Отчёт по лабораторной работе №3

Дисциплина: архитектура компьютера

Закиров Нурислам Дамирович

Содержание

# 1 Цель работы

Цель данной лабораторной работы состоит в том, чтобы мы научились составлять отчеты с помощью легковесного языка разметки Markdown.

# 2 Задание

1. Установка необходимого ПО
2. Заполнение отчета по выполнению лабораторной работы №4 с помощью языка разметки Markdown
3. Задание для самостоятельной работы

# 3 Теоретическое введение

Markdown - легковесный 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. Внутритекстовые формулы делаются аналогично формулам LaTeX. В Markdown вставить изображение в документ можно с помощью непосредственного указания адреса изображения. 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.

# 4 Выполнение лабораторной работы

## 4.1 Установление необходимого ПО

### 4.1.1 Установка TexLive

Я скачал TexLive с официального сайта, после чего распаковал архив (рис. [1](#fig%3A001))

Figure 1: Распаковка архива TexLive

Рисунок 1: Распаковка архива TexLive

Я использую cd, чтобы перейти в распакованную папку. Запускаю скрипт install-tl-\* с правами root. (рис. [2](#fig%3A002))

Figure 2: Запуск скрипта

Рисунок. 2: Запуск скрипта

Добавляю /usr/local/texlive/2023/bin/x86\_64-linux в свой PATH(рис. [3](#fig%3A003)).

Figure 3: Добавление в PATH

Рисунок. 3: Добавление в PATH

### 4.1.2 Установка pandoc и pandoc-crossref

Скачиваю архив pandoc версии 3.1.8. (рис. [4](#fig%3A004)).

Figure 4: Скачивание pandoc

Рисунок. 4: Скачивание pandoc

Скачиваю архив pandoc-crossref 0.3.13.0 (рис. -[5](#fig%3A005)).

Figure 5: Скачивание pandoc-crossref

Рисунок. 5: Скачивание pandoc-crossref

Распаковываю архивы (рис. [6](#fig%3A006)).

Figure 6: Распаковка архивов

Рисунок. 6: Распаковка архивов

Копирую файлы pandoc и pandoc-crossref в каталог /usr/local/bin/ с правами администатора(рис. [7](#fig%3A007)).

Figure 7: Копирование каталогов в другую директорию

Рисунок. 7: Копирование каталогов в другую директорию

Проверяем правильность выполненных действий (рис. [8](#fig%3A008)).

Figure 8: Проверка правильности выполнения команды

Рисунок. 8: Проверка правильности выполнения команды

## 4.2 Заполнение отчета по выполнению лабораторной работы №4 с помощью языка разметки Markdown

Открываю командныю строку. Перехожу в каталог курса(рис. [9](#fig%3A009)).

Figure 9: Перемещение между директориями

Рисунок. 9: Перемещение между директориями

С помощью команды git pull я обновляю локальный репозиторий, скачав изменения из удаленного репозитория. (рис. [10](#fig%3A010)).

Figure 10: Обновление локального репозитория

Рисунок. 10: Обновление локального репозитория

Перемещаюсь в каталог с шаблоном отчета по лабораторной работе №3 (рис. [11](#fig%3A011)).

Figure 11: Перемещение между директориями

Рисунок. 11: Перемещение между директориями

Ввводя команду make, компилирую шаблон с использованием Makefile. (рис. [12](#fig%3A012)).

Figure 12: Компиляция шаблона

Рисунок. 12: Компиляция шаблона

Открываю сгенерированный файл report.docx через LibreOffice (рис. [13](#fig%3A013)).

Figure 13: Открытие файла docx

Рисунок. 13: Открытие файла docx

Открываю сгенерированный файл report.pdf (рис. [14](#fig%3A014)). Теперь мы уверены в правильности компиляции.

Figure 14: Открытие файла pdf

Рисунок. 14: Открытие файла pdf

Удаляю полученные файлы, вводя команду make clean (рис. [15](#fig%3A015)). С помощью команды ls убеждаемся в том, что созданные файлы удалились.

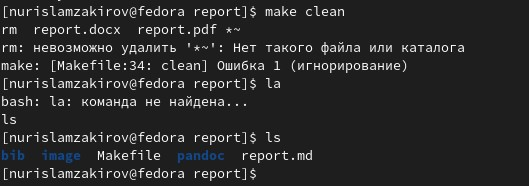


Рисунок. 15: Удаление файлов

Открываем файл report.md с помощью текстового редактора mousepad (рис. [16](#fig%3A016)).

Figure 16: Открытие файла

Рисунок. 16: Открытие файла

Копируем файл с новым названием с помощью утилиты cp (рис. [17](#fig%3A017)).

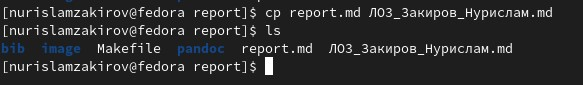


Рисунок. 17: Копирование файла с новым именем

Приступаю к заполнению отчет с помощью языка разметки Markdown. (рис. [18](#fig%3A018)).

Figure 18: Заполнение отчета

Рисунок. 18: Заполнение отчета

Компилирую файл с отчетом, при помощи make. Загружаю отчет на GitHub.

## 4.3 Задание для самостоятельной работы

1. Перехожу в директорию lab02/report с помощью cd для того, чтобы там заполнять отчет по прошлой лабораторной работе (рис. [19](#fig%3A019)).

Figure 19: Перемещение между директориями

Рисунок. 19: Перемещение между директориями

Как мы можем заметить, я уже создал файл для отчета в формате Markdown. Открываю файл с помощью текстового редактора и начинаю заполнять отчет (рис. [20](#fig%3A020)).

Figure 20: Работа над отчетом

Рисунок. 20: Работа над отчетом

Предыдущий файл отчета был удален, чтобы при компиляции он мне не мешал. Компилирую файл с отчетом по лабораторной работе (рис. [21](#fig%3A021)).

Figure 21: Компиляция файлов

Рисунок. 21: Компиляция файлов

1. Удаляем лсгенерированные файлы report.docx и report.pdf, ведь они нам не понадобятся ([22](#fig%3A022)).

Figure 22: Удаление лишних файлов

Рисунок. 22: Удаление лишних файлов

Я добавляю изменения на GitHub с помощью комнады git add и сохраняю эти изменения с помощью commit. ([23](#fig%3A023)).

Figure 23: Добавление файлов на GitHub

Рисунок. 23: Добавление файлов на GitHub

Отправлялю файлы на сервер с помощью команды git pull ([24](#fig%3A024)).

Figure 24: Отправка файлов

Рисунок. 24: Отправка файлов

# 5 Выводы

В результате выполнения данной лабораторной работы я научился составлять отчеты с помощью легковесного языка разметки Markdown.

# 6 Список литературы

1. [Архитектура ЭВМ](https://esystem.rudn.ru/pluginfile.php/1584625/mod_resource/content/1/Лабораторная работа №4.pdf)