

Отчёт по лабораторной работе №3

Markdown

Аскеров Александр Эдуардович

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
4	Выводы	11

Список иллюстраций

3.1	Заполнение титульного листа	6
3.2	Указание цели работы	6
3.3	Указание заданий	7
3.4	Заполнение раздела с выполнением лабораторной работы	8
3.5	Заполнение ответов на контрольные вопросы	9
3.6	Указание вывода	9
3.7	Отчёт в разных форматах	9
3.8	Архив с отчётом в разных форматах и скриншотами	10

1 Цель работы

Научиться оформлять отчёты с помощью легковесного языка разметки Markdown.

2 Задание

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

3 Выполнение лабораторной работы

Открываем файл report.md и указываем номер лабораторной работы, её название и имя автора.

```
## Front matter
title: "Отчёт по лабораторной работе №2"
subtitle: "Первоначальная настройка git"
author: "Аскеров Александр Эдуардович"
```

Рис. 3.1: Заполнение титульного листа

Указываем цель работы.

```
# Цель работы

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

Рис. 3.2: Указание цели работы

Указываем задачи лабораторной работы.

Задание

- ★ Создать базовую конфигурацию для работы с git
- ★ Создать ключ SSH
- ★ Создать ключ PGP
- ★ Настроить подписи git
- ★ Зарегистрироваться на Github
- ★ Создать локальный каталог для выполнения заданий по предмету

Рис. 3.3: Указание заданий

Заполняем раздел Выполнение лабораторной работы, указываем заголовки и добавляем скриншоты с поясняющими подписями.

Выполнение лабораторной работы

Базовая настройка `git`

Сначала сделаем предварительную конфигурацию `git`. Откроем терминал и введём следующие команды, указав имя и `email` владельца репозитория.

! [Создаём предварительную конфигурацию `git`] ([image/1.png](#)) { #fig:1 }

Настроим `utf-8` в выводе сообщений `git`.

! [Настраиваем `utf-8` в выводе сообщений `git`] ([image/2.png](#)) { #fig:2 }

Зададим имя начальной ветки (будем называть её `master`).

! [Задаём имя начальной ветки] ([image/3.png](#)) { #fig:3 }

Параметр `autocrlf`.

! [Параметр `autocrlf`] ([image/4.png](#)) { #fig:4 }

Параметр `safecrlf`.

! [Параметр `safecrlf`] ([image/5.png](#)) { #fig:5 }

Рис. 3.4: Заполнение раздела с выполнением лабораторной работы

Записываем ответы на контрольные вопросы.

Контрольные вопросы

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

Системы контроля версий (VCS) – это программные инструменты, которые помогают управлять изменениями исходного кода с течением времени. Они позволяют разработчикам отслеживать изменения, возвращаться к предыдущим версиям и сотрудничать с другими разработчиками. Они также позволяют хранить исходный код и делиться им с другими.

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

Хранилище – это центральное место, где хранится и управляется исходный код. Это основная область хранения для всех версий проекта.

Рис. 3.5: Заполнение ответов на контрольные вопросы

Указываем вывод к лабораторной работе.

Выводы

Изучены идеология и применение средств контроля версий и приобретены умения по работе с git.

Рис. 3.6: Указание вывода

После заполнения отчёта в файле report.md вводим в консоли команду make, которая формирует отчёт в форматах docx и pdf.



report.docx



report.md



report.pdf

Рис. 3.7: Отчёт в разных форматах

Создаём архив с отчётом в разных форматах и скриншотами.



Рис. 3.8: Архив с отчётом в разных форматах и скриншотами

4 Выводы

Приобретены навыки оформления отчётов с помощью легковесного языка разметки Markdown.