Отчёт по лабораторной работе №2

Система контроля версий Git

Мальянц В. К.

Содержание

# 1 Цель работы

Изучение идеологии применения средств контроля версий. Приобретение практических навыков о работе с системой git.

# 2 Задание

4.1 Настройка github. 4.2 Базовая настройка git. 4.3 Создание SSH-ключа. 4.4 Создание рабочего пространства и репозитория курса на основе шаблона. 4.5 Создание репозитория курса на основе шаблона. 4.6 Настройка каталога курса. 4.7 Выполнение заданий для самостоятельной работы.

# 3 Теоретическое введение

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. Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Можно объединить (слить) изменения, сделанные разными участниками (автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменения. В зависимости от настроек блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая таким образом, привилегированный доступ только одному пользователю, работающему с файлом. 7Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно доступна информация о том, кто из участников, когда и какие изменения вносил. Обычно такого рода информация хранится в журнале изменений, доступ к которому можно ограничить. В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным. Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.

# 4 Выполнение лабораторной работы

## 4.1 Настройка github

Создала учетную запись на сайте https://github.com/ и заполнила свои данные (рис.1).



Рис. 1: Создание учетной записи на github

## 4.2 Базовая настройка git

Открываю терминал и делаю предварительную конфигурацию git. Ввожу команды: git config –global user.name “” и указываю в ней имя владельца репозитория (свое имя), git config –global user.email “” и указываю в ней почту владельца репозитория (свою почту) (рис. 2) (рис. 3).

Рис. 2: Предварительная конфигурация git (имя владельца репозитория)

Рис. 2: Предварительная конфигурация git (имя владельца репозитория)

Рис. 3: Предварительная конфигурация git (почта владельца репозитория)

Рис. 3: Предварительная конфигурация git (почта владельца репозитория)

Настраиваю utf-8 в выводе сообщений git (рис. 4).

Рис. 4: Настройка кодировки

Рис. 4: Настройка кодировки

Задаю имя начальной ветки, называю ее master (рис. 5).

Рис. 5: Создание имени начальной ветки

Рис. 5: Создание имени начальной ветки

Параметр autocrlf (рис. 6).

Рис. 6: Параметр autocrlf

Рис. 6: Параметр autocrlf

Параметр safecrlf (рис. 7).

Рис. 7: Параметр safecrlf

Рис. 7: Параметр safecrlf

## 4.3 Создание SSH ключа

Генерирую пару ключей для идентификации пользователя на сервере репозиториев (рис. 8).

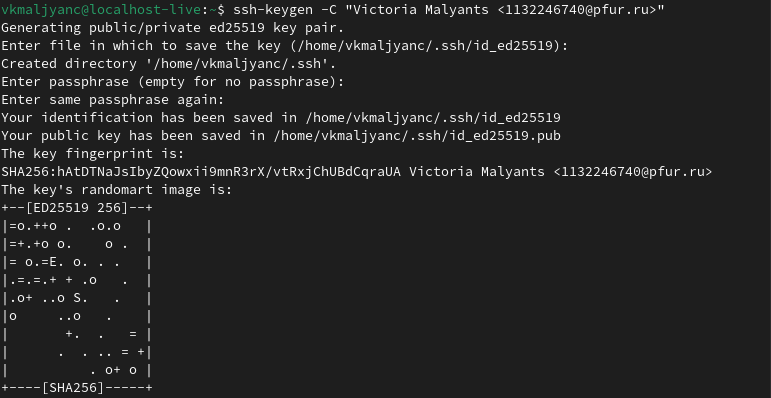


Рис. 8: Генерация SSH-ключа

Вывожу содержимое файла с открытым ключом с помощью команды cat (рис. 9).

Рис. 9: Вывод содержимого файла с открытым ключом

Рис. 9: Вывод содержимого файла с открытым ключом

Вставляю ключ в окно New SSH key, указываю имя ключа (рис. 10).

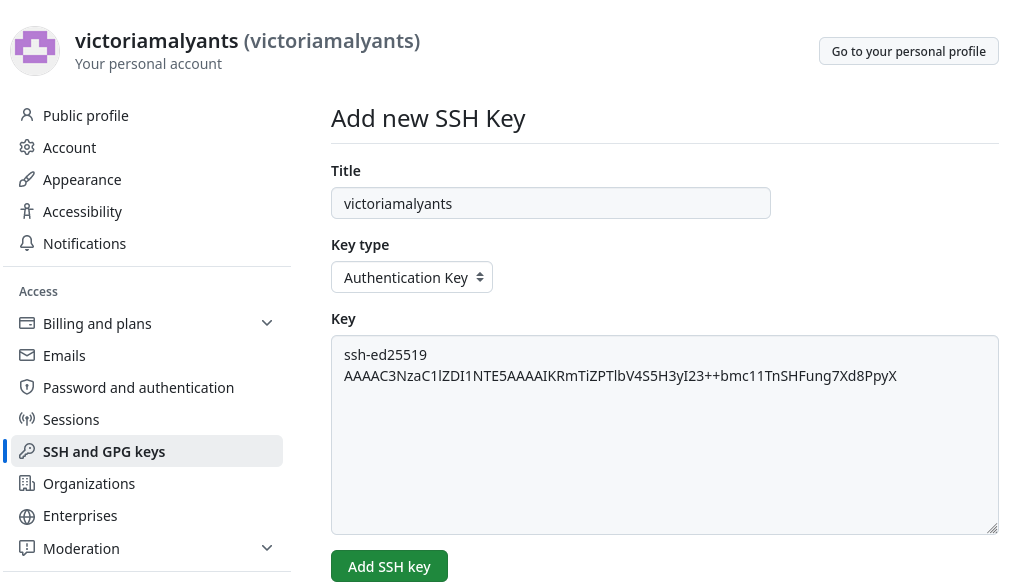


Рис. 10: Добавление ключа

Убеждаюсь в том, что SSH key создан (рис. 11).

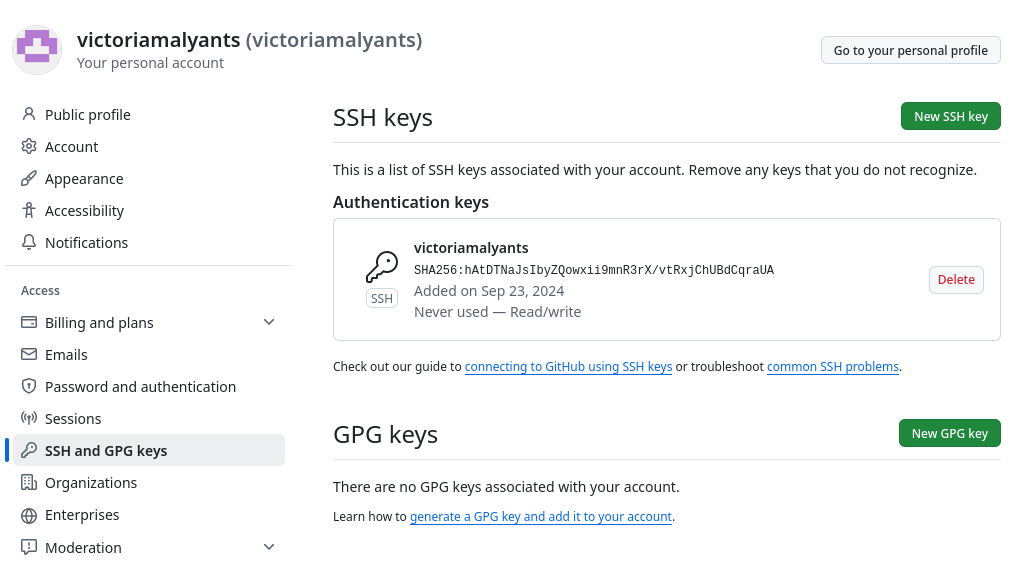


Рис. 11: Созданный SSH-ключ

## 4.4 Создание рабочего пространства и репозитория курса на основе шаблона

Создаю каталог для предмета «Архитектура компьютера» с помощью mkdir -p (рис. 12).

Рис. 12: Создание рабочего пространства

Рис. 12: Создание рабочего пространства

## 4.5 Создание репозитория курса на основе шаблона

Перехожу на страницу репозитория с шаблоном курса https://github.com/yamadharma/course-directory-student-template. Задаю имя репозитория и создаю его (рис. 13) (рис. 14).

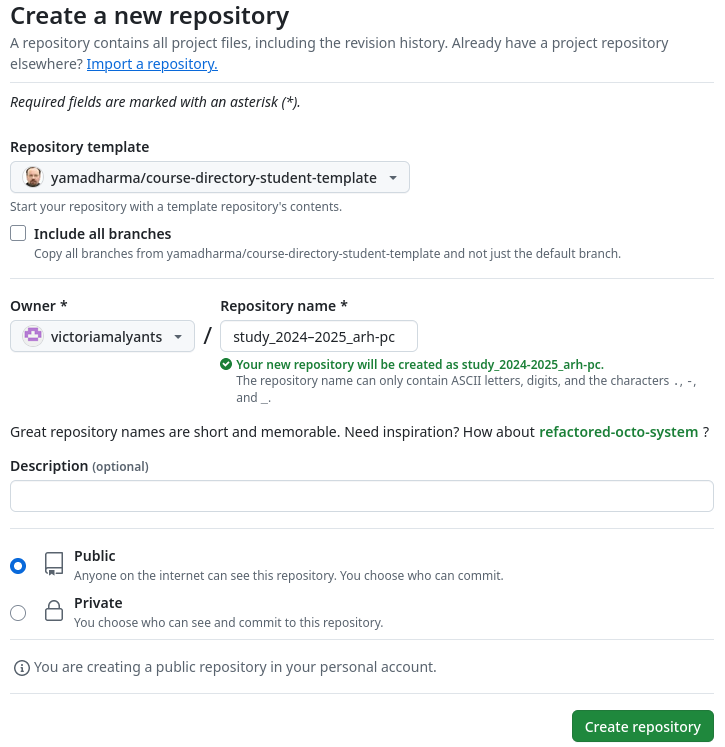


Рис. 13: Создание репозитория

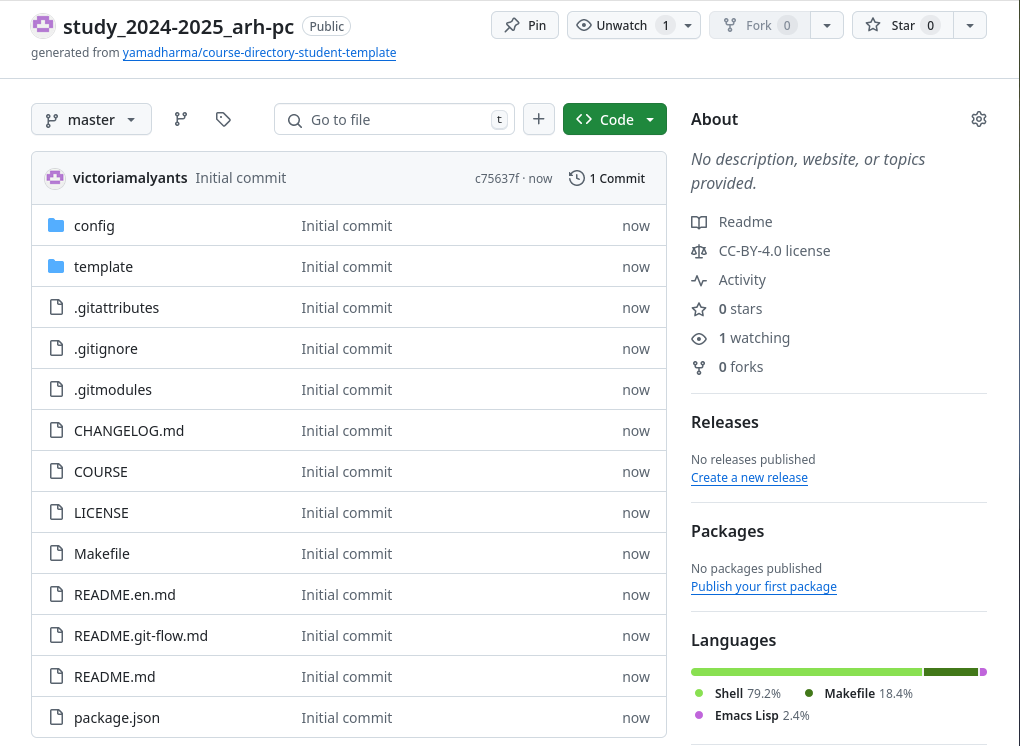


Рис. 14: Созданный репозиторий

Перехожу в каталог курса с помощью команды cd (рис. 15).

Рис. 15: Переход в каталог курса

Рис. 15: Переход в каталог курса

Клонирую созданный репозиторий (рис. 16).

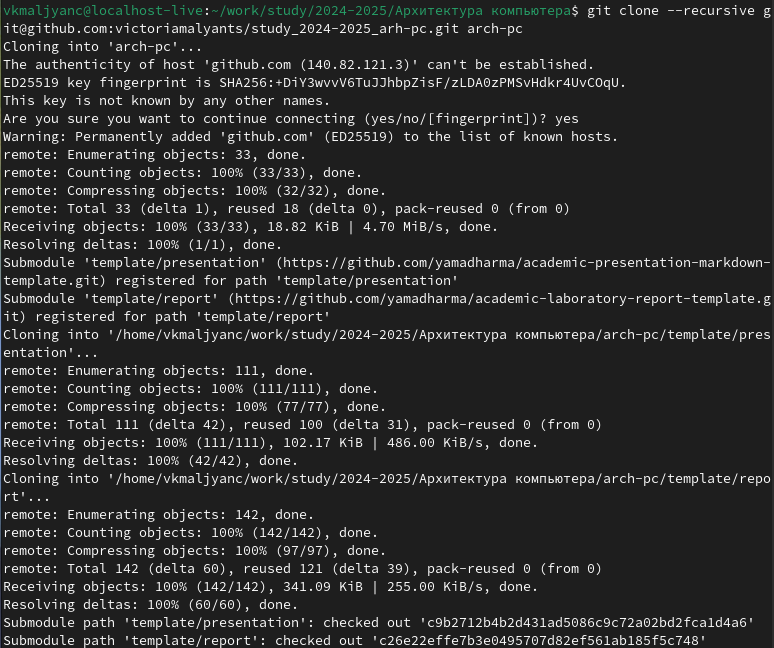


Рис. 16: Клонирование репозитория

Ссылку для клонирования копирую на странице созданного репозитория (рис. 17).

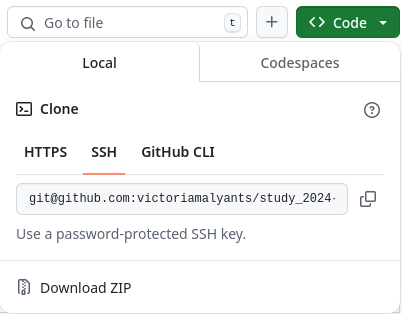


Рис. 17: Ссылка для копирования репозитория

## 4.6 Настройка каталога курса

Перехожу в каталог курса (рис. 18).

Рис. 18: Перемещение в каталог курса

Рис. 18: Перемещение в каталог курса

Удаляю лишние файлы (рис. 19).

Рис. 19: Удаление файлов

Рис. 19: Удаление файлов

Создаю необходимые каталоги (рис. 20).

Рис. 20: Создание каталогов

Рис. 20: Создание каталогов

Отправляю файлы на сервер (рис. 21) (рис. 22).

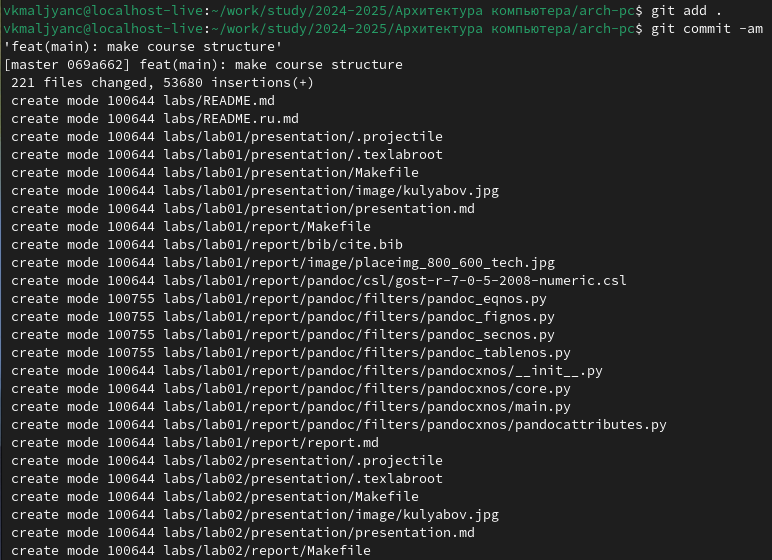


Рис. 21: Добавление и сохранение изменений на сервере

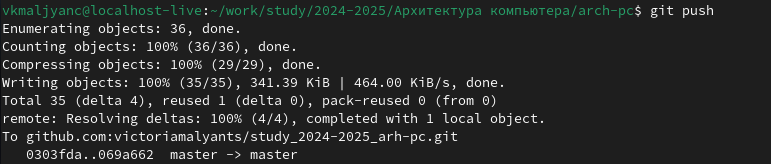


Рис. 22: Выгрузка изменений на сервер

Убеждаюсь в правильности создания иерархии рабочего пространства (рис. 23).

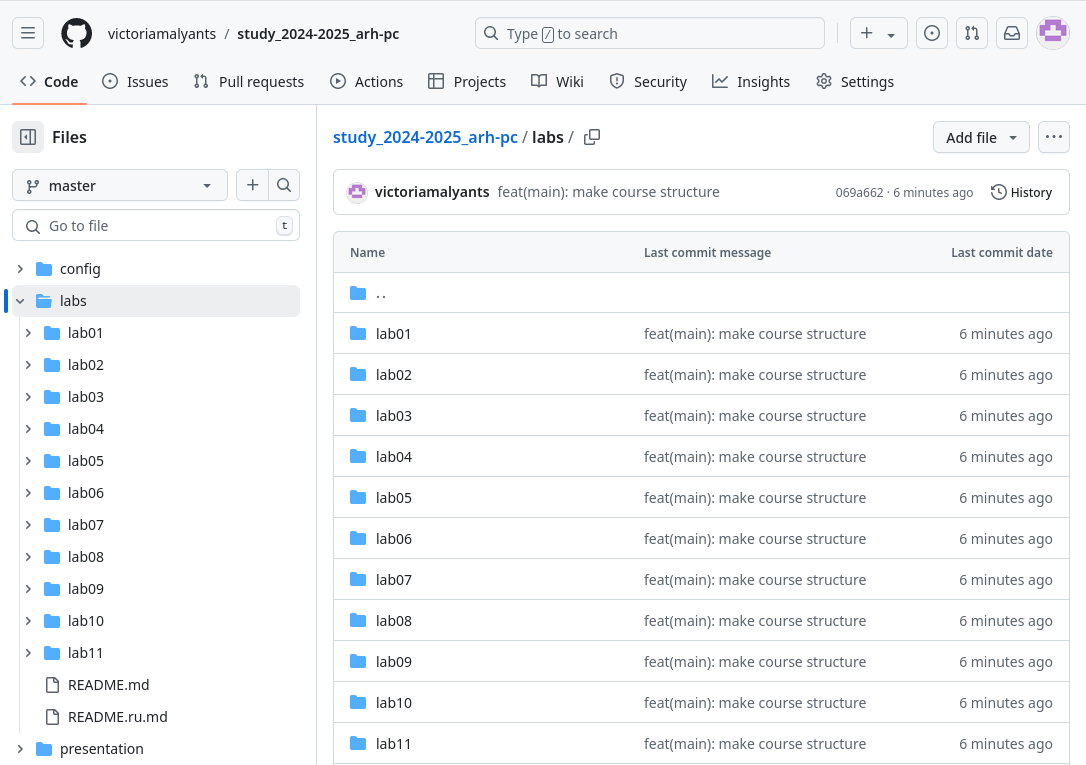


Рис. 23: Репозиторий

## 4.7 Выполнение заданий для самостоятельной работы (рис. 24) (рис. 25)

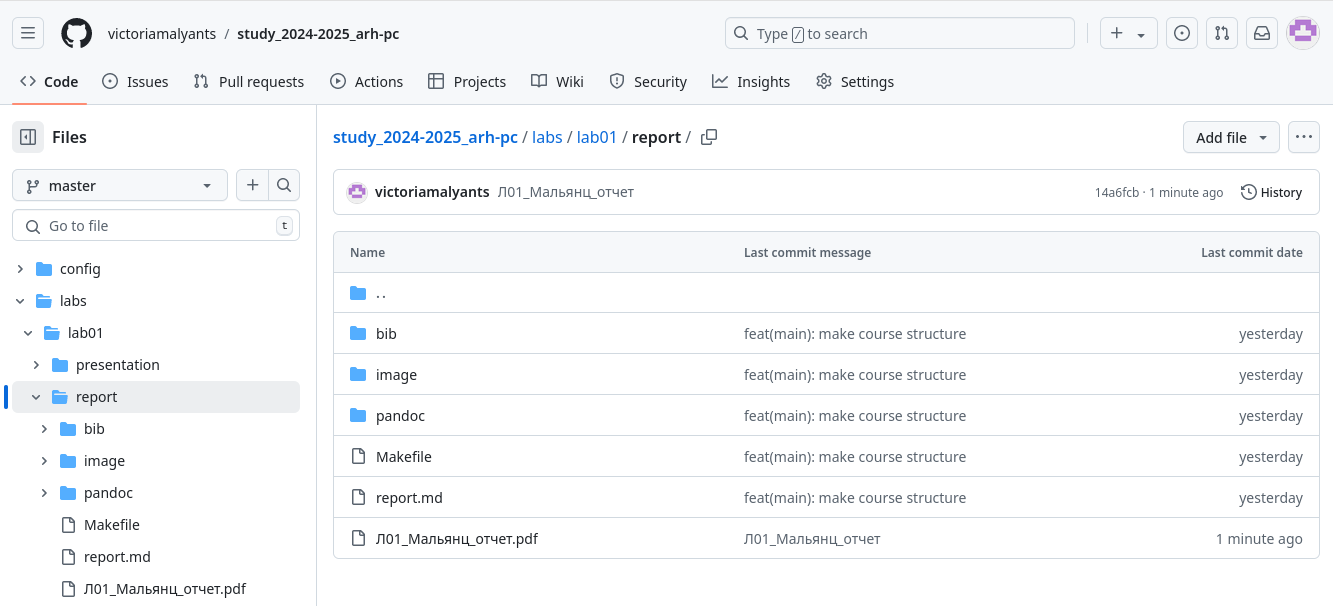


Рис. 24: Добавление отчета по Лабораторной работе № 1

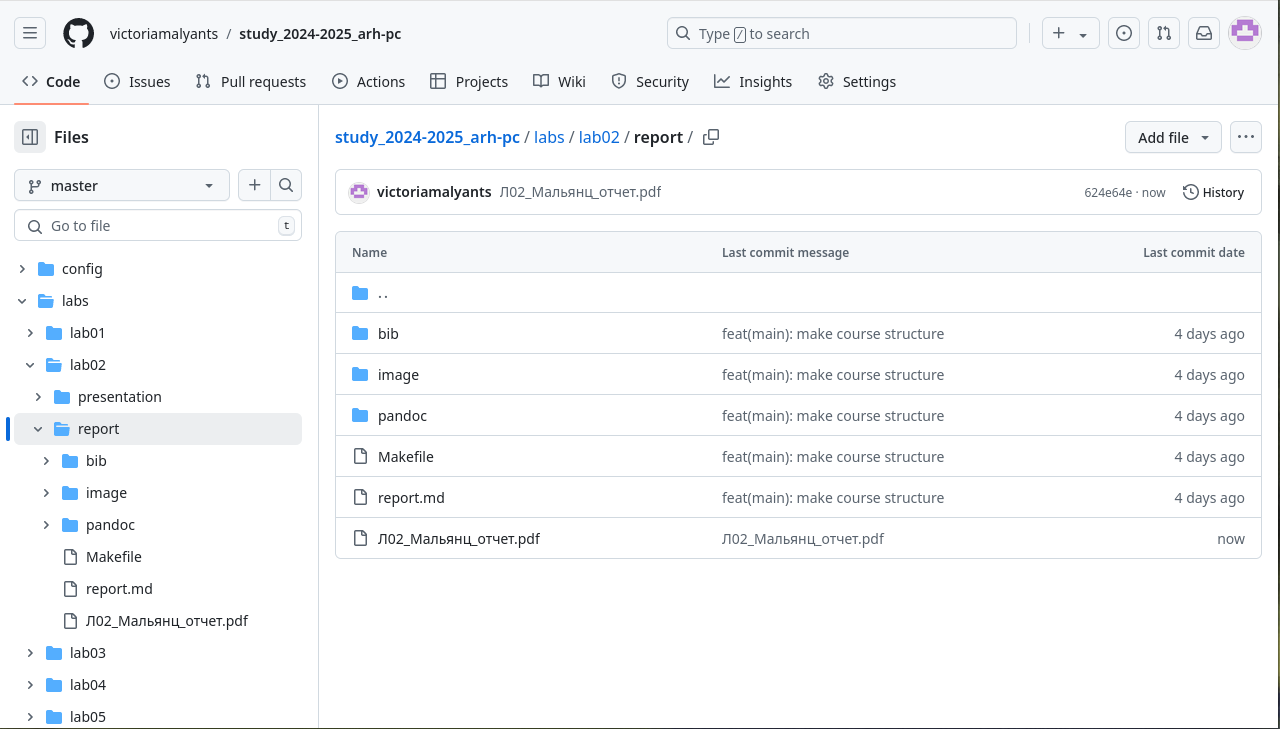


Рис. 25: Добавление отчета по лабораторной работе № 2

# 5 Вывод

Я изучила идеологию и применение средств контроля версий и приобрела практические навыки по работе с системой git.

# 6 Список литературы

1. GDB: The GNU Project Debugger. — URL: https://www.gnu.org/software/gdb/.
2. GNU Bash Manual. — 2016. — URL: https://www.gnu.org/software/bash/manual/.
3. Midnight Commander Development Center. — 2021. — URL: https://midnight-commander. org/.
4. NASM Assembly Language Tutorials. — 2021. — URL: https://asmtutor.com/.
5. Newham C. Learning the bash Shell: Unix Shell Programming. — O’Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: http://www.amazon.com/Learningbash-Shell-Programming- Nutshell/dp/0596009658.
6. Robbins A. Bash Pocket Reference. — O’Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL: https://www.nasm.us/docs.php.
8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
9. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
10. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс, 2017.
11. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
12. Расширенный ассемблер: NASM. — 2021. — URL: https://www.opennet.ru/docs/RUS/nasm/.
13. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВПетербург, 2010. — 656 с. — ISBN 978-5-94157- 538-1.
14. Столяров А. Программирование на языке ассемблера NASM для ОС Unix. — 2-е изд. — М. : МАКС Пресс, 2011. — URL: http://www.stolyarov.info/books/asm\_unix.
15. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
16. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — 1120 с. — (Классика Computer Science).