

Лабораторная работа № 3

Бондарь Т. В.

Российский университет дружбы народов, Москва, Россия

Информация

- Бондарь Татьяна Владимировна
- НКАбд-01-24, студ. билет №1132246711
- Российский университет дружбы народов
- https://github.com/tvbondar/study_2024-2025_os-intro

Научиться оформлять отчёты с помощью легковесного языка разметки Markdown.

1. Сделайте отчёт по предыдущей лабораторной работе в формате Markdown.
2. В качестве отчёта просьба предоставить отчёты в 3 форматах: pdf, docx и md (в архиве, поскольку он должен содержать скриншоты, Makefile и т.д.)

Чтобы создать заголовок, используйте знак (#). Чтобы задать для текста полужирное начертание, заключите его в двойные звездочки. Чтобы задать для текста курсивное начертание, заключите его в одинарные звездочки. Чтобы задать для текста полужирное и курсивное начертание, заключите его в тройные звездочки. Блоки цитирования создаются с помощью символа >. Неупорядоченный (маркированный) список можно отформатировать с помощью звездочек или тире. Чтобы вложить один список в другой, добавьте отступ для элементов дочернего списка. Упорядоченный список можно отформатировать с помощью соответствующих цифр. Чтобы вложить один список в другой, добавьте отступ для элементов дочернего списка. 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. Внутритекстовые формулы делаются аналогично формулам LaTeX. Для обработки файлов в формате Markdown будем использовать Pandoc. Конкретно, нам понадобится программа `pandoc`, `pandoc-citeproc` <https://github.com/jgm/pandoc/releases>, `pandoc-crossref` <https://github.com/lierdakil/pandoc-crossref/releases>. Преобразовать файл README.md можно следующим образом: `1 pandoc README.md -o README.pdf` или так `1 pandoc README.md -o README.docx` Можно использовать следующий Makefile `1 FILES = $(patsubst %.md, %.docx, $(wildcard *.md)) 2 FILES += $(patsubst %.md, %.pdf, $(wildcard *.md))`

1. Открываем подготовленный файл .md, меняем в нем имя, фамилию автора, изменяем название, добавляем данные автора.

```
---  
## Front matter  
title: "Отчет по лабораторной работе №2"  
author: "Бридарь Татьяна Владимировна"  
  
## Generic options  
lang: ru-RU  
toc-title: "Содержание"  
  
## Bibliography  
bibliography: bib/cite.bib  
csl: pandoc/csl/gost-r-7-0-5-2008-numeric.csl  
---
```

Рис. 1: Начало заполнения отчета

2. Записывем цель работы, задание, теоретическое введение.

Цель работы

Изучить назначение и основные свойства версионных систем, научиться работать со версией в GIT.

Задание

1. Изучить основные терминологические понятия в GIT.

2. Загрузить репозиторий на GitHub.

3. Создать ветку test.

4. Создать файл test.py.

5. Настроить локальный репозиторий.

6. Создать локальный репозиторий без удаленного репозитория в локальной среде.

Теоретическое введение

Система версионного контроля (Version Control System, VCS) предназначена для работы с исходными текстами программного обеспечения. Она позволяет хранить историю изменений кода, создавать ветки для разработки, делать коммиты и т.д. В зависимости от того, как используется система, она может быть централизованной или распределенной. В централизованной системе есть один сервер, на котором хранятся все данные. В распределенной системе каждый участник имеет свою копию репозитория.

Система версионного контроля позволяет хранить историю изменений кода, создавать ветки для разработки, делать коммиты и т.д. В зависимости от того, как используется система, она может быть централизованной или распределенной. В централизованной системе есть один сервер, на котором хранятся все данные. В распределенной системе каждый участник имеет свою копию репозитория.

Система версионного контроля позволяет хранить историю изменений кода, создавать ветки для разработки, делать коммиты и т.д. В зависимости от того, как используется система, она может быть централизованной или распределенной. В централизованной системе есть один сервер, на котором хранятся все данные. В распределенной системе каждый участник имеет свою копию репозитория.

Рис. 2: Цель, задание, теоретическое введение

3. Начинаем заполнять основную часть работы: прописываем шаги выполнения лабораторной работы, добавляем иллюстрации и подписи к ним.

```
# Выполнение лабораторной работы

## Установка программного обеспечения

1. Установка git, перейдя на сайт скачиваем установщик. (рис. @fig:001).

![[Установка git]](image/1.png){#fig:001 width=70%}

2. Установка svn (рис. @fig:002).

![[Установка svn]](image/2.png){#fig:002 width=70%}

## Настройка git.

3. Задаем имя и email владельца репозитория. Настроили svn-8 и выведем список файлов. Задаем имя начальной ветки, параметрам autoCRLF, autoCRLF. (рис. @fig:003).
```

Рис. 3: Заполнение части “Выполнение лабораторной работы”

4. Прописываем вывод, отвечаем на контрольные вопросы к лабораторной работе.

Выводы

В ходе данной лабораторной работы мы рассмотрели возможности и возможности системы контроля версий, используя знания по работе с Git.

Контрольные вопросы

1. Что такое система контроля версий (СКВ) и для чего она предназначена?

- Система контроля версий - программное обеспечение для отслеживания изменений в компьютерных файлах и управления версиями. Система позволяет отслеживать изменения кода, сохранять версии кода и легко их восстанавливать. Она предоставляет инструменты для работы с ветками, слияния, реверса, тегов и других функций. Это и есть СКВ. Для нее есть несколько вариантов, в том числе Git. Систему контроля версий (Version Control System, VCS) представляют так:

- Система может отслеживать изменения
- Может работать с кодом или с файлами на диске
- Позволяет работать с изменениями на протяжении времени в процессе
- Сохраняет историю изменений и позволяет вернуться к предыдущим версиям
- Возможность работать с кодом и файлами в режиме реального времени

2. Какие существуют типы СКВ и их особенности? Какие есть, например, Git, Mercurial, Subversion?

- Системный тип СКВ - это тип СКВ, который не требует установки на компьютер и работает на сервере. Он используется для хранения и управления версиями кода.

- Commit - это команда, которая используется для сохранения изменений.

- Проверка кода - это команда, которая используется для проверки кода на соответствие требованиям. Она используется для проверки кода на соответствие требованиям.

- Система контроля версий может использоваться для отслеживания изменений в файлах и папках. Она используется для отслеживания изменений в файлах и папках.

3. Что представляет собой и для чего предназначена система контроля версий? Какие есть, например, Git, Mercurial, Subversion?

- Система контроля версий (СКВ) - это программное обеспечение, которое позволяет отслеживать изменения в файлах и папках. Она используется для отслеживания изменений в файлах и папках.

- Это программное обеспечение, которое позволяет отслеживать изменения в файлах и папках.

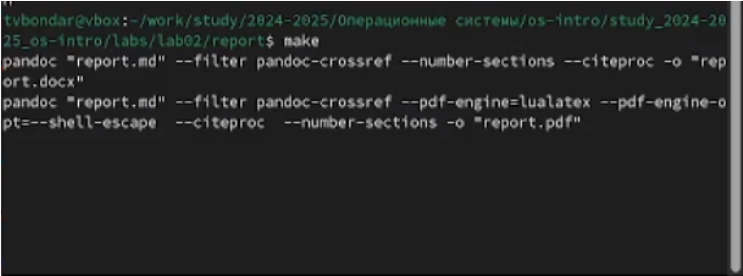
- Можно использовать систему контроля версий для отслеживания изменений в файлах и папках. Она используется для отслеживания изменений в файлах и папках.

- У каждого пользователя есть свой репозиторий (хранилище) для хранения изменений. Он используется для хранения изменений.

- Предоставляет возможность работать с кодом и файлами в режиме реального времени. В режиме реального времени можно использовать систему контроля версий. Система контроля версий позволяет отслеживать изменения в файлах и папках. Она используется для отслеживания изменений в файлах и папках.

Рис. 4: Заполнение части “Выводы” и “Контрольные вопросы”

5. С помощью команды make создаем отчет в форматах .docx и .pdf.



```
tvbondar@vbox:~/work/study/2024-2025/Операционные системы/os-intro/study_2024-2025_os-intro/labs/lab02/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"
```

Рис. 5: Команда make

Мы научились оформлять отчёты с помощью легковесного языка разметки Markdown.

Руководство по оформлению Markdown файлов. [Электронный ресурс]. GitHub Gist URL:
<https://gist.github.com/Jekins/2bf2d0638163f1294637>