

Отчет по лабораторной работе №3

Алмазова Елизавета Андреевна

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

Цель работы и задание

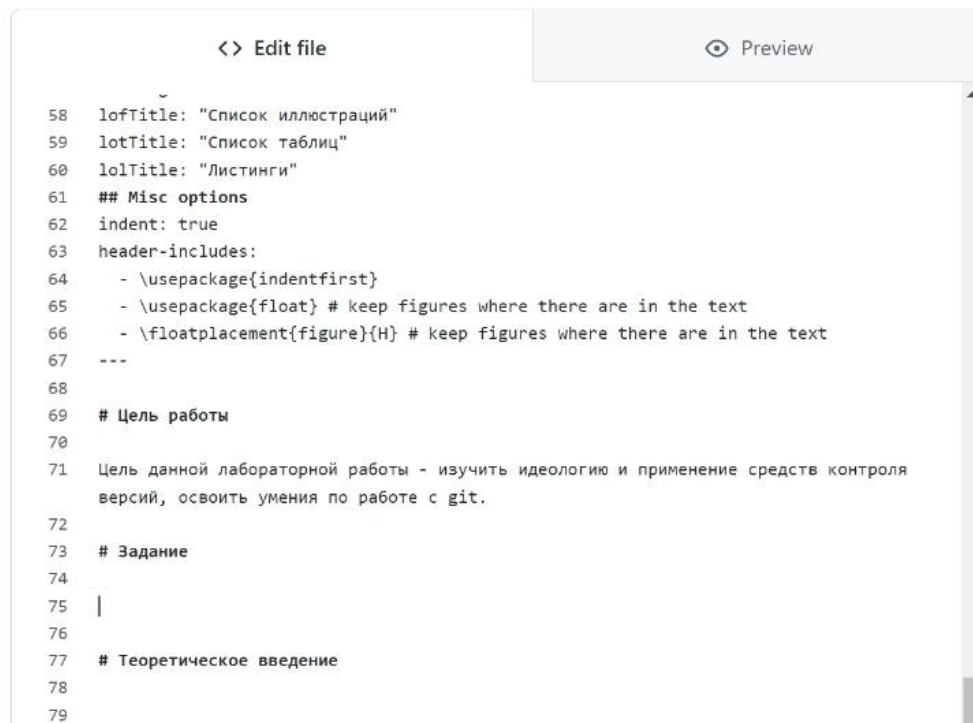
Цель данной лабораторной работы - научиться оформлять отчёты с помощью легковесного языка разметки Markdown. Задание - сделать отчёт по предыдущей лабораторной работе в формате Markdown в трех форматах: md, pdf и docx.

Ход работы. Редактирование шаблона

Я отредактировала шаблон для отчета в соответствии с требованиями лабораторной работы №3 (рис.1).

study_2021-2022_os-intro / labs / lab02 / report / report.md

in master



```
58 lofTitle: "Список иллюстраций"
59 lotTitle: "Список таблиц"
60 lolTitle: "Листинги"
61 ## Misc options
62 indent: true
63 header-includes:
64   - \usepackage[indentfirst]
65   - \usepackage{float} # keep figures where there are in the text
66   - \floatplacement{figure}{H} # keep figures where there are in the text
67 ---
68
69 # Цель работы
70
71 Цель данной лабораторной работы - изучить идеологию и применение средств контроля
72 версий, освоить умения по работе с git.
73
74 # Задание
75 |
76
77 # Теоретическое введение
78
79
```

Рисунок 1 - Редактирование шаблона.

Ход работы. Заполнение шаблона

Я заполнила шаблон, основываясь на уже выполненном ранее отчете (рис. 2,3).

263 - Создание новой ветки, базирующейся на текущей - `git checkout -b [имя ветки]`;
264 - Переключение на некоторую ветку - `git checkout [имя ветки]`;
265 - Отправка изменений конкретной ветки в центральный репозиторий - `git push origin [имя ветки]`;
266 - Слияние ветки с текущим деревом - `git merge -no-ff [имя ветки]`;
267 - Удаление ветки - `git branch -d [имя ветки]` (удаление локальной ветки, уже слитой с основным деревом)/`git branch -D [имя ветки]` (принудительное удаление локальной ветки)/ `git push origin: [имя ветки]` (удаление ветки с центрального репозитория);
268 - Проверка, на какой ветке находится пользователь - `git branch`;
269 - Завершение работы на ветке - `git flow <ветка> finish`;
270 - Инициализация структуры `git-flow` в репозитории - `git flow init`;
271 - Создание функциональной ветки - `git flow feature start feature_branch`;
272 - Окончание работы с функциональной веткой - `git flow feature finish feature_branch`;
273 - Создание ветки выпуска - `git flow release start 1.0.0`;
274 - Окончание работы с веткой выпуска - `git flow release finish 1.0.0`;
275 - Создание ветки исправления - `git flow hotfix start hotfix_branch`;

Рисунок 2 - Создание списка команд git.

<> Edit file

Preview

```
Другие пользователи также могут видеть внесенные изменения. Примеры: CVS, Sub-
version.
199 В распределенных системах центральный репозиторий не является обя-
зательным, т.е.
каждый пользователь пользуется полной копией проекта со всеми версиями всех файлов.
Изменения вносятся группами (chang-eset вместо commit). Примеры: Git, Bazaar,
Mercurial.
200 4. Опишите действия с VCS при единоличной работе с хранилищем.
201 - Создаем локальный репозиторий.
202 - Сначала делаем предварительную конфигурацию, указывая имя и email с помощью
команд git config --global user.name "<Name Surname>", git config --global user.email
"<email>" соответственно.
203 - Настраиваем utf-8 в выводе сообщений git: git config --global
core.quotePath false.
204 - С помощью команды cd переходим в папку, где хотим создать репозито-
рий, с помощью mkdir создаем каталог, переходим в него с помощью ко-манды cd и командой git
init создаем основное дерево репозитория - по-является каталог .git для хранения
истории изменений.
205 Файлы добавляются в локальный репозиторий так:
206 - git add [имя файла]
207 | git commit -am '<Комментарий>'
208 Git status покажет изменения в рабочем каталоге с последней ревизии.
209 Иногда создаются файлы, которые не требуется добавлять в репозиторий. Тогда можно
прописать шаблоны игнорируемых файлов в файл .gitignore в корневом каталоге
репозитория. Шаблоны можно прописать с помощью сервисов, для этого нужно получить
список имеющихся шаблонов с помо-щью curl -L -s https://www.gitignore.io/api/list,
затем скачать шаблон, например, на C++ curl -L -s https://www.gitignore.io/api/c++ >>
.gitignore.
```

Рисунок 3 - Процесс работы над оформлением.

Ход работы. Сохранение файлов

Я сделала коммит изменений в файле отчета ко второй лабораторной работе, предварительно проверив файл с помощью preview (рис.4), а затем перевела его в форматы docx и pdf (рис.5).

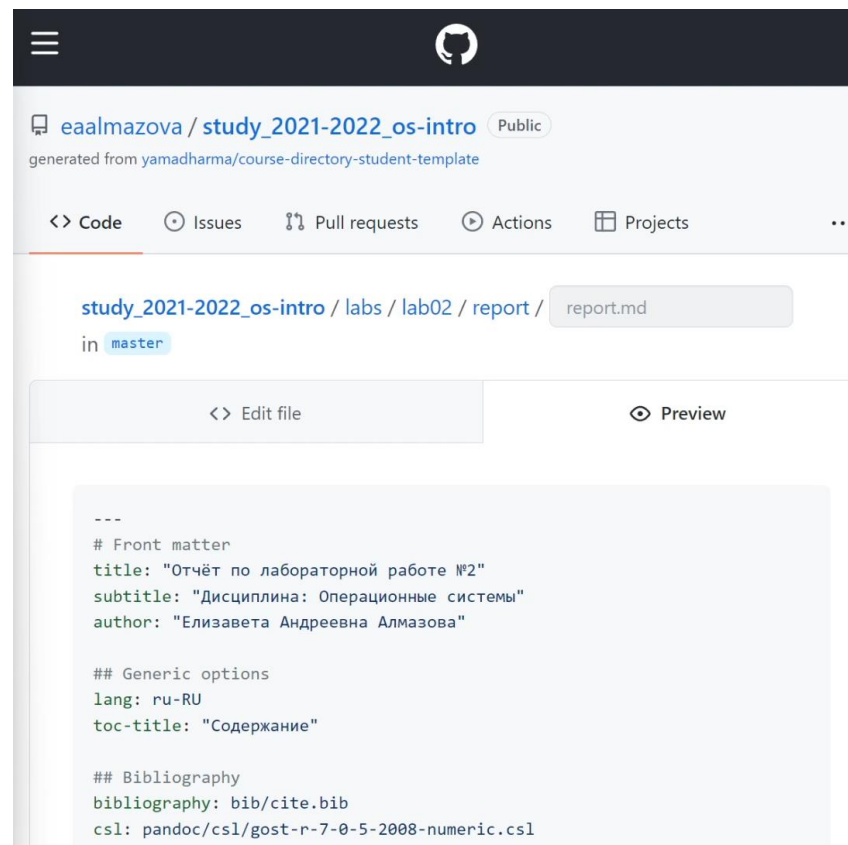


Рисунок 4 - Preview report.md

Отчёт по лабораторной работе №2

Дисциплина: Операционные системы

Елизавета Андреевна Алмазова

Содержание

Цель работы

Задание

Теоретическое введение

Выполнение лабораторной работы

Выводы

Ответы на контрольные вопросы

Цель работы

Цель данной лабораторной работы - изучить идеологию и применение средств контроля версий, освоить умения по работе с git.

Задание

- Создать базовую конфигурацию для работы с git;
- Создать ключ SSH;
- Создать ключ PGP;
- Настроить подписи git;

Рисунок 5 - Скриншот части готового файла report.pdf

Выводы

В ходе выполнения данной лабораторной работы я сделала отчет по предыдущей лабораторной работе в формате Markdown и создала отчет в двух других форматах: docx и pdf.

Спасибо за внимание!