

Отчет по лабораторной работе №3

Дисциплина: Операционные системы

Астраханцева Анастасия Александровна

Содержание

1	Цель работы	4
2	Задание	5
3	Теоретическое введение	6
4	Выполнение лабораторной работы	7
5	Выводы	10

Список иллюстраций

4.1	Переход в нужный каталог	7
4.2	gedit	8
4.3	make для конвентирования в формат pdf и docx	8
4.4	Ответы на контрольные вопросы	9

1 Цель работы

Научиться оформлять отчёты с помощью легковесного языка разметки Markdown

2 Задание

- Выполнить отчёт по предыдущей лабораторной работе в формате Markdown.

3 Теоретическое введение

Чтобы задать заголовок, можно использовать "#"

Чтобы задать полужирное **начертание**, можно использовать двойные звездочки

****полужирное начертание****

Чтобы задать курсивное *начертание*, можно использовать одинарные звездочки

курсивное начертание

Чтобы задать курсивное полужирное ***начертание***, можно использовать тройные звездочки

*****курсивное полужирное начертание*****

Блоку цитирования можно задать с помощью символа >

Список можно создать с помощью звездочек * или тире -

- элемент списка 1 - элемент списка 2 - элемент списка 3

Можно вкладывать список в другой список:

' - элемент списка 1 - элемент списка А - элемент списка Б - элемент списка 2

' **Обработка файлов в формате Markdown**

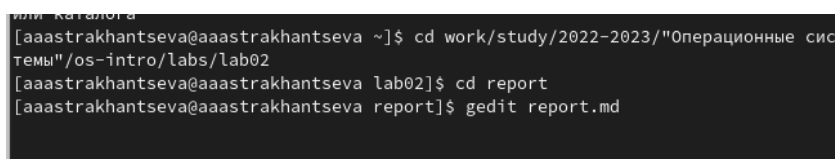
Для обработки файлов в формате Markdown будем использовать Pandoc. Конкретно, нам понадобится программа pandoc , pandoc-citeproc, pandoc-crossref. Мы будем использовать Makefile, с помощью которого наш файл в формате Markdown сконвертируется в формат pdf и docx.

4 Выполнение лабораторной работы

Для начала переходим в соответствующий каталог, содержащий нужный нам файл “report.md”. Для этого в термине прописываем команду

```
cd work/study/2022-2023/"Операционные системы"/os-intro/labs/lab02
```

(рис. 4.1).



```
или каталога  
[aaastrakhantseva@aaastrakhantseva ~]$ cd work/study/2022-2023/"Операционные сис  
темы"/os-intro/labs/lab02  
[aaastrakhantseva@aaastrakhantseva lab02]$ cd report  
[aaastrakhantseva@aaastrakhantseva report]$ gedit report.md
```

Рис. 4.1: Переход в нужный каталог

После этого нам нужно перейти в файловый редактор, “gedit”, например: (рис. 4.2).

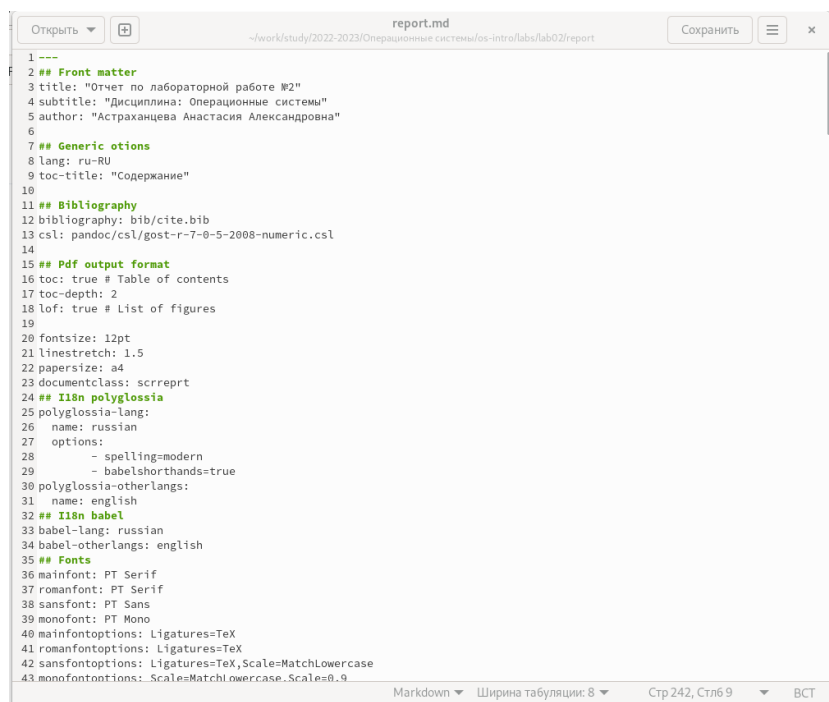


Рис. 4.2: gedit

Далее, после того, как отчет выполнен, нам необходимо сохранить изменения в файле и закрыть окно gedit. После этого в том же окне терминала, с помощью которого мы переходили в gedit, нужно прописать команду make. (рис. 4.3)

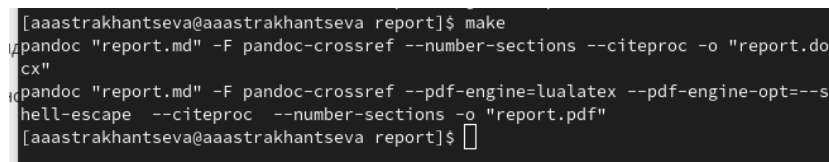


Рис. 4.3: make для конвентирования в формат pdf и docx

После этого сконвентуруется наш отчет в формате pdf и docx.

В выполнении отчета по ЛР №2 так же присутствуют ответы на контрольные вопросы. На данном скриншоте видно то, как оформляется добавление изображений. (рис. 4.4)


```
"Создание каталогов "операционная система" в каталоге "*/netx/1000/2012-2013" (Fig. 104);

[[Создание каталогов "операционные системы"] (Image 1.png) (Fig. 105 k105b-790)

Создание reposit0000 (Fig. 106);

[[Создание reposit0000] (Image 11.png) (Fig. 109 k109b-790)

Далее переходим в каталог курса "is-intro". Удаляем лишние файлы, создаем необходимые каталоги, отправляем файлы на сервер (Fig. 110 - Fig. 111).

[[Перемещение файлов, создание каталогов] (Image 12.png) (Fig. 110 k110b-790)

[[Отправка файлов на сервер] (Image 13.png) (Fig. 111 k111b-790)

[[Отправка файлов на сервер] (Image 14.png) (Fig. 112 k112b-790)

«Контрольные вопросы»

1. Что такое системы контроля версий (СКВ) и для решения каких задач они предназначены?
Системы контроля версий (Version Control System, VCS) - программное обеспечение для облегчения работы с изменяющейся информацией.

Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Также объединять (слиять) изменения, сделанные разными участниками (автоматически или вручную), вручную выбирать конкретные версии, отменять изменениями файлы или записывать файлы для изменений.

Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удаленном репозитории, к которому настроен доступ для участников проекта.

2. Объясните следующие понятия VCS и их отклонения: кранчи, commit, история, рабочая копия.

Reposit0000 - кранчи версий - в нем хранятся все документы вместе с историей их изменений и другой служебной информацией.

Кранчи - это содержимое скрытой папки .git. В этой папке хранятся все версии рабочей области и служебная информация. Этим версиям система автоматически даёт название, состоящее из букв и цифр.

Рабочая копия - копия проекта, связанная с reposit0000

commit - это команда, которая делает так, что новая версия проекта сохраняется и добавляется в кранчи. В файле с сохранением отображаются: все изменения, которые происходили в рабочей области, автор изменений и краткий комментарий, описывающий суть изменений. Каждый commit хранит полное состояние рабочей области, её папки и файлы проекта.

История хранит все изменения проекта и позволяет при необходимости обратиться к ним.
```

Рис. 4.4: Ответы на контрольные вопросы

5 Выводы

В ходе выполнения лабораторной работы №3 мною были получены навыки оформления отчёта с помощью легковесного языка разметки Markdown.