

Лабоаторная работа №4. Язык Markdown

Дисциплина - Архитектура Компьютера

Шевырев Иван

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Откроем терминал	6
2.2	Перейдем в каталог созданный во время выполнения 3й лабораторной	6
2.3	Обновим локальный репозиторий с помощью <code>git pull</code>	7
2.4	Перейдем в каталог с 4 лабораторной	7
2.5	исполним команду <code>make</code>	7
2.6	Проверим новые файлы	7
2.7	Удалим полученные с использованием Makefile файлы	10
2.8	Откроем <code>report.md</code> с помощью текстового редактора	11
2.9	Заполним отчет 4ой лабораторной	11
2.10	Загрузим Файлы на GitHub	12
3	Задания для самостоятельной работы	14
3.1	Сделать отчет по 3й лабораторной работе	14
3.1.1	Оформим содержимое 3й лабораторной в формате Markdown	14
3.2	Скомпилируем отчет по 3 лабораторной	14
3.3	Загрузим отчет на GitHub	16
4	Выводы	17

Список иллюстраций

2.1	Открытие терминала	6
2.2	Переход в каталог курса	6
2.3	Вывод команды make	7
2.4	Переход в каталог 4ой лабораторной	7
2.5	make	7
2.6	Просмотр сгенерированного PDF Файла	8
2.7	Просмотр сгенерированного PDF Файла 2	9
2.8	Просмотр DOCX файла через libreoffice	10
2.9	Вывод команды make	10
2.10	Вывод команды ls	11
2.11	Просмотр файла report.md	11
2.12	Отчет 4ой лабораторной	12
2.13	Выгрузка на GitHub	13
3.1	Заполнение отчета для 3 лабораторной	14
3.2	Выполнение команды make	15
3.3	просмотр report.pdf	15

Список таблиц

1 Цель работы

Освоение процедуры оформления отчетов с помощью легковесного языка разметки Markdown.

2 Выполнение лабораторной работы

2.1 Откроем терминал

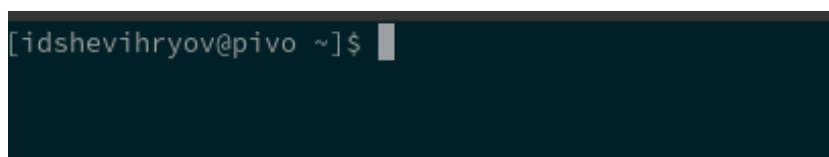


Рис. 2.1: Открытие терминала

2.2 Перейдем в каталог созданный во время выполнения 3й лабораторной

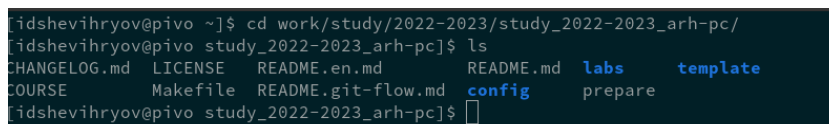


Рис. 2.2: Переход в каталог курса

2.3 Обновим локальной репозиторий с помощью git pull

```
[idshevihryov@pivo study_2022-2023_arh-pc]$ git pull
Already up to date.
[idshevihryov@pivo study_2022-2023_arh-pc]$
```

Рис. 2.3: Вывод команды make

2.4 Перейдем в каталог с 4 лабораторной

```
[idshevihryov@pivo study_2022-2023_arh-pc]$ cd labs/lab04/report/
[idshevihryov@pivo report]$
```

Рис. 2.4: Переход в каталог 4ой лабораторной

2.5 исполним команду make

Запустим make чтобы выполнить команды из Makefile

```
[idshevihryov@pivo report]$ make
pandoc "report.md" -F pandoc-crossref --number-sections --citeproc -o "report.docx"
pandoc "report.md" -F pandoc-crossref --pdf-engine=lualatex --pdf-engine-opt=--shell-escape
--citeproc --number-sections -o "report.pdf"
[idshevihryov@pivo report]$ ls
Makefile  bib  image  pandoc  report.docx  report.md  report.pdf
[idshevihryov@pivo report]$
```

Рис. 2.5: make

2.6 Проверим новые файлы

Откроем report.md и report.pdf

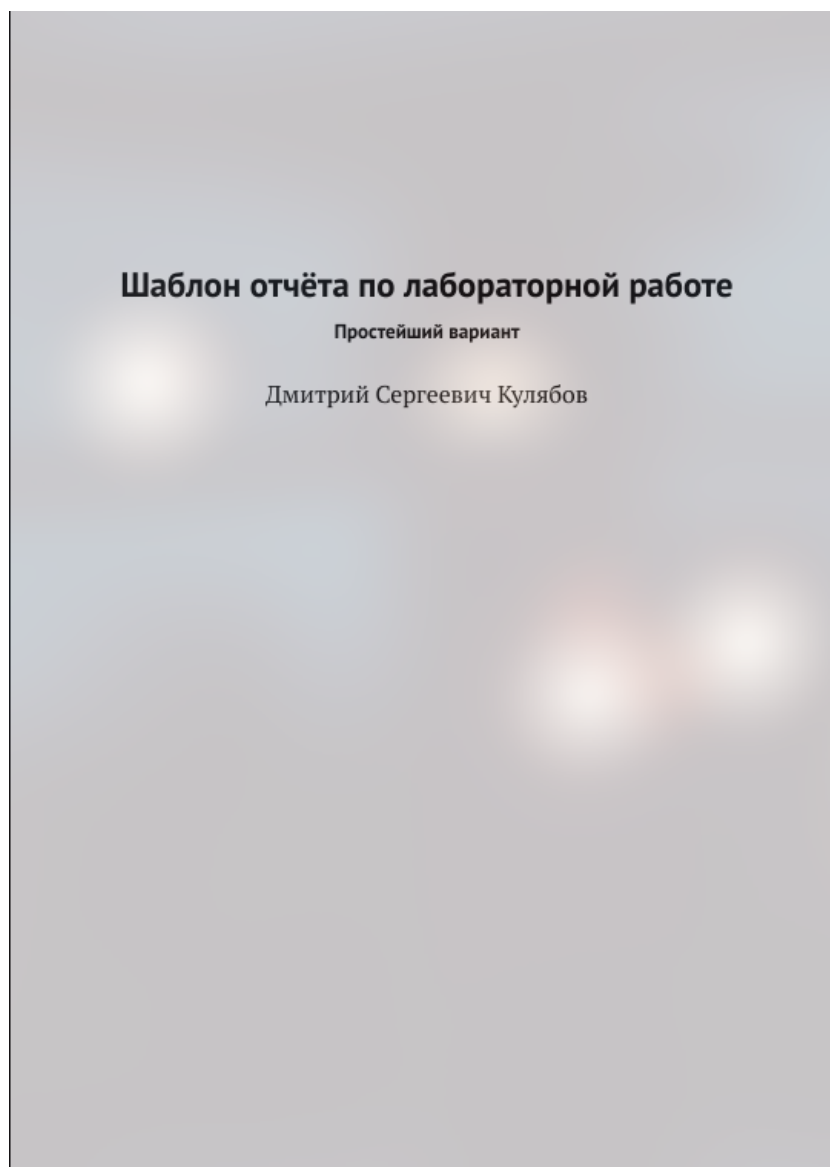


Рис. 2.6: Просмотр сгенерированного PDF Файла

Содержание		
1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	9
	Список литературы	10

Рис. 2.7: Просмотр сгенерированного PDF Файла 2

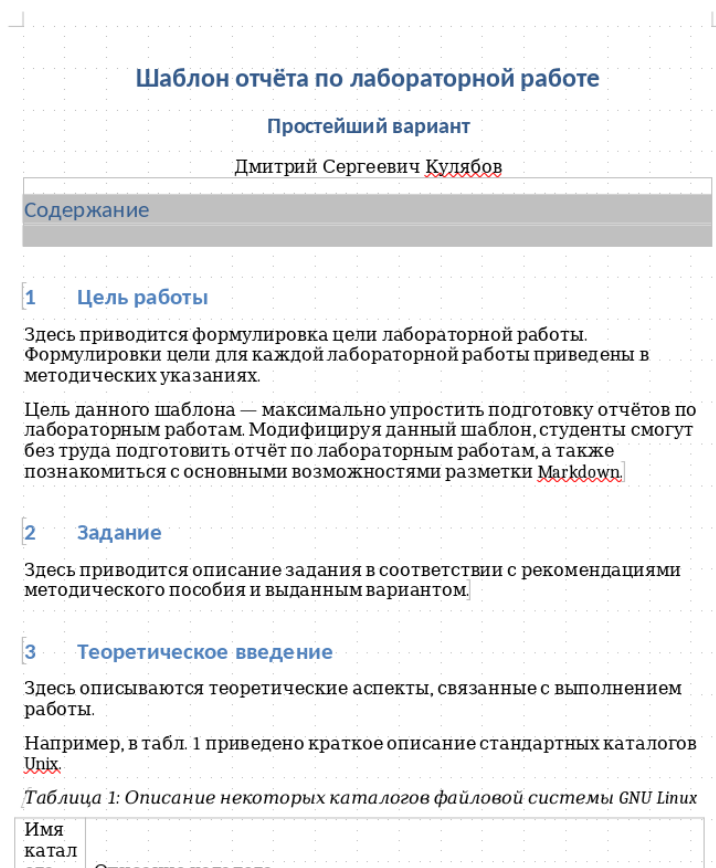


Рис. 2.8: Просмотр DOCX файла через libreoffice

2.7 Удалим полученные с использованием Makefile файлы

Исполним команду `make clean`

```
[idshevihryov@pivo report]$ make clean
rm report.docx report.pdf *~
rm: cannot remove '*~': No such file or directory
make: [Makefile:26: clean] Error 1 (ignored)
[idshevihryov@pivo report]$ ls
Makefile bib image pandoc report.md
[idshevihryov@pivo report]$
```

Рис. 2.9: Вывод команды `make`

Убедимся, что файлы удалены с помощью `ls`

```
[idshevi@pivo report]$ ls
Makefile  bib  image  pandoc  report.md
[idshevi@pivo report]$
```

Рис. 2.10: Вывод команды `ls`

2.8 Откроем `report.md` с помощью текстового редактора

```
1 ---
2 ## Front matter
3 title: "Шаблон отчёта по лабораторной работе"
4 subtitle: "Простейший вариант"
5 author: "Дмитрий Сергеевич Кулябов"
6
7 ## Generic options
8 lang: ru-RU
9 toc-title: "Содержание"
10
11 ## Bibliography
12 bibliography: bib/cite.bib
13 csl: pandoc/csl/gost-r-7-0-5-2008-numeric.csl
14
15 ## Pdf output format
16 toc: true # Table of contents
17 toc-depth: 2
18 lof: true # List of figures
19 lot: true # List of tables
20 fontsize: 12pt
21 linespacing: 1.5
22 papersize: a4
23 documentclass: scrreprt
24 ## I18n polyglossia
25 polyglossia-lang:
26   name: russian
27   options:
28     - spelling=modern
29     - babelshorthands=true
30 polyglossia-otherlangs:
31   name: english
32 ## I18n babel
33 babel-lang: russian
34 babel-otherlangs: english
35 ## Fonts
36 mainfont: PT Serif
37 romanfont: PT Serif
38 sansfont: PT Sans
39 monofont: PT Mono
40 mainfontoptions: Ligatures=TeX
41 romanfontoptions: Ligatures=TeX
42 sansfontoptions: Ligatures=TeX,Scale=MatchLowercase
43 monofontoptions: Scale=MatchLowercase,Scale=0.9
44 ## Biblatex
45 biblatex: true
46 biblio-style: "gost-numeric"
47 biblatexoptions:
48   - parenttracker=true
49   - backend=biber
50   - hyperref=auto
51   - language=auto
52   - autolanguage=auto
```

Рис. 2.11: Просмотр файла `report.md`

Мы видим вспомогательную информацию в начале файла и основную часть документа: заголовки, вставка изображений с подписями и пример таблицы.

2.9 Заполним отчет 4ой лабораторной

Сделаем 4ю лабораторную и составим ее отчет в markdown

```

69 # Цель работы
70
71     Освоить процедуры компиляции и сборки программ, написанных на ассемблере NASM.
72
73 # Выполнение лабораторной работы
74
75 ## Программа "Hello world!"
76
77 ### Создадим файл `hello.asm`
78
79 ### Создадим каталок lab05
80
81 ![Создание директории](image/1.png){ #fig:001 width=70% }
82
83 ### Создадим текстовый файл lab05.asm
84
85 ![Создание lab05.asm и команда ls](image/2.png){ #fig:002 width=70% }
86
87 ### Откроем файл через Gedit
88
89 ![Открытие файла через текстовый редактор gedit](image/3.png){ #fig:003 width=70% }
90
91 ### Введем код в файл
92
93 ![Код введенный в файл](image/4.png){ #fig:004 width=70% }
94
95 ### Воспользуемся транслятором NASM
96
97 Скомпилируем вышенаписанную программу с помощью команды `nasm -f elf hello.asm`
98
99 ![Трансляция кода](image/5.png){ #fig:005 width=70% }
100
101
102 Создадим файл с другим именем используя `-o` и создадим листинг, с помощью `-l`
103
104 ![Создание объектного файла с другим именем](image/6.png){ #fig:006 width=70% }
105
106 ### Воспользуемся компоновщиком LD
107
108 Выполним команду `ld -m elf_i386 hello.o -o hello`
109
110 ![Компоновщик `ld`](image/7.png){ #fig:007 width=70% }
111
112 С помощью команды `ls`, увидим что файл hello создался
113
114
115 ### Скомпируем в файл с именем `main` с помощью ключа `-o`
116
117
118 ![Исполняемый файл с именем main](image/8.png){ #fig:008 width=70% }
119
120
121 ### Запустим исполняемый файл
122
123 ![Запуск `./hello`](image/9.png){ #fig:009 width=70% }

```

Рис. 2.12: Отчет 4ой лабораторной

2.10 Загрузим Файлы на GitHub

Исполним команды `git add`, `git commit` и `git push`

```
[idshevihryov@pivo report]$ git add .
[idshevihryov@pivo report]$ git commit -am 'feat(main): add files lab-4'
[master f88c594] feat(main): add files lab-4
20 files changed, 94 insertions(+), 38 deletions(-)
create mode 100644 labs/lab04/report/image/1.png
create mode 100644 labs/lab04/report/image/10.png
create mode 100644 labs/lab04/report/image/11.png
create mode 100644 labs/lab04/report/image/12.png
create mode 100644 labs/lab04/report/image/13.png
create mode 100644 labs/lab04/report/image/14.png
create mode 100644 labs/lab04/report/image/15.png
create mode 100644 labs/lab04/report/image/16.png
create mode 100644 labs/lab04/report/image/17.png
create mode 100644 labs/lab04/report/image/2.png
create mode 100644 labs/lab04/report/image/3.png
create mode 100644 labs/lab04/report/image/4.png
create mode 100644 labs/lab04/report/image/5.png
create mode 100644 labs/lab04/report/image/6.png
create mode 100644 labs/lab04/report/image/7.png
create mode 100644 labs/lab04/report/image/8.png
create mode 100644 labs/lab04/report/image/9.png
create mode 100644 labs/lab04/report/report.docx
create mode 100644 labs/lab04/report/report.pdf
[idshevihryov@pivo report]$ git push
Enumerating objects: 32, done.
Counting objects: 100% (32/32), done.
Delta compression using up to 8 threads
Compressing objects: 100% (26/26), done.
Writing objects: 100% (26/26), 805.31 KiB | 4.28 MiB/s, done.
Total 26 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:IvanShevyrev/study_2022-2023_arh-pc.git
 256b855..f88c594 master -> master
[idshevihryov@pivo report]$
```

Рис. 2.13: Выгрузка на GitHub

3 Задания для самостоятельной работы

3.1 Сделать отчет по 3й лабораторной работе

3.1.1 Оформим содержимое 3й лабораторной в формате Markdown

```
10 ## Откроем терминал
79 ![[Открытие терминала]](image/1.png){#fig:001 width=70%}
80 ## Перейдем в каталог созданный во время выполнения 3й лабораторной
81 ![[Переход в каталог курса ]](image/2.png){#fig:002 width=70%}
82
83 ## Обновим локальный репозиторий с помощью 'git pull'
84 ![[Выход команды make]](image/3.png){#fig:003 width=70%}
85
86 ## Перейдем в каталог с 4 лабораторной
87
88 ![[Переход в каталог 4ой лабораторной]](image/4.png){#fig:004 width=70%}
89
90 ## Исполним команду make
91 Запустим 'make' чтобы выполнить команды из Makefile
92
93 ![[make]](image/5.png){#fig:005 width=70%}
94
95 ## Проверим новые файлы
96
97 Откроем report.md и report.pdf
98
99 ![[Просмотр сгенерированного PDF файла ]](image/6.png){#fig:006 width=70%}
100 ![[Просмотр сгенерированного PDF файла 2 ]](image/7.png){#fig:007 width=70%}
101
102 ![[Просмотр DOCX файла через libreoffice]](image/8.png){#fig:008 width=70%}
103
104 ## Удалим полученные с использованием Makefile файлы
105
106 Исполним команду 'make clean'
107 ![[Выход команды make]](image/9.png){#fig:009 width=70%}
108
109 Убедимся, что файлы удалены с помощью 'ls'
110 ![[Выход команды ls]](image/10.png){#fig:010 width=70%}
111
112
113 ## Откроем 'report.md' с помощью текстового редактора
114 ![[Просмотр файла report.md]](image/11.png){#fig:011 width=70%}
115
116 Мы видим вспомогательную информацию в начале файла и основную часть документа: заголовки, вставка изображений с подписями и пример таблицы.
117
118
119 ## Заполним отчет 4ой лабораторной
120
121 Сделаем 4ю лабораторную и составим ее отчет в markdown
122
123 ![[Отчет 4ой лабораторной ]](image/12.png){#fig:012 width=70%}
124
125 ## Загрузим файлы на GitHub
126
127 Исполним команды git add, git commit и git push
128 ![[Выгрузка на GitHub]](image/13.png){#fig:013 width=70%}
129
130 # Задания для самостоятельной работы
131
132 ## Сделать отчет по 3й лабораторной работе
133
134 ### Оформим содержимое 3й лабораторной в формате Markdown
135
136
137
138
```

Рис. 3.1: Заполнение отчета для 3 лабораторной

3.2 Скомпилируем отчет по 3 лабораторной

Исполним команду make и проверим, создались ли файлы с помощью ls

```
pandoc "report.md" -F pandoc-crossref --number-sections --citeproc -o "report.docx"
pandoc "report.md" -F pandoc-crossref --pdf-engine=lualatex --pdf-engine-opt=--shell-escape --citeproc --number-sections -o "report.pdf"
[idshevihr@pivo report]$ ls
makefile bib image pandoc report.docx report.md report.pdf
[idshevihr@pivo report]$
```

Рис. 3.2: Выполнение команды make

Просмотрим созданный report.pdf

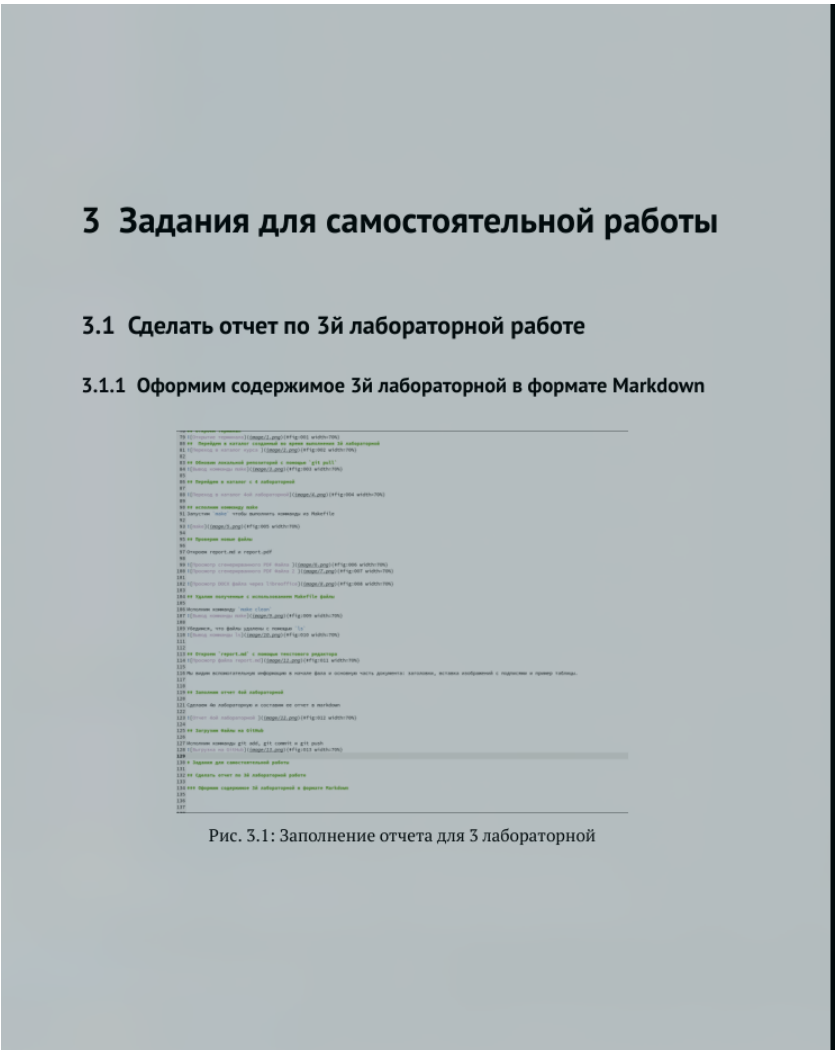


Рис. 3.1: Заполнение отчета для 3 лабораторной

Рис. 3.3: просмотр report.pdf

Сконвертируем в pdf и docx командой make

3.3 Загрузим отчет на GitHub

Исполним комманды `git add`, `git commit`, `git push`

4 Выводы

В ходе данной лабораторной работы, мы освоили процедуры оформления отчетов с помощью

Легковесного языка разметки Markdown, сделали лабораторную работу №4 и отчет к ней в markdown и оформили саму 3ю лабораторную в markdown. А также научились пользоваться командой make и конвертировать отчеты в другие форматы