

Лабораторная работа №3

Дисциплина: Архитектура компьютера

Орлов Илья Сергеевич

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	Выводы	11

Список иллюстраций

4.1	Редактирование отчета	9
4.2	Заполнение основной информации	9
4.3	Цель работы, задание и теоретическое введение	9
4.4	Выполнение отчета по лабораторной работе	10

Список таблиц

1 Цель работы

Научиться оформлять отчёты с помощью легковесного языка разметки Markdown.

2 Задание

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

3 Теоретическое введение

Чтобы создать заголовок, используйте знак (#). Чтобы задать для текста полужирное начертание, заключите его в двойные звездочки. Чтобы задать для текста курсивное начертание, заключите его в одинарные звездочки. Чтобы задать для текста полужирное и курсивное начертание, заключите 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, добавьте 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))*

4 Выполнение лабораторной работы

В рабочей директории курса открываю через текстовый редактор файл с шаблоном отчета. (рис. 4.1)

```
isorlov@isorlov:~/work/study/2024-2025/os/os-intro/labs/lab02/report$ gedit report.md
```

Рис. 4.1: Редактирование отчета

Указываю основную информацию о лабораторной работе. (рис. 4.2)

```
1 ---
2 ## Front matter
3 title: "Лабораторная работа №2"
4 subtitle: "Дисциплина: Архитектура компьютера"
5 author: "Орлов Илья Сергеевич"
6
```

Рис. 4.2: Заполнение основной информации

Формирую цель работы, задание и заполняю теоретическое введение. (рис. 4.3)

```
71 # Цель работы
72
73 Целью данной работы является изучение идеологии и применения средств контроля версий и освоение умения по
74 работе с git.
75
76 # Задание
77
78 - базовую конфигурацию для работы с git.
79 - ключ SSH.
80 - ключ PGP.
81 - подписи git.
82 - на Github.
83 - локальный каталог для выполнения заданий по предмету.
84
85 # Теоретическое введение
86
87 Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним
88 проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен
89 доступ для участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет
90 их фиксировать, совмещать изменения, произведённые разными участниками проекта, производить откат к любой
91 более ранней версии проекта, если это требуется.
92
93 В классических системах контроля версий используется централизованная модель, предполагающая наличие единого
94 репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется
95 специальным сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд
96 получает нужную ему версию файлов. После внесения изменений, пользователь размещает новую версию в хранилище.
97 При этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент.
98 Сервер может сохранять не полную версию изменённых файлов, а производить так называемую дельта-компрессию –
99 сохранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных.
```

Рис. 4.3: Цель работы, задание и теоретическое введение

Описываю процесс выполнения лабораторной работы. (рис. 4.4)

```
91 # Выполнение лабораторной работы
92
93 Произвожу базовую настройку git. (рис. -@fig:001)
94
95 ![Пример конфигурации git](image/1.png){#fig:001 width=70%}
96
97 Создаю ssh и gpg ключи. (рис. -@fig:002)
98
99 ![Генерация ключей](image/2.png){#fig:002 width=70%}
100
101 Экспортирую gpg ключ для авторизации на github. (рис. -@fig:003)
102
103 ![Экспорт ключа](image/3.png){#fig:003 width=70%}
104
105 Настраиваю автоматические подписи для коммитов. (рис. -@fig:004)
106
107 ![Конфигурация подписей для коммитов](image/4.png){#fig:004 width=70%}
108
109 Авторизуюсь на github для работы через терминал. (рис. -@fig:005)
110
111 ![Авторизация на github](image/5.png){#fig:005 width=70%}
112
113 Создаю директорию курса по шаблону (рис. -@fig:006)
114
115 ![Создание директории курса](image/6.png){#fig:006 width=70%}
116
117 Настраиваю рабочую директорию (рис. -@fig:007)
118
119 ![Настройка директории](image/7.png){#fig:007 width=70%}
```

Рис. 4.4: Выполнение отчета по лабораторной работе

5 Выводы

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