

Лабораторная работа №3

Операционные системы

Панина Ж. В.

04 марта 2025

Российский университет дружбы народов, Москва, Россия

Информация

- Панина Жанна Валерьевна
- НКАбд-02-24, студ. билет № 1132246710
- студент направления “Компьютерные и информационные науки”
- Российский университет дружбы народов
- 1132246710@pfur.ru
- https://github.com/zvpanina/study_2024-2025_os-intro

Вводная часть

В современных IT-сферах ведение технической документации является важной частью работы разработчиков, системных администраторов и исследователей. Markdown — это удобный и широко используемый легковесный язык разметки, который позволяет легко создавать структурированные и читаемые документы, включая отчёты, README-файлы и технические статьи. Навыки работы с Markdown актуальны для написания документации в репозиториях, ведения вики-страниц и публикации контента в блогах или системах управления проектами.

Объект исследования:

Легковесный язык разметки Markdown и его применение для оформления технической документации.

Предмет исследования:

Методы разметки и форматирования текста в Markdown, вставка изображений и ссылок, а также оформление отчётов в соответствии с требованиями.

Цель работы - научиться оформлять отчёты с помощью легковесного языка разметки Markdown.

Задачи:

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

Материалы:

- Операционная система (Fedora, установленная в VirtualBox)
- Редактор с поддержкой Markdown
- Фотографии и изображения для иллюстрации отчёта
- Хостинг изображений или локальные файлы для вставки ссылок

Методы:

- Изучение базового синтаксиса Markdown
- Практическое оформление отчёта с использованием заголовков, списков, таблиц и выделения текста
- Вставка изображений и ссылок на них
- Проверка итогового оформления отчёта в разных Markdown-редакторах или репозиториях
- Сохранение и публикация отчёта в GitHub

Теоретическое введение

Чтобы создать заголовок, используется знак (#). Чтобы задать для текста полужирное начертание, нужно заключить его в двойные звездочки. Чтобы задать для текста курсивное начертание, нужно заключить его в одинарные звездочки. Чтобы задать для текста полужирное и курсивное начертание, заключаем 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, нужно добавить 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))`

Выполнение лабораторной работы

Создание отчёта по предыдущей лабораторной работе

1. Открываю шаблон отчёта формата .md, меняю в нём ФИО автора, изменяю название, добавляю свои данные.

```
---  
  
## Front matter  
title: "Лабораторная работа №2"  
subtitle: "Дисциплина: Операционные системы"  
author: "Панина Жанна Валерьевна"  
  
## 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: Первый этап заполнения отчёта

2. Пишу цель работы, задание и теоретическое введение.

Цель работы

Цель работы – изучить идеологию и применения средств контроля версий, освоить умения по работе с `git`.

Задание

1. Создать базовую конфигурацию для работы с `git`.
2. Зарегистрироваться на `github`.
3. Создать ключ `SSH`.
4. Создать ключ `PGP`.
5. Настроить подписи `git`.
6. Создать локальный каталог для выполнения заданий по предмету.

Теоретическое введение

Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное де удалённом `репозитории`, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля совмещать изменения, произведённые разными участниками проекта, производить откат к любой более ранней версии проекта, если это требует версий используется централизованная модель, предполагающая наличие единого `репозитория` для хранения файлов. Выполнение большинства фу осуществляется специальным сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нуж

Рис. 2: Цель работы, задание и теоретическое введение

3. Заполняю основную часть работы: прописываю шаги выполнения работы, добавляю ссылки на иллюстрации и подписи к ним.

```
# Выполнение лабораторной работы

## Создание ключа SSH

Поскольку я уже была зарегистрирована на GitHub в первом семестре, начинаю работу с создания ключа SSH (рис. [-@fig:001]).

!{Создание ключа SSH} (image/1.png) {#fig:001 width=70%}

## Создание ключа GPG

1. Создаю ключ GPG, указав необходимые параметры (рис. [-@fig:002]).

!{Создание ключа GPG} (image/2.png) {#fig:002 width=70%}

2. Вывожу список ключей и копирую отпечаток приватного ключа. Копирую сгенерированный ключ GPG (рис. [-@fig:003]).

!{Ключ GPG} (image/3.png) {#fig:003 width=70%}
```

Рис. 3: Основная часть: выполнение лабораторной работы

4. Отвечаю на контрольные вопросы к лабораторной работе.

Ответы на контрольные вопросы

1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются?

- Система контроля версий – программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями позволяет у того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое.

(Version Control System, VCS) применяются для:

- Хранение полной истории изменений
- причин всех производимых изменений
- Откат изменений, если что-то пошло не так
- Поиск причины и ответственного за появления ошибок в программе
- Совместная работа группы над одним проектом
- Возможность изменять код, не мешая работе других пользователей

2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия

- **Репозиторий** – хранилище версий – в нем хранятся все документы вместе с историей их изменения и другой служебной информацией.

- **Commit** – отслеживание изменений

- Рабочая копия – копия проекта, связанная с **репозиторием** (текущее состояние файлов проекта, основанное на версии из хранилища (обычно на локальном компьютере))

- История хранит все изменения в проекте и позволяет при необходимости обратиться к нужным данным.

Рис. 4: Заключительная часть: ответы на контрольные вопросы

5. Прописываю вывод о работе.

git commit -m "

Выводы

Во время выполнения работы я изучила идеологию и применения средств контроля версий, освоила умения по работе с `git`.

Рис. 5: Вывод

С помощью команды make создаю отчёт в форматах .docx и .pdf.

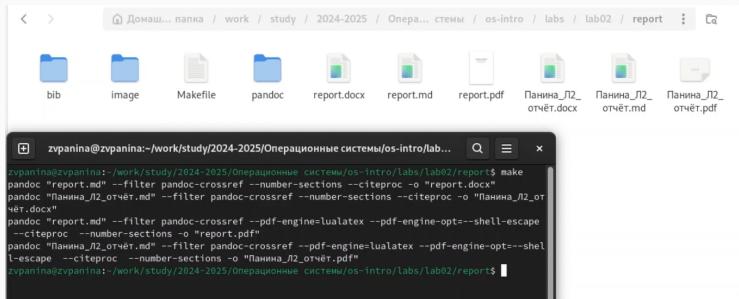


Рис. 6: Команда make

Результаты

В ходе выполнения лабораторной работы я научилась оформлять отчёты с помощью легковесного языка разметки Markdown.

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