Лабораторная работа №3

Markdown

Титков Ярослав Максимович

Российский университет дружбы народов, Москва, Россия

Цель работы

Научиться оформлять отчёты с помощью легковесного языка разметки Markdown

Задание

– Сделать 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.)

Мы открыли файл Markdown, чтобы начать его заполнять,офрмили титульный лист

```
---
## 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:
  - parentracker=true
```

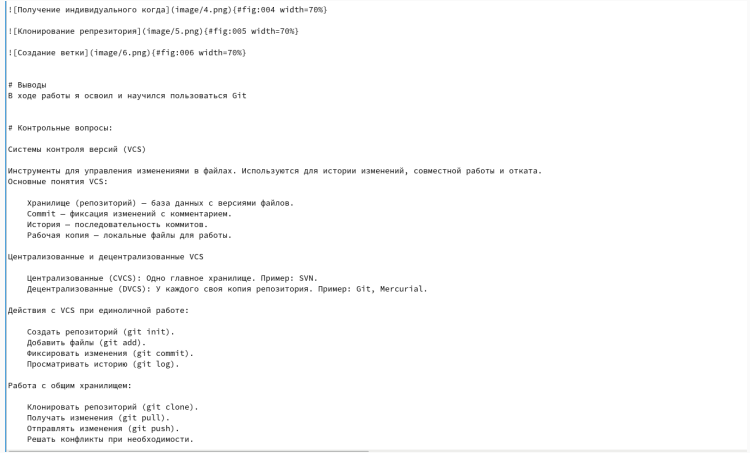

Рис. 1: Открытие файла Markdown

Затем мы начали прикреплять скрины, отвечающие за выполнение нашей работы

```
# Цель работы

    Изучить идеологию и применение средств контроля версий.
    Освоить умения по работе с git.


# Задание
1.Создание Гит ключей
2.Подключение репозитория
3. Работа с GitHub

# Теоретическое введение
Системы контроля версий. Общие понятия

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 участни

В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. 

Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Можно объединить (слить) изменения, сделанные разными участниками (автомат

Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки ветвлени

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, отличаются они в основном синтаксисом используемых в работе команд.


# Выполнение лабораторной работы

|[Установка Git](image/1.jpg){#fig:001 width=70%}

|[Создание ключа](image/2.jpg){#fig:001 width=70%}

|

# Выводы

Здесь кратко описывается итоги проделанной работы.

# Список литературы{.unnumbered}
```

Рис. 2: Прикрепление скринов

После этого оформили вывод и ответили на контрольные вопросы по прошлой лабораторной работе



```
![Получение индивидуального когда](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. 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. 3: Оформления вывода и ответа на контрольные вопросы

Скоомпелировав файлы мы открыли наш pdf файл и проверили как все работает

4 Выполнение лабораторной работы

Рис. 4.1: Установка Git

Рис. 4.2: Создание ключа

Рис. 4: Проверка

В ходе работы я убедился, что все скрины стоят правильно и все ответы присутствуют

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.

Вывод

Я научился оформлять отчёты с помощью языка Markdown