Лабораторная работа №2

Система контроля версии Git

Корчагин Алексей Павлович

Содержание

# 1 Цель работы

Целью работы является изучить идеологию и применение средств контроля версийю Приобрести практические навыки по работе с системой git

# 2 Теоретическое введение

Git - это набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды git с различными опциями. Git является диверсифецированной системой контроля версий. Резервную копию локального поэтому хранилища можно сделать простым копированием или архивацией

Системы контроля версий (Version Control System, VCS) используется 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.

Среди стандартных VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.

Если мы хотим сохранить все изменения в текущем каталоге, то используем:

git add .

Затем сохраняем изменения, поясняя, что было сделано:

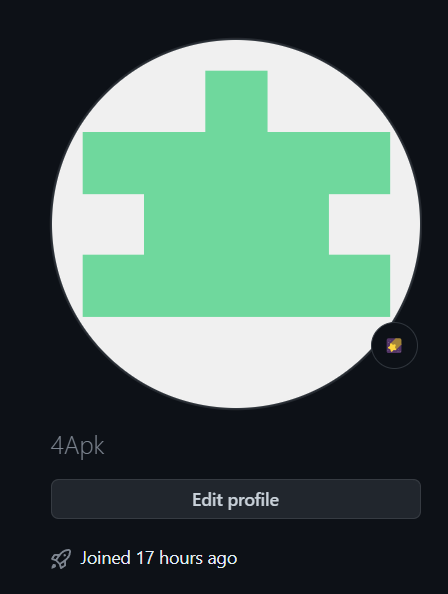
git commit -am “Some commit message”

и отправляем в центральный репозиторий:

git push origin имя\_ветки или git push

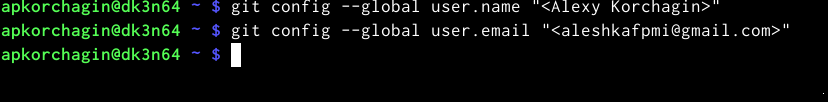
# 3 Выполнение лабораторной работы

Создал аккаунт на Github (рис. ??).



Созданный аккаунт

Открываю виртуальную машину, затем открываю терминал и делаю пред- варительную конфигурацию git. Ввожу команду git config –global user.name “”, указывая свое имя и команду git config –global user.email “work@mail”, указывая в ней электронную почту владельца, то есть мою (рис. ??).



Название рисунка

Настраиваю utf-8 в выводе сообщений git для корректного отображения сим- волов (рис. ??).

Найстройка кодировки

Найстройка кодировки

Задаю имя «master» для начальной ветки (рис. ??).

Имя для начальной ветки

Имя для начальной ветки

Задаю параметр autocrlf со значением input, так как я работаю в системе Linux, чтобы конвертировать CRLF в LF только при коммитах . CR и LF – это сим- волы, которые можно использовать для обозначения разрыва строки в текстовых файлах.(рис. ??).

Параметр autocrlf

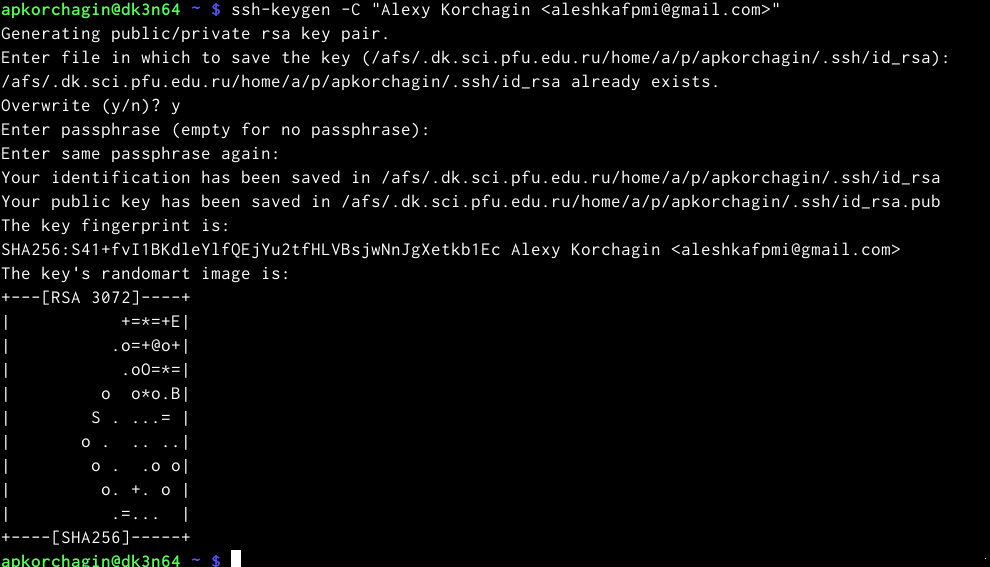
Параметр autocrlf

Задаю параметр safecrlf со значением warn, так Git будет проверять преоб- разование на обратимость. При значении warn Git только выведет предупреждение, но будет принимать необратимые конвертации. (рис. ??).

Параметр safecrlf

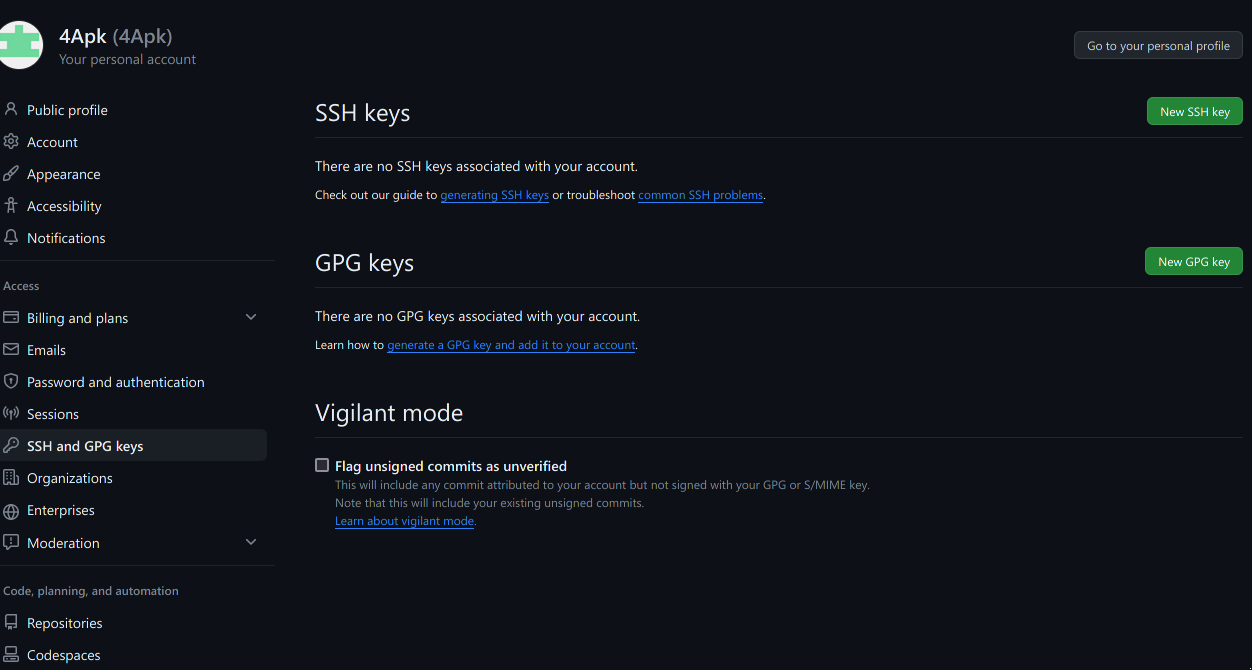
Параметр safecrlf

Для последующей идентификации пользователя на сервере репозиториев необ- ходимо сгенерировать пару ключей (приватный и открытый). Для этого ввожу команду ssh-keygen -C “Имя Фамилия, work@email”, указывая имя владельца и электронную почту владельца. Ключ автоматически сохранится в каталоге ~/.ssh/. (рис. ??).



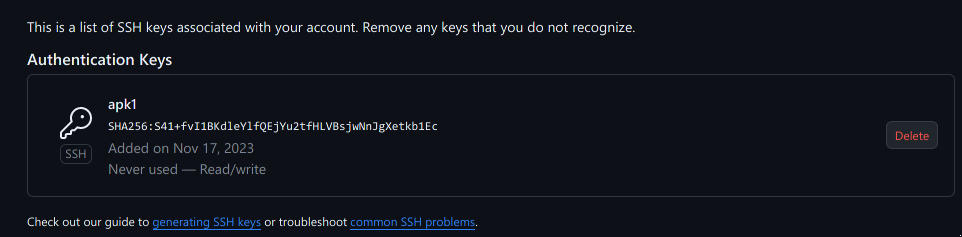
Ssh ключ

Открываю браузер, захожу на сайт GitHub. Открываю свой профиль и выбираю страницу «SSH and GPG keys». Нажимаю кнопку «New SSH key» (рис. ??).



окно SSh and GPG keys

Вставляю скопированный ключ в поле «Key». В поле Title указываю имя для ключа. Нажимаю «Add SSH-key», чтобы завершить добавление ключа (рис. ??).



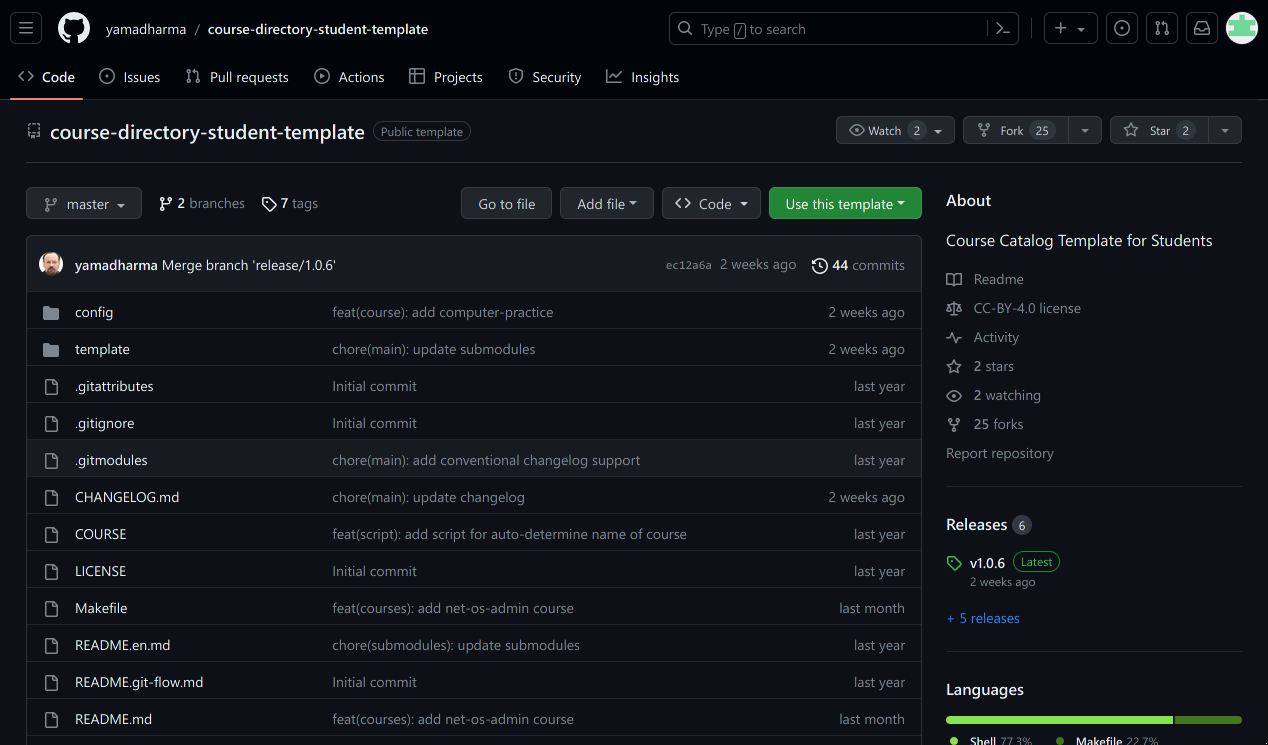
Добавил ключ

Закрываю браузер, открываю терминал. Создаю директорию, рабочее про- странство, с помощью утилиты mkdir, блягодаря ключу -p создаю все директории после домашней ~/work/study/2023-2024/“Архитектура компьютера” рекурсивно. Далее проверяю с помощью ls, действительно ли были созданы необходимые мне каталоги (рис. ??).

создание рабочего пространства

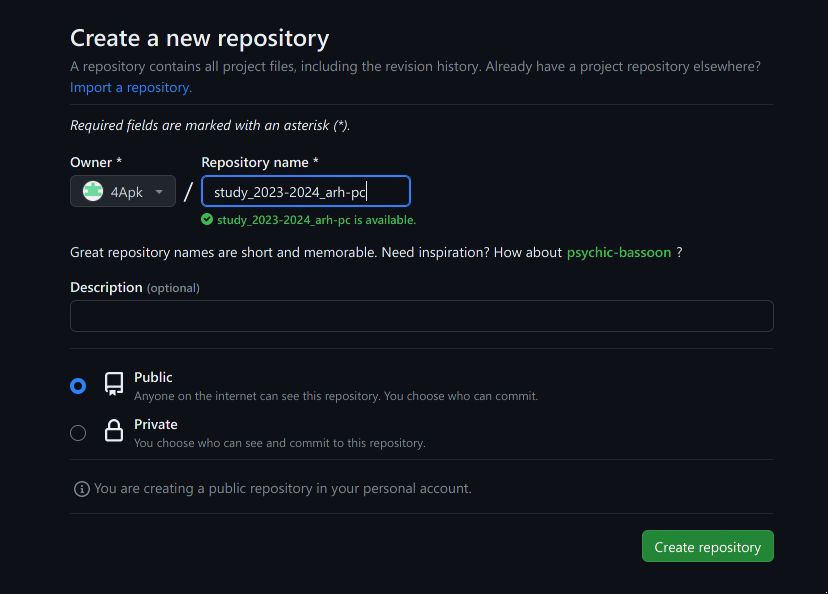
создание рабочего пространства

В браузере перехожу на страницу репозитория с шаблоном курса по адресу https://github.com/yamadharma/course-directory-student-template. Далее выбираю «Use this template», чтобы использовать этот шаблон для своего репозитория (рис. ??).



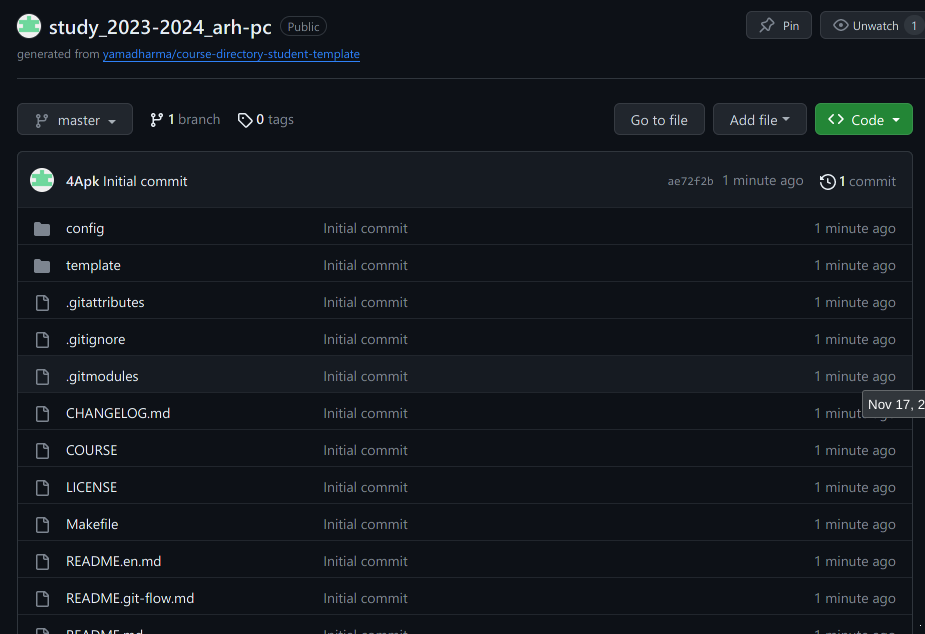
Страница шаблона для репозитория

В открывшемся окне задаю имя репозитория (Repository name): study\_2023–2024\_arh- pc и создаю репозиторий, нажимаю на кнопку «Create repository from template» (рис. ??).



окно создания репозитория

Рипозиторий создан (рис. ??).



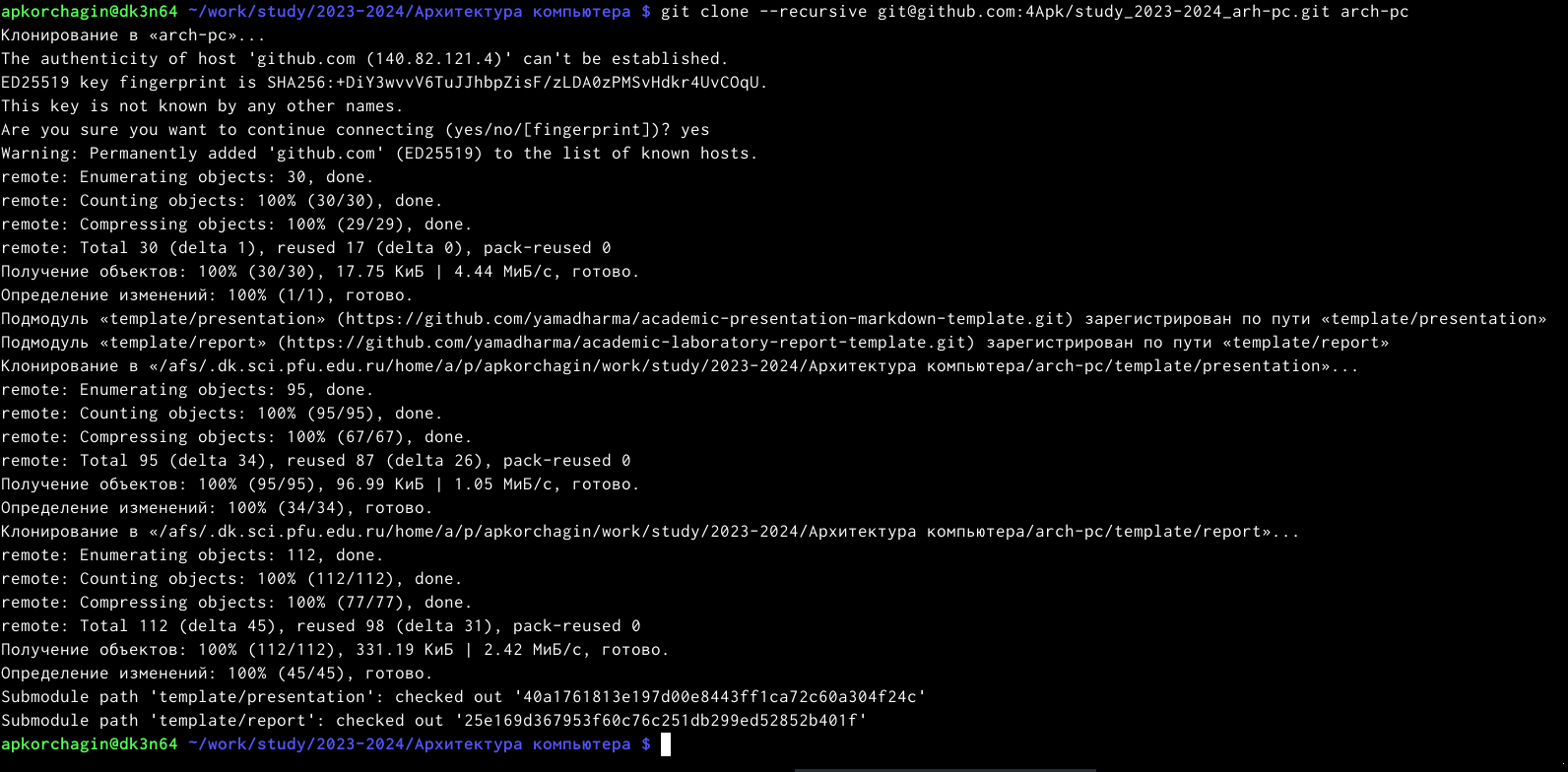
Созданный

Через терминал перехожу в созданный каталог курса с помощью утилиты cd (рис. ??).

Перемещаюсь между директориями

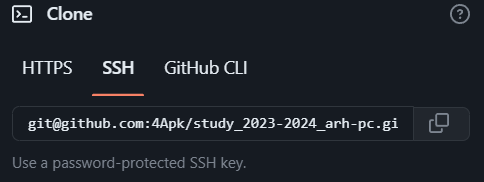
Перемещаюсь между директориями

Клонирую созданный репозиторий с помощью команды git clone –recursive git@github.com:/study\_2023–2024\_arh-pc.git arch-pc (рис. ??).



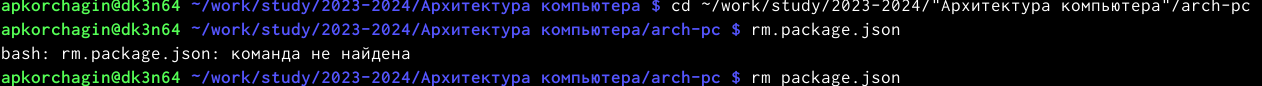
Клонирование репозитория

Копирую ссылку для клонирования на странице созданного репозитория, сначала перейдя в окно «code», далее выбрав в окне вкладку “SSH” (рис. ??).



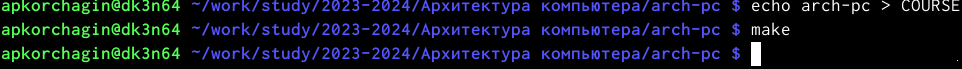
Ссылка на клонирование

Перехожу в каталог arch-pc с помощью утилиты cd и удаляю лишние файлы при помощи rm (рис. ??).



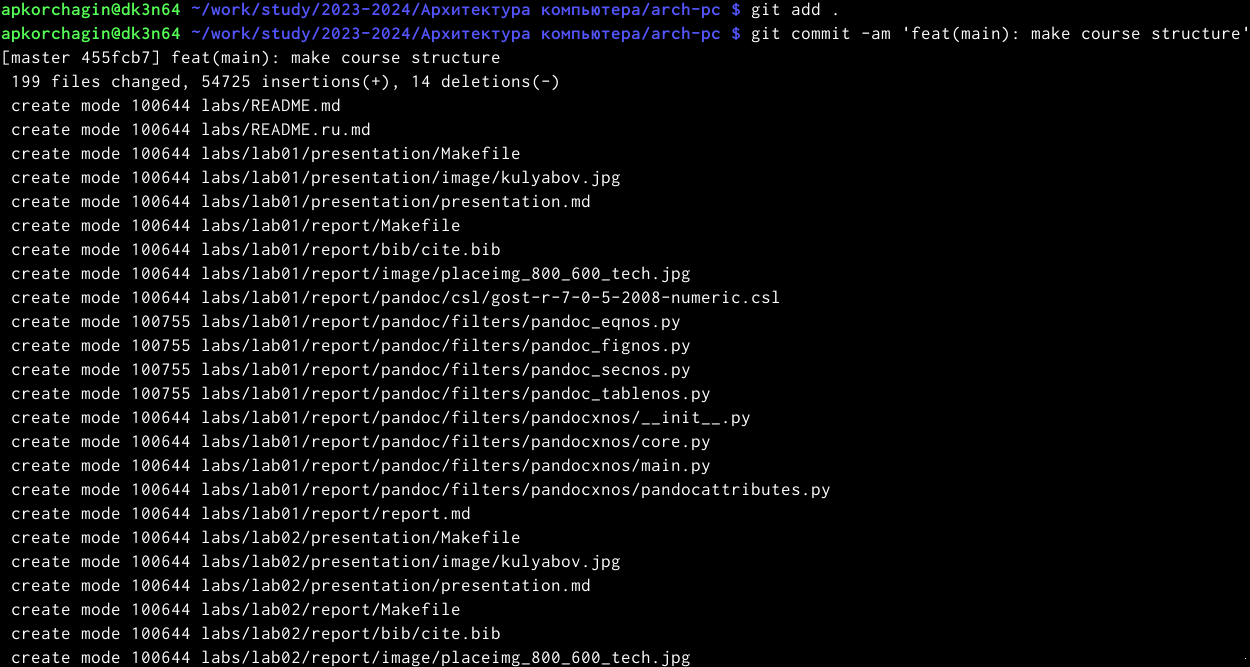
Удаление файлов

Создаю необходимые каталоги (рис. ??).



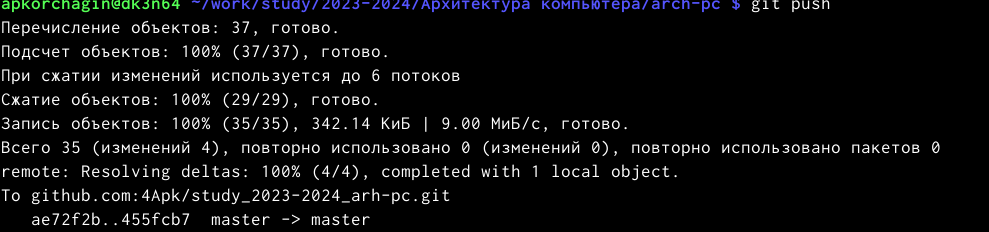
Создание каталогов

Отправляю созданные каталоги с локального репозитория на сервер: добавляю все созданные каталоги с помощью git add, комментирую и сохраняю изменения на сервере как добавление курса с помощью git commit (рис. ??).



Добавленние изменнений на сервер

Отправляю все на сервер с помощью push (рис. ??).



Выгрузка имзенений на сервер

Проверяю коректность выполнения работы сначала на самом сайте GitHub (рис. ??).



Страница репозитория

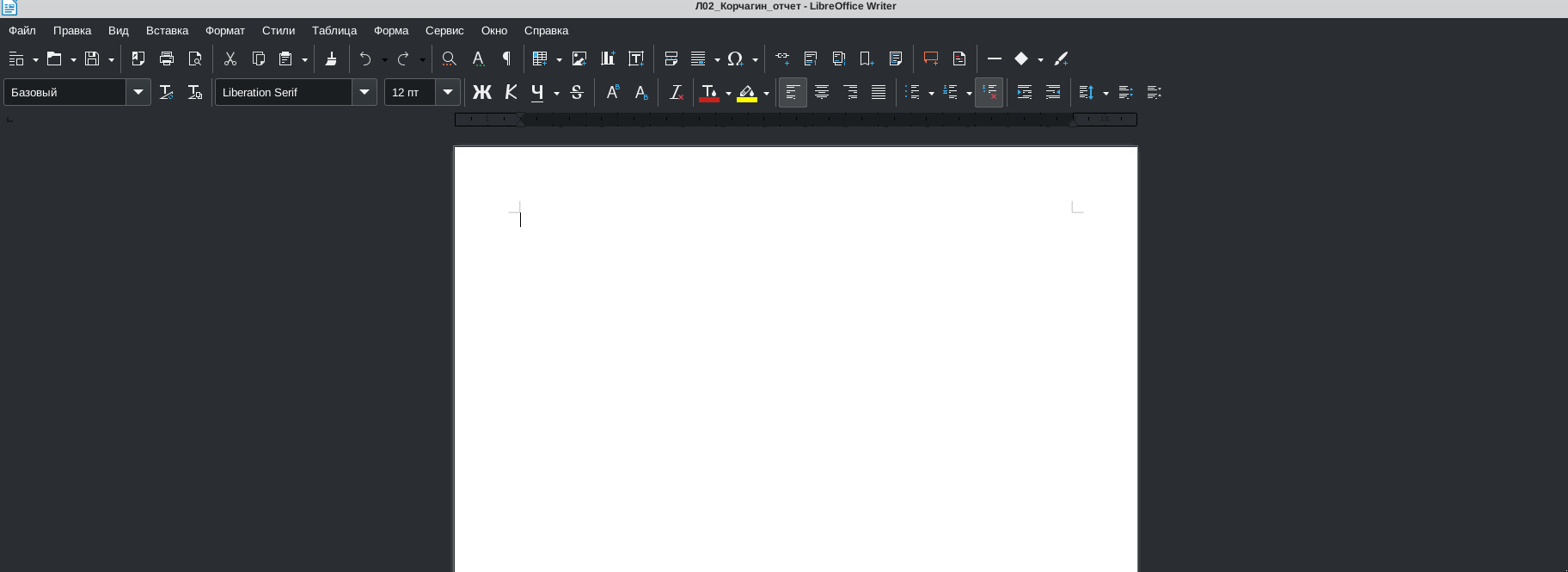
# 4 Выполнение заданий для самостоятельной работы

Перехожу в директорию labs/lab02/report с помощью утилиты cd. Создаю в каталоге файл для отчета по второй лабораторной работе с помощью утилиты touch (рис. ??).

Создание файла

Создание файла

Оформить отчет я смогу в текстовом процессоре LibreOffice Writer (рис. ??).



Текстовая работа на отчётом

Перехожу из подкаталога lab02/report в подкаталог lab01/report с помощью утилиты cd (рис. ??).

Перемещаюсь в лаб 1

Перемещаюсь в лаб 1

Проверяю местонахождение файла с отчетом по первой лабораторной работе. Он должен быть в подкаталоге домашней директории «Загрузки», для проверки использую команду ls (рис. ??).

Проверяю местополажения файлов

Проверяю местополажения файлов

Копирую первую лабораторную с помощью утилиты cp и проверяю коректность выполнения команды cp с помощью ls(рис. ??).

Скопирывал отчёт в lab1

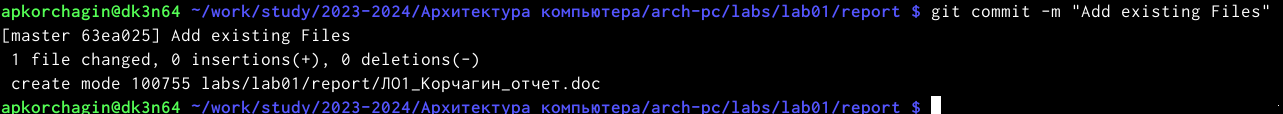
Скопирывал отчёт в lab1

Добавляю с помощью команды git add в коммит созданные файлы: Л01\_Корчагин\_отчет (рис. ??).

Добовляю лаб 1 на гит

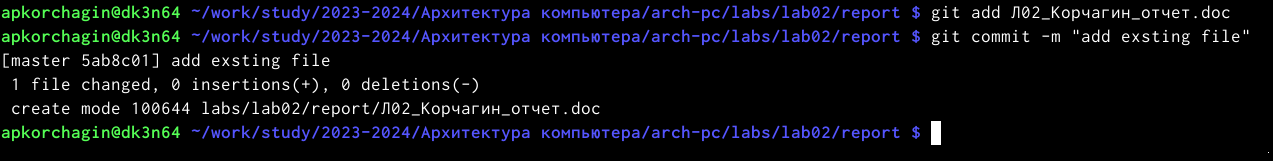
Добовляю лаб 1 на гит

Сохраняю изменения на сервере командой git commit -m “…”, поясняя, что добавила файлы (рис. ??).



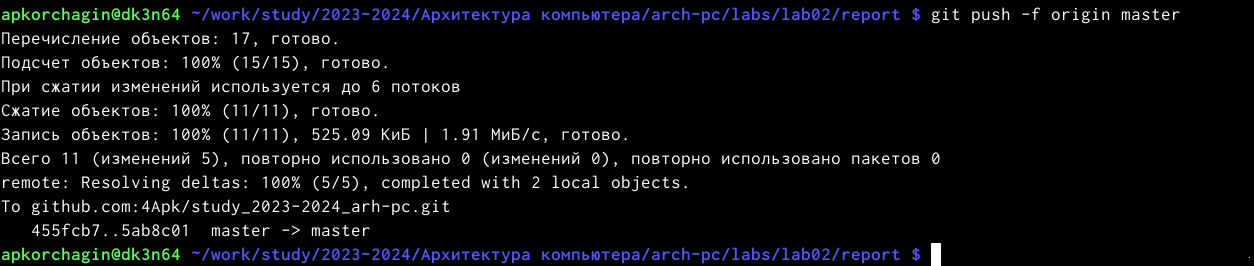
Сохранил изминения лаб 1

То же самое делаю для отчета по второй лабораторной работе: перехожу в директорию labs/lab02/report с помощью cd, добавляю с помощью git add нужный файл, сохраняю изменения с помощью git commit (рис. ??).



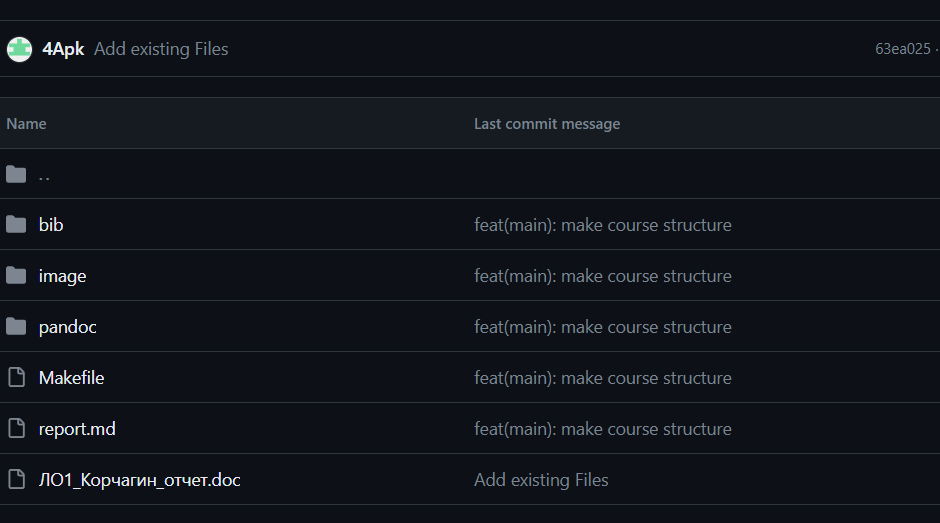
Добавил отчёт по второй лабороторной

Отправляю в центральный репозиторий сохраненные изменения командой git push -f origin master (рис. ??).



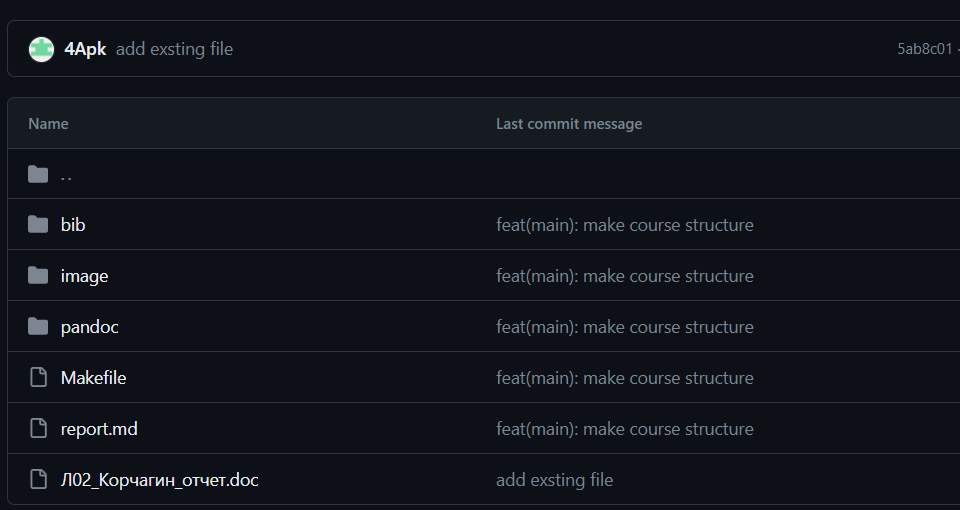
Отпрака в центральный репозиторий

При просмотре изменений на Github вижу, что были добавлены файлы с отчетами по лабораторным работам (рис. ??).



Каталог lab01/report

(рис. ??).



Каталог lab012/report

# 5 Выводы

При выполнении данной лабораторной работы я изучил идеи и при- менение средств контроля версий, а также приобрела практические навыки по работе с системой git.

# Список литературы