

Отчёт по лабораторной работе №3

Дисциплина: Операционные системы

Панина Жанна Валерьевна

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	8
3.1	Создание отчёта по предыдущей лабораторной работе	8
3.2	Генерация отчёта в терминале	9
4	Выводы	11
	Список литературы	11

Список иллюстраций

3.1	Цель работы, задание и теоретическое введение	8
3.2	Основная часть: выполнение лабораторной работы	9
3.3	Заключительная часть: ответы на контрольные вопросы	9
3.4	Вывод	9
3.5	Команда make	10

Список таблиц

1 Цель работы

Научиться оформлять отчёты с помощью легковесного языка разметки Markdown.

2 Теоретическое введение

Чтобы создать заголовок, используется знак (#). Чтобы задать для текста полужирное начертание, нужно заключить его в двойные звездочки. Чтобы задать для текста курсивное начертание, нужно заключить его в одинарные звездочки. Чтобы задать для текста полужирное и курсивное начертание, заключаем его в тройные звездочки. Блоки цитирования создаются с помощью символа >. Неупорядоченный (маркированный) список можно отформатировать с помощью звездочек или тире. Чтобы вложить один список в другой, добавьте отступ для элементов дочернего списка. Упорядоченный список можно отформатировать с помощью соответствующих цифр. Чтобы вложить один список в другой, нужно добавить отступ для элементов дочернего списка. Синтаксис Markdown для встроенной ссылки состоит из части [link text] , представляющей текст гиперссылки, и части (file-name.md) – URL-адреса или имени файла, на который дается ссылка. Markdown поддерживает как встраивание фрагментов кода в предложение, так и их размещение между предложениями в виде отдельных огражденных блоков. Огражденные блоки кода — это простой способ выделить синтаксис для фрагментов кода. Внутритекстовые формулы делаются аналогично формулам LaTeX. Для обработки файлов в формате Markdown будем использовать Pandoc. Конкретно, нам понадобится программа pandoc, [pandoc-citeproc https://github.com/jgm/pandoc/releases](https://github.com/jgm/pandoc/releases), [pandoc-crossref https://github.com/lierdakil/pandoc-crossref/releases](https://github.com/lierdakil/pandoc-crossref/releases). Преобразовать файл README.md можно следующим образом: 1 pandoc README.md -o README.pdf или так 1 pandoc README.md -o README.docx Можно использовать следующий

Makefile 1 FILES = \$(patsubst %.md, %.docx, \$(wildcard .md)) 2 FILES += \$(patsubst
%.md, %.pdf, \$(wildcard .md))

3 Выполнение лабораторной работы

3.1 Создание отчёта по предыдущей лабораторной работе

1. Открываю шаблон отчёта формата .md, меняю в нём ФИО автора, изменяю название, добавляю свои данные (рис. ??).

```
---  
## Front matter  
title: "Лабораторная работа №2"  
subtitle: "Дисциплина: Операционные системы"  
author: "Панина Жанна Валерьевна"  
  
## Generic options  
lang: ru-RU  
toc-title: "Содержание"  
  
## Bibliography  
bibliography: bib/cite.bib  
csl: pandoc/csl/gost-r-7-0-5-2008-numeric.csl
```

(рис. 3.1).

2. Пишу цель работы, задание и теоретическое введение.

```
# Цель работы  
Цель работы – изучить идеологию и применения средств контроля версий, освоить умения по работе с git.  
  
# Задание  
1. Создать базовую конфигурацию для работы с git.  
2. Зарегистрироваться на github.  
3. Создать ключ SSH.  
4. Создать ключ GPG.  
5. Настроить подписи git.  
6. Создать локальный каталог для выполнения заданий по предмету.  
  
# Теоретическое введение  
Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное де  
удалённом депозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля  
совмещать изменения, произведённые разными участниками проекта, производить откат к любой более ранней версии проекта, если это требуе  
версий используется централизованная модель, предполагающая наличие единого депозитория для хранения файлов. Выполнение большинства фу  
осуществляется специальным сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нуж
```

Рис. 3.1: Цель работы, задание и теоретическое введение

3. Заполняю основную часть работы: прописываю шаги выполнения работы, добавляю ссылки на иллюстрации и подписи к ним (рис. 3.2).

```
# Выполнение лабораторной работы

## Создание ключа SSH

Поскольку я уже была зарегистрирована на GitHub в первом семестре, начинаю работу с создания ключа SSH (рис. [-@fig:001]).

![[Создание ключа SSH](image/1.png){#fig:001 width=70%}]

## Создание ключа GPG

1. Создаю ключ GPG, указав необходимые параметры (рис. [-@fig:002]).

![[Создание ключа GPG](image/2.png){#fig:002 width=70%}]

2. Вывожу список ключей и копирую отпечаток приватного ключа. Копирую сгенерированный ключ GPG (рис. [-@fig:003]).

![[Ключ GPG](image/3.png){#fig:003 width=70%}]
```

Рис. 3.2: Основная часть: выполнение лабораторной работы

4. Отвечаю на контрольные вопросы к лабораторной работе (рис. 3.3).

```
# Ответы на контрольные вопросы

1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначены?
- Система контроля версий – программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями позволяет того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое.
(Version Control System, VCS) применяются для:
- Хранение полной истории изменений
- причин всех производимых изменений
- Откат изменений, если что-то пошло не так
- Поиск причины и ответственного за появления ошибок в программе
- Совместная работа группы над одним проектом
- Возможность изменять код, не мешая работе других пользователей

2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия
- Репозиторий – хранилище версий – в нем хранятся все документы вместе с историей их изменения и другой служебной информацией.
- commit – отслеживание изменений
- Рабочая копия – копия проекта, связанная с репозиторием (текущее состояние файлов проекта, основанное на версии из хранилища (обычно на основе последней версии)).
- История хранит все изменения в проекте и позволяет при необходимости обратиться к нужным данным.
```

Рис. 3.3: Заключительная часть: ответы на контрольные вопросы

5. Прописываю вывод о работе (рис. 3.4).

```
# Выводы

Во время выполнения работы я изучила идеологию и применения средств контроля версий, освоила умения по работе с git.
```

Рис. 3.4: Вывод

3.2 Генерация отчёта в терминале

С помощью команды make создаю отчёт в форматах .docx и .pdf (рис. 3.5).

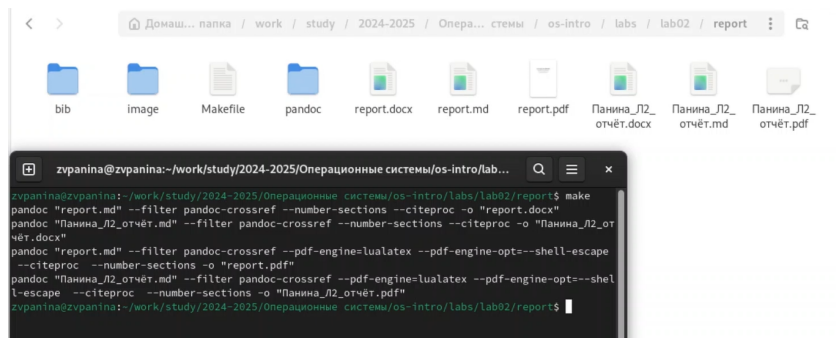


Рис. 3.5: Команда make

4 Выводы

В ходе выполнения лабораторной работы я научилась оформлять отчёты с помощью легковесного языка разметки Markdown.

Список литературы

Руководство по оформлению Markdown файлов. [Электронный ресурс]. GitHub
Gist URL: <https://gist.github.com/Jekins/2bf2d0638163f1294637>