

Отчёт по лабораторной работе №3

Дисциплина: Операционные системы

Шошина Евгения Александровна, группа: НКАбд-03-22

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
3.1	Базовые сведения о Markdown	7
3.2	Обработка файлов в формате Markdown	8
3.3	Оформление отчета по лабораторной работе	8
3.4	Структура отчёта	9
3.5	Содержание основных элементов отчета	9
4	Выполнение лабораторной работы	11
5	Выводы	16

Список иллюстраций

fignoЦели и задачи Лабораторной работы	11
fignoВыполнение лабораторной работы	12
fignosкриншоты	13
fignovывод	13
fignoОтвет на контрольные вопросы	14
fignosчет в форматах pdf и docx	15
fignoЗагрузка на GitHub	15

Список таблиц

1 Цель работы

Научиться оформлять отчёты с помощью легковесного языка разметки Markdown

2 Задание

- 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.)

3 Теоретическое введение

3.1 Базовые сведения о Markdown

- 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. Общий формат огражденных блоков кода

- Верхние (\sim) и нижние (\wedge) индексы
- Внутритекстовые формулы делаются аналогично формулам LaTeX. Например, формула $\sin^2(x) + \cos^2(x) = 1$ запишется как `sin2(x) + cos2(x) = 1`.
Выключные формулы: `{#eq:eq:sin2+cos2}` со ссылкой в тексте “Смотри формулу”

3.2 Обработка файлов в формате Markdown

Для обработки файлов в формате Markdown будем использовать Pandoc <https://pandoc.org/>. 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: `pandoc README.md -o README.pdf` или так `pandoc README.md -o README.docx`

3.3 Оформление отчета по лабораторной работе

Лабораторная работа является небольшой научно-исследовательской работой, которую и оформлять следует по всем утверждённым требованиям. При подготовке отчета по лабораторной работе вы освоите ряд важных элементов, которые в дальнейшем пригодятся вам при написании курсовой и дипломной работы.

3.4 Структура отчёта

Согласно ГОСТ 7.32-2001, любая научно-исследовательская работа должна обязательно содержать следующие элементы: – титульный лист; – реферат; – введение; – основную часть; – заключение. Также ГОСТ рекомендует включить в работу и такие элементы: – список исполнителей; – содержание; – нормативные ссылки; – определения; – обозначения и сокращения; – список использованных источников; – приложения. Если вы проводите сложную работу, выполняемую в несколько этапов, то вам может понадобиться включить в работу часть или все элементы второго списка.

3.5 Содержание основных элементов отчета

– 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 развернутое 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 понадобится 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 проведенной работы.

4 Выполнение лабораторной работы

4.0.0.0.1 Открыли ранее созданный файл lab02 и приступили к созданию отчета

Цель работы

- 1) Изучить идеологию и применение средств контроля версий.
- 2) Освоить умения по работе с [git](#).

Задание

- 1) Создать базовую конфигурацию для работы с [git](#).
- 2) Создать ключ [SSH](#).
- 3) Создать ключ [PGP](#).
- 4) Настроить подписи [git](#).
- 5) Зарегистрироваться на [Github](#).
- 6) Создать локальный каталог для выполнения заданий по предмету.

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 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.

В классических системах контроля версий используется централизованная модель, предполагающая наличие единого [репозитория](#) для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную ему версию файлов. После внесения изменений, пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент. Сервер может сохранять не полную версию изменённых файлов, а производить так называемую дельта-компрессию – сохранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных.

Цели и задачи Лабораторной работы

4.0.0.0.2 Написали цели и задачи Лабораторной работы

```

# Выполнение лабораторной работы
### Установка программного обеспечения
Установка git - dnf install git
Установка gh - dnf install gh
! [Рис. 1.2 Установка gh - dnf install gh] (image/1.2.png)

### Базовая настройка git
Зададим имя и email владельца репозитория:
! [Рис. 2.2 Имя и email владельца репозитория] (image/2.2.png)
Настроим utf-8 в выводе сообщений git:
! [Рис. 3.2 utf-8 в выводе сообщений git] (image/3.2.png)
Настройте верификацию и подписание коммитов git (см. Верификация коммитов git с помощью GPG).
Зададим имя начальной ветки (будем называть её master):
! [Рис. 4.2 имя начальной ветки] (image/4.2.png)
Параметр autocrlf:
! [Рис. 5.2 Параметр autocrlf] (image/5.2.png)
Параметр safecrlf:
! [Рис. 6.2 Параметр safecrlf:] (image/6.2.png)

### Создали ключи ssh
по алгоритму rsa с ключом размером 4096 бит (ssh-keygen -t rsa -b 4096)
по алгоритму ed25519 (ssh-keygen -t ed25519)

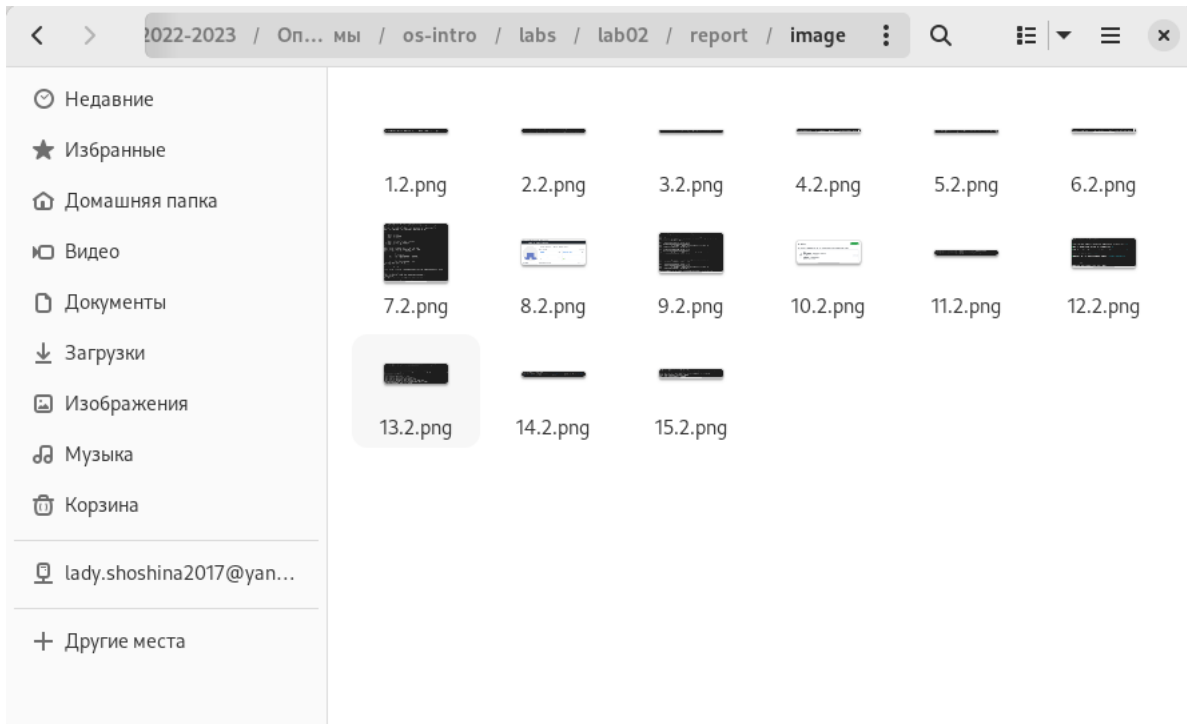
### Создали ключи pgp
Генерируем ключ
! [Рис. 7.2 Генерация ключа] (image/7.2.png)

### Настройка github
Создали учётную запись на https://github.com.
Заполнили основные данные на https://github.com.
! [Рис. 8.2 Создание учётной записи] (image/8.2.png)

```

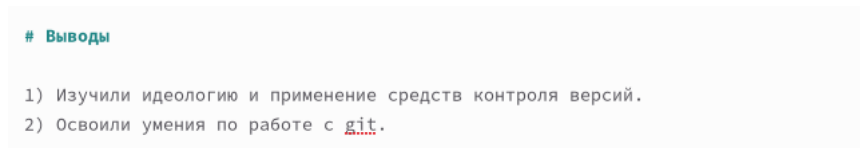
Выполнение лабораторной работы

4.0.0.0.3 Выполнили лабораторную работу



СКРИНШОТЫ

4.0.0.0.4 Сделали скриншоты выполнения по мере выполнения заданий



ВЫВОД

4.0.0.0.5 Написали вывод

Контрольные вопросы

1) Система контроля версий (Version Control System, VCS)
 – программное обеспечение для облегчения работы с изменяющейся информацией.

Система контроля версий (Version Control System, VCS)
 – это система, регистрирующая изменения в одном или нескольких файлах с тем, чтобы в дальнейшем была возможность вернуться к определённым старым версиям этих файлов.

VCS позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое.

2) Хранилище (repository), или репозиторий,
 – место хранения файлов и их версий, служебной информации.

Commit («[трудовой] вклад», не переводится)
 – процесс создания новой версии; иногда синоним версии.

Версия (revision), или ревизия,
 – состояние всего хранилища или отдельных файлов в момент времени («пункт истории»).

Рабочая копия (working copy)
 – текущее состояние файлов проекта (любой версии), полученных из хранилища и, возможно, изменённых.

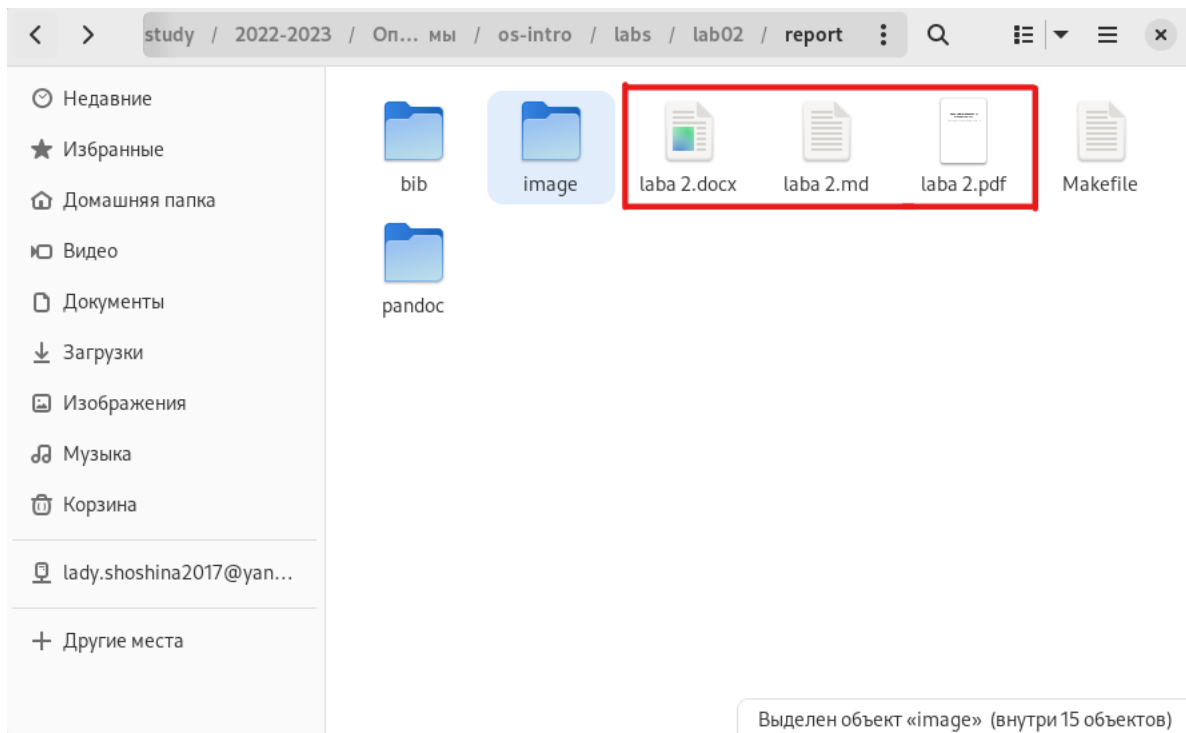
Хранилище – репозиторий – место хранения всех версий и служебной информации.
Commit – это команда для записи индексированных изменений в репозиторий.
 История – место, где сохраняются все коммиты, по которым можно посмотреть данные о коммитах.
 Рабочая копия – текущее состояние файлов проекта, основанное на версии, загруженной из хранилища.

3) Централизованные VCS:
 Одно основное хранилище всего проекта. Каждый пользователь копирует себе необходимые ему файлы из этого репозитория, изменяет и, затем, добавляет свои изменения обратно (Subversion, CVS, TFS, VAULT, AccuRev)

Децентрализованные VCS:
 У каждого пользователя свой вариант (возможно не один) репозитория
 Присутствует возможность добавлять и забирать изменения из любого репозитория (Git, Mercurial, Bazaar)

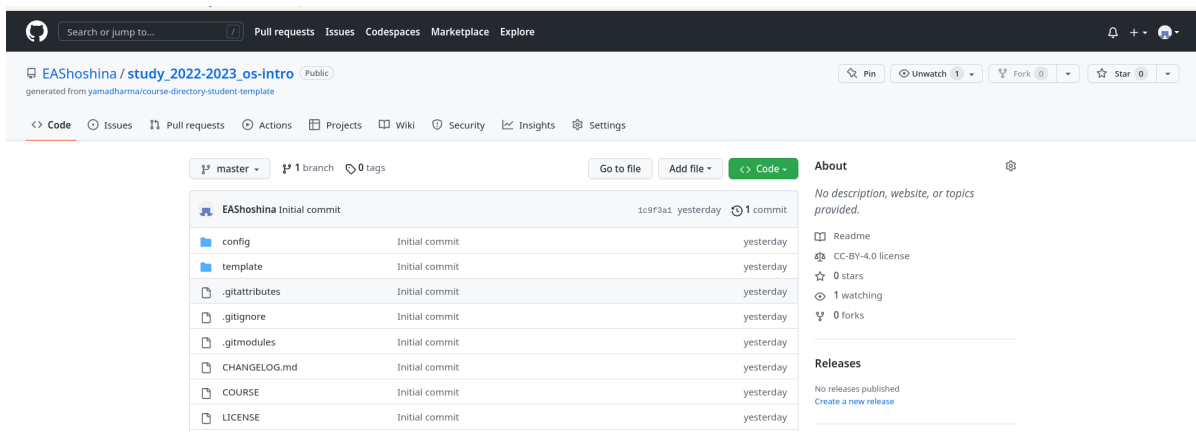
Ответ на контрольные вопросы

4.0.0.0.6 Ответили на контрольные вопросы



отчет в форматах pdf и docx

4.0.0.0.7 Сохранили отчет и конвертировали в формат pdf и docx



Загрузка на GitHub

4.0.0.0.8 Загрузили на GitHub

5 Выводы

Научились оформлять отчёты с помощью легковесного языка разметки Markdown.