

# **Лабораторная работа №3**

**Язык разметки Markdown**

Тойчубекова Асель Нурлановна

# Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выводы	15
	Список литературы	16

## Список иллюстраций

3.1	Подключение образа диска . . . . .	8
3.2	Подключение общей папки . . . . .	8
3.3	Копирование скриншотов . . . . .	9
3.4	Запуск терминала и переход в нужный каталог . . . . .	9
3.5	Установка команды gedit . . . . .	9
3.6	Открытие файла для редактирования . . . . .	10
3.7	Редактирование файла . . . . .	10
3.8	Редактирование файла . . . . .	10
3.9	Редактирование файла . . . . .	11
3.10	Редактирование файла . . . . .	11
3.11	Редактирование файла . . . . .	12
3.12	Редактирование файла . . . . .	12
3.13	Компиляция файла . . . . .	12
3.14	Скомпилированные файлы . . . . .	13
3.15	Отправка файлов в репозиторий . . . . .	13

## Список таблиц

# 1 Цель работы

Целью лабораторной работы №3 является научиться оформлять отчеты с помощью языка разметки Markdown

## 2 Задание

- Сделать отчет по предыдущей лабораторной работе в формате Markdown.
- В качестве отчета предоставить отчет в 3 форматах:pdf,docx,rm.

### 3 Теоретическое введение

**Язык разметки Markdown**- это облегченный язык текстовой разметки документов, созданный с целью обозначения формирования в простом тексте, с максимальным сохранением его читаемости человеком, и пригодный для машинного преобразования в языки для продвинутых преобразований(html, rich text и др.) . Он позволяет создавать тексты без использования word и других редакторов. Разметку можно прочитать и воспроизвести в любой системе или браузере. Используя различные знаки мы можем редактировать текст, например:

‘#’-для создания заголовка

’\*\* \*\*’- для создания полужирного начертания текста

’\*\*’-для курсивного начертания курсива

‘>’-блок цитирования

’\*’ ; ‘-’ ; ‘1.’ -для упорядоченного, неупорядоченного, вложенного списка

Для встроенной ссылки действует схема link text или link text. Также существуют определенные правила по оформлению тех или иных формул и изображений в Markdown. В Markdown вставить изображение в документ можно с помощью непосредственного указания адреса изображения. С помощью команды randoc мы можем обработать файлы. | # Выполнение лабораторной работы

Для начала нам, нужно перекинуть все скриншоты, которые нужны при написании отчета на лабораторную работу №2. Для этого я перехожу в настройки устройства и подключаю образ диска дополнений гостевой ОС. (рис. 3.1)

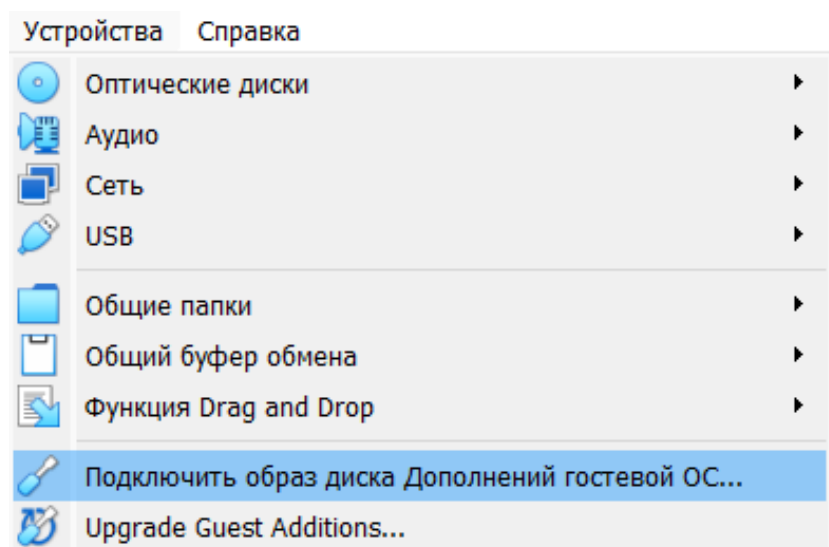


Рис. 3.1: Подключение образа диска

Дальше я перехожу в настройки виртуальной машины и подключаю общую папку, в данном случае эта папка, в которой расположены все скриншоты ко второй лабораторной работе. (рис. 3.2)

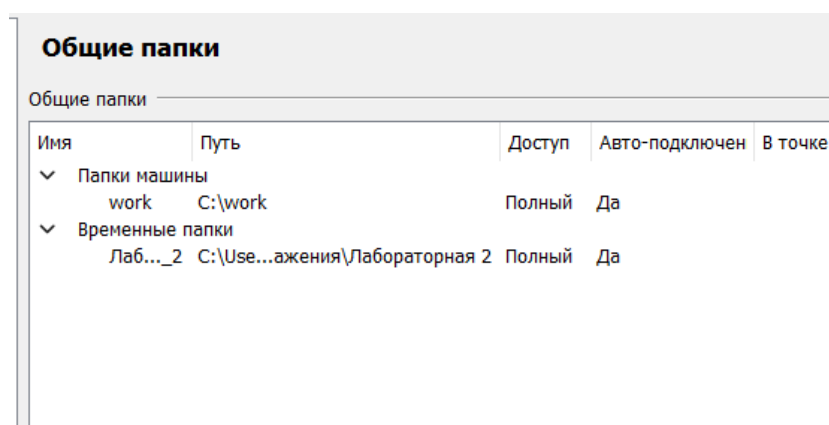


Рис. 3.2: Подключение общей папки

Затем открываю папки в виртуальной машине и перехожу в общую папку, копирую все скриншоты и вставляю их по адресу -> /home/antoychubekova/work/study/2023-2024/Операционные системы/os-intro/labs/lab02/report/image/, там где я буду писать отчет к лабораторной работе №2. (рис. 3.3)





Рис. 3.3: Копирование скриншотов

Открыв терминал перехожу в нужный мне каталог для начала написания отчета к лабораторной работе №2(по адресу /home/antoyjchubekova/work/study/2023-2024/Операционные системы/os-intro/labs/lab02/report) (рис. 3.4)

```
[antoyjchubekova@antoyjchubekova ~]$ cd ~/work/study/2023-2024/"Операционные системы"/os-intro/labs/lab02
[antoyjchubekova@antoyjchubekova lab02]$ ls
presentation report
```

Рис. 3.4: Запуск терминала и переход в нужный каталог

С помощью команды gedit(заранее ее установив) я открываю файл report.md для дальнейшего его редактирования. (рис. 3.5 и рис. 3.6)

```
[antoyjchubekova@antoyjchubekova report]$ sudo dnf install gedit
[sudo] пароль для antoyjchubekova:
Попробуйте ещё раз.
[sudo] пароль для antoyjchubekova:
Fedora 39 - x86_64 - Updates
Fedora 39 - x86_64 - Updates
Последняя проверка окончания срока действия метаданных: 0:00:14 назад, Вт 27 фев 2024 14:43:40.
Зависимости разрешены.
=====
Пакет                                Архитектура
-----
Установка:
  gedit                                x86_64
Установка зависимостей:
  atk                                  x86_64
  libgedit-gtksourceview-299.0.5-1.fc39.x86_64.rpm x86_64
  libpeas1                             x86_64
  libpeas1-gtk                         x86_64
  libpeas1-loader-python3              x86_64
  tcl1                                  x86_64
=====
Результат транзакции
=====
Установка 7 Пакетов

Объем загрузки: 3.3 М
Объем изменений: 16 М
Продолжить? [д/н]: y
Загрузка пакетов:
(1/7): atk-5.6.1-3.fc39.x86_64.rpm
(2/7): tcl1-6.6.0-1.fc39.x86_64.rpm
(3/7): libgedit-gtksourceview-299.0.5-1.fc39.x86_64.rpm
(4/7): libpeas1-gtk-1.36.0-2.fc39.x86_64.rpm
(5/7): libpeas1-loader-python3-1.36.0-2.fc39.x86_64.rpm
(6/7): libpeas1-1.36.0-2.fc39.x86_64.rpm
(7/7): gedit-45.0-1.fc39.x86_64.rpm
=====
Общий размер
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверка транзакции
Тест транзакции проведен успешно.
Выполнение транзакции
Подготовка      :
Установка       : libpeas1-1.36.0-2.fc39.x86_64
Установка       : libgedit-gtksourceview-299.0.5-1.fc39.x86_64
```

Рис. 3.5: Установка команды gedit

```
[antoyjchubekova@antoyjchubekova report]$ gedit report.md
[antoyjchubekova@antoyjchubekova report]$ ls
bib image Makefile pandoc report.md
```

Рис. 3.6: Открытие файла для редактирования

Далее я приступаю к самому написанию отчета. Заполняю титульный лист лабораторной работы. (рис. 3.7)

```
1 ---
2 ## Front matter
3 title: "Лабораторная работа №2"
4 subtitle: "Первоначальная настройка git"
5 author: "Тойчубекова Асель Нурлановна"
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 linestretch: 1.5
22 papersize: a4
23 documentclass: scrreprt
24 ## I18n polyglossia
25 polyglossia-lang:
26   name: russian
27   options:
```

Рис. 3.7: Редактирование файла

Описываю цель работы и задание, которое я должна выполнить. (рис. 3.8)

```
# Цель работы
Целью лабораторной работы №2 является изучение идеологии и применение средств контроля версий, а также освоение умений по работе с git.

# 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. 3.8: Редактирование файла

Расписываю теоретическое введение. (рис. 3.9)

```

1  # Теоретическое введение
2
3  Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта
4  участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещая изменения, произведённые
5  участниками проекта.
6
7  В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выпo
8  сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную ему версию файлов. После внесения изме
9  не удаляются из центрального хранилища и к ним можно вернуться в любой момент. Сервер может сохранять не полную версию изменённых файлов, а производит
10 последовательными версиями, что позволяет уменьшить объём хранимых данных. Системы контроля версий (Version Control System, VCS) применяются при работе
11 локальным или удалённым репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля
12 участниками проекта, производить откат к любой более ранней версии проекта, если это требуется.
13
14 В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выпo
15 сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную ему версию файлов. После внесения изме
16 не удаляются из центрального хранилища и к ним можно вернуться в любой момент. Сервер может сохранять не полную версию изменённых файлов, а производит
17 последовательными версиями, что позволяет уменьшить объём хранимых данных.
18
19 Система контроля версий Git представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды
20 контроля версий, резервную копию локального хранилища можно сделать простым копированием или архивацией.
21

```

Рис. 3.9: Редактирование файла

## Расписываю ход выполнения лабораторной работы. (рис. 3.10)

```

1  # Выполнение лабораторной работы
2
3  ## Установка программного обеспечения ##
4
5  Сперва захожу в терминал и устанавливаю git, используя команду 'dnf install git'. (рис. [-@fig:002])
6
7  ![Установка git ](image/2.1.png){#fig:002 width=70%}
8
9  Устанавливаю gh, используя команду 'dnf install gh'. (рис. [-@fig:003])
10
11 ![Установка gh ](image/2.2.png){#fig:003 width=70%}
12
13 ## Базовая настройка git. ##
14
15 Пользуюсь командой 'git config --global user.name "Asel Toychubekova"' задаю свое имя, а командой 'git config --g
16 сообщений git командой 'git config --global core.quotepath false'. (рис. [-@fig:004])
17
18 ![Установка имени, email пользователя и utf-8 ](image/2.3.png){#fig:004 width=70%}
19
20 Задаю имя начальной ветки, название-master, командой 'git config --global init.defaultBranch master', затем настр
21 (рис. [-@fig:005])
22
23 ![Задание начальной ветки и настройка параметров autocrlf, safecrlf](image/2.4.png){#fig:005 width=70%}
24
25 ## Создание ключей ssh ##
26
27 Создаю ключ по алгоритму rsa с ключём размером 4096 бит, после чего создаю ключ по алгоритму ed25519, используя к
28
29 ![Создание ssh ключей.](image/2.5.png){#fig:006 width=70%}
30
31 ## Создание ключа pgp ##
32
33 Генерирую ключ, командой 'gpg --full-generate-key', выбирая подходящие из предложенных опций. (рис. [-@fig:007] и
34
35 ![Создание pgp ключа](image/2.6.png){#fig:007 width=70%}

```

Рис. 3.10: Редактирование файла

## Отвечаю на контрольные вопросы. (рис. 3.11)

```

}
}
} ## Ответы на контрольные вопросы ##
}
}
} контрольные вопросы
}
}
} 1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются?
}
} Системы контроля (VCS) – программное обеспечение для облегчения работы с изменяющейся информацией,
} ранние версии этой информации – проекта. Это программное обеспечение позволяет просматривать кем и ко
} удобства работы над проектом нескольких людей.\
}
}
} 2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.
}
} Хранилище – репозиторий, где сохраняются все версии проекта, в нем хранятся все документы, история их
} Commit – Отлаживает все изменения, сохраняет разницу версий, их изменения.\
} История – хранит все изменения в проекте и позволяет при необходимости использовать нужные данные из
} Рабочая копия – копия проекта, основанная на версии из хранилища, последней версии.\
}
}
} 3. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите при
}
} Централизованные VCS – одно основное хранилище всего проекта. Каждый пользователь копирует себе не
} Децентрализованные VCS – в этих VCS у каждого пользователя свой вариант репозитория, есть возможность
} В отличие от классических в децентрализованных системах контроля версий центральный репозиторий
} Централизованные VCS – CVS, TFS, AccuRev.\
} Децентрализованные VCS – Git, Bazaar.\
}
}
} 4. Опишите действия с VCS при единоличной работе с хранилищем.
}
} Для начала для всех создается и подключается центральный репозиторий, затем по мере изменений про
}
} 5. Опишите порядок работы с общим хранилищем VCS.
}
} После начала работы все участники проекта получают версию проекта в хранилище, после изменений поль

```

Рис. 3.11: Редактирование файла

Делаю вывод и указываю литературные источники. (рис. 3.12)

```

# Выводы
}
}
} В ходе выполнения лабораторной работы №2 я изучила идеологию и примечание средств контроля ве
}
}
} # Список литературы{.unnumbered}
} - https://esystem.rudn.ru/mod/page/view.php?id=1098933#org2151722.
}
}
}

```

Рис. 3.12: Редактирование файла

После изменения файла, используя команду make, я выполняю его компиляцию из формата md в docx, pdf. (рис. 3.13). Перейдя в папку лабораторной работы мы видим, что файлы скомпилировались. (рис. 3.14)

```

[antoyjchubekova@antoyjchubekova report]$ make
pandoc "report.md" --filter pandoc-crossref --number-sections --citeproc -o "report.docx"
pandoc "report.md" --filter pandoc-crossref --pdf-engine=lualatex --pdf-engine-opt=--shell-escape --citeproc --number-sections -o "report.pdf"
[antoyjchubekova@antoyjchubekova report]$

```

Рис. 3.13: Компиляция файла




Рис. 3.14: Скомпилированные файлы









Затем отправляю измененный и скомпилированные файлы на глобальный репозиторий, используя команды `git add . -> git commit -> git push`. (рис. 3.15 и рис.??). Заходим в `github` и видим, что файлы были успешно доставлены и сохранены.

```
[antoychubekova@antoychubekova report]$ git add .
[antoychubekova@antoychubekova report]$ git commit -am 'feat(main): add files lab02'
[master c42b012] feat(main): add files lab02
28 files changed, 191 insertions(+), 35 deletions(-)
create mode 100755 labs/lab02/report/image/1.png
create mode 100755 labs/lab02/report/image/10.png
create mode 100755 labs/lab02/report/image/11.png
create mode 100755 labs/lab02/report/image/12.png
create mode 100755 labs/lab02/report/image/13.png
create mode 100755 labs/lab02/report/image/14.png
create mode 100755 labs/lab02/report/image/15.png
create mode 100755 labs/lab02/report/image/16.png
create mode 100755 labs/lab02/report/image/17.png
create mode 100755 labs/lab02/report/image/18.png
create mode 100755 labs/lab02/report/image/19.png
create mode 100755 labs/lab02/report/image/2.png
create mode 100755 labs/lab02/report/image/20.png
create mode 100755 labs/lab02/report/image/21.png
create mode 100755 labs/lab02/report/image/22.png
create mode 100755 labs/lab02/report/image/23.png
create mode 100755 labs/lab02/report/image/3.png
create mode 100755 labs/lab02/report/image/4.png
create mode 100755 labs/lab02/report/image/5.png
create mode 100755 labs/lab02/report/image/6.png
create mode 100755 labs/lab02/report/image/7.png
create mode 100755 labs/lab02/report/image/8.png
create mode 100755 labs/lab02/report/image/9.png
create mode 100644 labs/lab02/report/report.docx
create mode 100644 labs/lab02/report/report.pdf
[antoychubekova@antoychubekova report]$ git push
Перечисление объектов: 44, готово.
Подсчет объектов: 100% (42/42), готово.
При сжатии изменений используется до 8 потоков
Сжатие объектов: 100% (35/35), готово.
Запись объектов: 100% (35/35), 8.54 МБ | 550.00 КиБ/с, готово.
Всего 35 (изменений 3), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:aseltoichubekova/study_2023-2024_os-intro.git
 167046b..c42b012 master -> master
[antoychubekova@antoychubekova report]$
```

Рис. 3.15: Отправка файлов в репозиторий

[study\\_2023-2024\\_os-intro](#) / [labs](#) / [lab02](#) / [report](#) / 

 **aseltoichubekova** feat(main): add files lab02

Name	La
 ..	
 bib	fe
 image	fe
 pandoc	fe
 Makefile	fe
 report.docx	fe
 report.md	fe
 report.pdf	fe

## 4 Выводы

В ходе выполнения лабораторной работы №3 я научилась оформлять отчеты с помощью языка разметки Markdown

## Список литературы

- <https://esystem.rudn.ru/course/view.php?id=5790>.
- <https://help.vivaldi.com/ru/services-ru/forum-ru/markdown-formatting/>
- <https://ru.wikipedia.org/wiki/Markdown>.