

Лабораторная работа номер 3

Markdown

Титков Ярослав Максимович

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	Выводы	12

Список иллюстраций

4.1	Заполнение титульного листа	9
4.2	Общая структура, цели, теор.часть, прикрепление выполненных заданий	9
4.3	Вывод и ответ на контрольные вопросы	10
4.4	Выполнив команду make мы создали dosx и пдф файл, проверив все ли правильно работает	10
4.5	Проверка	11

Список таблиц

1 Цель работы

Научиться оформлять отчёты с помощью легковесного языка разметки Markdown

2 Задание

Сделайте отчёт по предыдущей лабораторной работе в формате Markdown. 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

Титульный лист. Первый лист работы оформляется строго по образцу, который обычно приводится в методических пособиях по вашему предмету. В нем не просто требуется указать такие элементы, как название образовательного учреждения, вид работы и сведения об исполнителе, но и расположить их в строгом соответствии со стандартами. Реферат. Реферат фактически является кратким представлением всего вашего отчета и содержит ряд статистических сведений. В нем нужно указать количество частей, страниц работы, иллюстраций, приложений, таблиц, использованных литературных источников и приложений. Здесь же приводится перечень ключевых слов работы и собственно текст реферата. Последний подразумевает основные элементы работы от поставленных целей до результатов и рекомендаций по их внедрению. В практике вузов в отчеты по лабораторным работам реферат обычно не включают. Введение. Во введении типовой лабораторной работы обычно прописывают цели проводимого исследования и задачи, выполнение которых поможет достичь поставленных целей. В то же время существуют работы, в которых студенты становятся настоящими первооткрывателями. Приходилось ли вам хотя бы однажды испытывать чувство крайнего любопытства и нетерпения при проведении лабораторной работы? Ощущать, что буквально через пару минут вы найдете ответ на вопрос, на который еще никто и никогда не находил ответа? Именно для таких исследований пишется раз-вернутое введение с доказательством актуальности и новизны изучаемой темы. Чтобы действительно провести исследование в той области, в которой, как говорится, еще не ступала нога человека, во введении вам понадо-

бится привести оценку современного состояния рассматриваемой проблемы и обосновать необходимость ее решения Основная часть. Так как в разных вузах и в разных дисциплинах существуют свои тонкости проведения лабораторных работ, содержание основной части подробно описывают в соответствующих методичках. Важно, чтобы в этом разделе работы была отражена ее суть, описана методика и результаты проделанной работы. В основной части прописывают следующие элементы: – цели проводимого исследования; – задачи, выполнение которых поможет достичь поставленных целей; – ход работы, в котором описываются выполненные действия; – прочие разделы, предусмотренные методическими материалами по изучаемой дисциплине. Заключение. В этой части работы вам потребуется сделать выводы по полученным в ходе лабораторной работы результатам. Для этого оцените, насколько полно выполнены поставленные задачи. В сложных работах могут присутствовать и другие элементы, например, рекомендации для дальнейшего применения результатов проведённой работы.

4 Выполнение лабораторной работы

```
---
## Front matter
title: "Лабораторная работа номер 2"
subtitle: "Выполнение лабораторной работы"
author: "Титков Ярослав Максимович"

## Generic otions
lang: ru-RU
toc-title: "Содержание"

## Bibliography
bibliography: bib/cite.bib
csl: pandoc/csl/gost-r-7-0-5-2008-numeric.csl

## Pdf output format
toc: true # Table of contents
toc-depth: 2
lof: true # List of figures
lot: true # List of tables
fontsize: 12pt
linestretch: 1.5
papersize: a4
documentclass: scrreprt
## I18n polyglossia
polyglossia-lang:
  name: russian
  options:
    - spelling=modern
    - babelshorthands=true
polyglossia-otherlangs:
  name: english
## I18n babel
babel-lang: russian
babel-otherlangs: english
## Fonts
mainfont: IBM Plex Serif
romanfont: IBM Plex Serif
sansfont: IBM Plex Sans
monofont: IBM Plex Mono
mathfont: STIX Two Math
mainfontoptions: Ligatures=Common,Ligatures=TeX,Scale=0.94
romanfontoptions: Ligatures=Common,Ligatures=TeX,Scale=0.94
sansfontoptions: Ligatures=Common,Ligatures=TeX,Scale=MatchLowercase,Scale=0.94
monofontoptions: Scale=MatchLowercase,Scale=0.94,FakeStretch=0.9
mathfontoptions:
## Biblatex
biblatex: true
biblio-style: "gost-numeric"
biblatexoptions:
  - parenttracker=true
```

Рис. 4.1: Заполнение титульного листа

```
! цель работы
! Вучать идеологию и применение средств контроля версий.
! Освоить умения по работе с GIT.

! Задание
1. Создание Git-клиней
2. Подключение репозитория
3. Работа с Github

! Теоретическое введение
Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен доступ для участников.
В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальными сервером.
Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Можно объединять (слить) изменения, сделанные разными участниками (автоматически).
Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохранение общей истории изменений до точки ветвления.
В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным.
Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых – Git, Bazaar, Mercurial. Принципы их работы схожи, отличается они в основном синтаксисом используемых в работе команд.

! Выполнение лабораторной работы
! [Установка Git](img/1.jpg){fig:001 width=700}
! [Создание клена](img/2.jpg){fig:002 width=700}

! Выводы
Здесь кратко описываются итоги проделанной работы.

! Список литературы(,optional!)
```

Рис. 4.2: Общая структура, цели, теор.часть, прикрепление выполненных заданий

```

![[Получение индивидуального кода](image/4.png){#fig:004 width=70%}]
![[Клонирование репозитория](image/5.png){#fig:005 width=70%}]
![[Создание ветки](image/6.png){#fig:006 width=70%}]

# Выводы
В ходе работы я освоил и научился пользоваться Git

# Контрольные вопросы:

Системы контроля версий (VCS)

Инструменты для управления изменениями в файлах. Используется для истории изменений, совместной работы и отката.
Основные понятия VCS:

Хранилище (репозиторий) – база данных с версиями файлов.
Commit – фиксация изменений с комментарием.
История – последовательность коммитов.
Рабочая копия – локальные файлы для работы.

Централизованные и децентрализованные VCS

Централизованные (CVCS): Одно главное хранилище. Пример: SVN.
Децентрализованные (DVCS): У каждого своя копия репозитория. Пример: Git, Mercurial.

Действия с VCS при одиночной работе:

Создать репозиторий (git init).
Добавить файлы (git add).
Фиксировать изменения (git commit).
Просматривать историю (git log).

Работа с общим хранилищем:

Клонировать репозиторий (git clone).
Получать изменения (git pull).
Отправлять изменения (git push).
Решать конфликты при необходимости.

```

Рис. 4.3: Вывод и ответ на контрольные вопросы

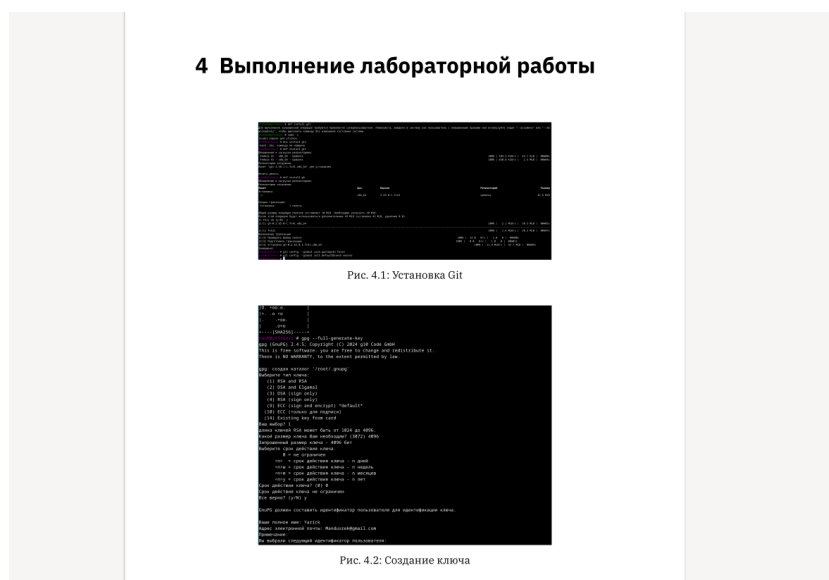


Рис. 4.4: Выполнив команду make мы создали docx и pdf файл, проверив все ли правильно работает

6 Контрольные вопросы:

Системы контроля версий (VCS)

Инструменты для управления изменениями в файлах. Используются для истории изменений, совместной работы и отката. Основные понятия VCS:

Хранилище (репозиторий) – база данных с версиями файлов.

Commit – фиксация изменений с комментарием.

История – последовательность коммитов.

Рабочая копия – локальные файлы для работы.

Централизованные и децентрализованные VCS

Централизованные (CVCS): Одно главное хранилище. Пример: SVN.

Децентрализованные (DVCS): У каждого своя копия репозитория. Пример: Git, Mercurial.

Действия с VCS при единоличной работе:

Создать репозиторий (`git init`).

Добавить файлы (`git add`).

Фиксировать изменения (`git commit`).

Просматривать историю (`git log`).

Работа с общим хранилищем:

Клонировать репозиторий (`git clone`).

Получать изменения (`git pull`).

Отправлять изменения (`git push`).

Решать конфликты при необходимости.

Рис. 4.5: Проверка

5 Выводы

В ходе выполнения работы мы научились составлять отчёты с помощью языка Markdown