

# Лабораторная работа №3

Язык разметки Markdown

---

Рогожина Н.А.

25 февраля 2023

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Рогожина Надежда Александровна
- Студентка 1го курса, НКАбд-02-22
- Компьютерные и информационные науки
- Российский университет дружбы народов
- Github

## Вводная часть

---

- Умение работать с файлами формата *Markdown* может существенно сократить время, потраченное на подготовку отчётов и презентаций по лабораторным работам, снимая вопросы технической части и оставляя на нас только наполнение самих отчётов и презентаций.

- Легковесный язык разметки Markdown

- Научиться оформлять отчёты с помощью легковесного языка разметки Markdown

- Процессор **pandoc** для входного формата Markdown
- Результирующие форматы
  - pdf
  - html
- Автоматизация процесса создания: **Makefile**



## Содержание лабораторной работы

---

Первым шагом сделаем титульный лист нашего отчёта:

```
Sta ---  
Hor ## Front matter  
Vid title: "Отчёт по лабораторной работе №2"  
Док subtitle: "Операционные системы"  
author: "Рогожина Надежда Александровна, НКАбд-02-22"  
## Generic otions
```

Рис. 1: Титульный лист

Напишем цель и задачи лабораторной работы:

```
30 # Цель работы
у: - Изучить идеологию и применение средств контроля версий.
а - Освоить умения по работе с git.
# Задание
```

Рис. 2: Цель работы

## # Задание

Вид 1. Установка программного обеспечения

Доп - Установка git

Заг - Установка gh

Изс 2. Базовая настройка git

Му: 3. Создайте ключи ssh

Wa 4. Создайте ключи gpg

5. Настройка github

6. Добавление GPG ключа в GitHub

7. Настройка автоматических подписей коммитов git

8. Настройка gh

9. Шаблон для рабочего пространства

- Создание репозитория курса на основе шаблона

- Настройка каталога курса

Оформляем кратко теоретические аспекты нашей лабораторной работы:

```
До # Теоретическое введение
Заг ## Примеры использования git:
Изс
Му: - Система контроля версий Git представляет собой набор программ командной строки.
Wa - Доступ к ним можно получить из терминала посредством ввода команды git с различными
опциями.
- Благодаря тому, что Git является распределённой системой контроля версий, резервную
копию локального хранилища можно сделать простым копированием или архивацией.
```

Рис. 4: Теоретическое введение

```
копию локального хранилища можно сделать простым копированием или архивацией.
```

**## Основные команды git:**

Например, в табл. @tbl:std-dir приведено краткое описание основных команд Git.

: Описание некоторых команд системы контроля версий Git {#tbl:std-dir}

Команда	Описание
git init	Создание основного дерева репозитория
git pull	Получение обновлений(изменений текущего дерева из центрального репозитория
git push	Отправка всех произведённых изменений локального дерева в центральный репозиторий
git status	Просмотр списка изменённых файлов в текущей директории
git diff	Просмотр текущих изменений
git add .	Добавление все изменённые и/или созданные файлы и/или каталоги
git rm имена файлов	Удаление файлов и/или каталогов из индекса репозитория
git commit -am 'Описание коммита'	Сохранение всех добавленных изменений и всех изменённых файлов
git commit	Сохранение добавленных изменений с внесением комментария через встроенный редактор
git checkout -b имя ветки	Создание новой ветки, базирующейся на текущей
git branch -d имя ветки	Удаление локальной уже слитой с основным деревом ветки
git branch -D имя ветки	Принудительное удаление локальной ветки

## Описываем выполнение лабораторной работы

```
Изс # Выполнение лабораторной работы

My: Первоначально установим гит(рис. @fig:001):

Na ! [Установка системы контроля версий GIT](/home/narogozhina/work/study/2022-2023/
    Операционные системы/os-intro/labs/lab02/report/image/1.jpg){#fig:001 width=70%}

Производим первичную настройку параметров git, а именно:

1. Зададим имя и email владельца репозитория(рис. @fig:002):

! [Первичная настройки git](/home/narogozhina/work/study/2022-2023/Операционные системы/
    os-intro/labs/lab02/report/image/2.jpg){#fig:002 width=70%}
```

Рис. 6: Выполнение ЛР

Описываем выводы и отвечаем на контрольные вопросы:

## # Выводы

В ходе данной лабораторной работы мы приобрели навыки работы с системой контроля версий git.

## # Контрольные вопросы

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

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 проекта.

При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещать изменения, произведённые разными участниками проекта, производить откат к любой более ранней версии проекта, если это требуется.

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

**\*\*Хранилище\*\*** - место хранения всех версий и служебной информации.



Добавляем ссылки на источники:

```
# Список литературы{.unnumbered}

1. [Руководство по лабораторной работе №2] (https://esystem.rudn.ru/mod/page/view.php?id=970819#org3b74ce0)

2. [Основная информация LINUX] (https://uii.bitbucket.io/study/courses/cs/lecture02\_vcs.slides-2021.pdf)

3. [Система контроля версий. Горвиц Евгений] (https://glebradchenko.susu.ru/courses/bachelor/engineering/2016/SUSU\_SE\_2016\_REP\_3\_VCS.pdf)

4. [Руководство пользования GIT] (https://docs.github.com/ru/get-started/getting-started-with-git/ignoring-files)
```

Рис. 9: Ссылки

- Таким образом, в ходе лабораторной работы, я научилась пользоваться языком разметки Markdown и оформила отчёты по предыдущим лабораторным работам с его помощью.