самый удобный вариант с облачным хранилищем. Главный репозиторий хранится в облаке, а его локальные копии — у разработчиков на компьютерах.

Commit - запись изменений.

История - список предыдущих изменений.

Рабочая копия — текущее состояние файлов проекта, основанное на версии из хранилища (обычно на последней).

Рабочую копию получают из репозитория. Изменения вносятся в рабочую версию, потом посредством коммитов они заносятся в хранилище. История

Рис. 4.17: Просмотр контрольных вопросов

5 Выводы

Я научилась оформлять отчёты с помощью легковесного языка разметки Markdown

6 Список литературы

<https://ru.wikipedia.org/wiki/Markdown>