РУДН. Архитектура компьютеров

Отчёт по лабораторной работе №2

Косинов Никита Андреевич, НПМбв-02-20

Содержание

# 1 Цель работы

При работе большой команды людей над некоторым проектом неизменно возникают ситуации с накопленными одновременно изменениями. Для решения этой проблемы разработаны различные систимы контроля версий проектов, чтобы любой член комманды вовремя получал изменения проекта и мог их учитывать.

Цель данной работы - приобретение теоретических и практических навыков по работе с системой контроля версий на примере **Git**.

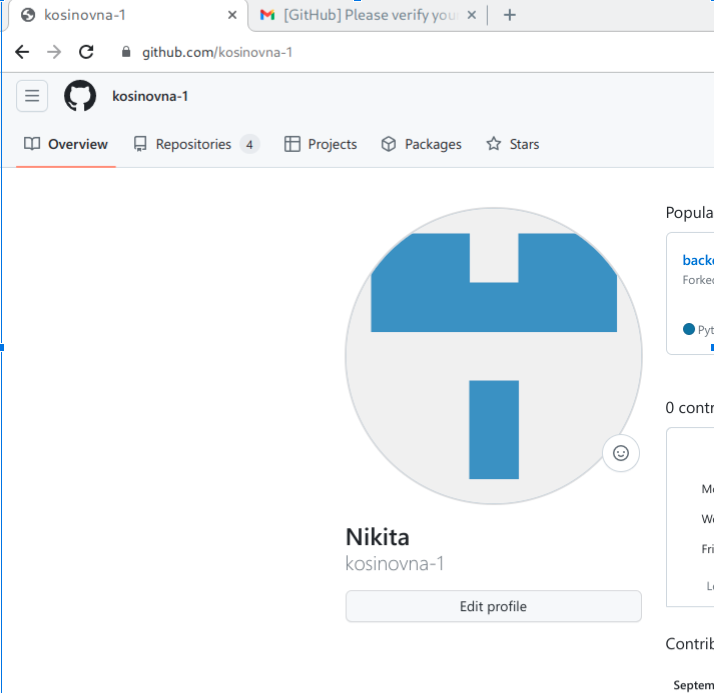
# 2 Ход работы

Лабораторная работа выполнена в терминале **OC Linux** и хостинге хранения проектов **Github** с использованием **VCS Git**. Действия по лабораторной работе представлены в следующем порядке: 1. Настройка **git** и учётной записи **github**; 2. Создание рабочего пространства; 3. Самостоятельная работа.

# 3 Настройка **github** и **git**

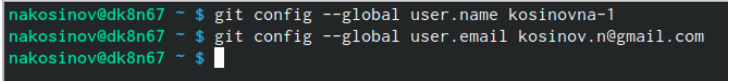
Настраиваем рабочий репозиторий и предварительную конфигурацию СКВ.

1. Создаём учётную запись на сайте **github**.



Учётная запись

1. Познакомим локальный компьютер с глобальным репозиторием, указав **e-mail** и имя с помощью команды **git**. Эта настройка занимает около двух минут.



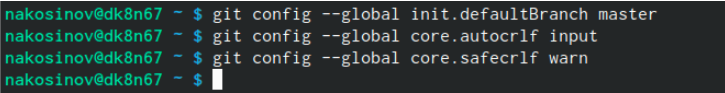
Настройка связи

1. Задаём кодировку для вывода сообщений **git**.

Указание кодировки

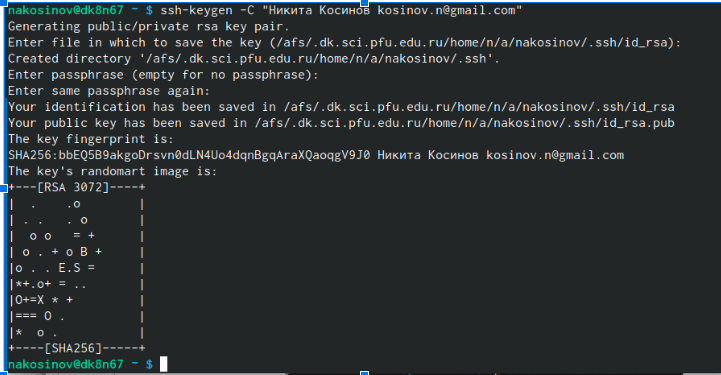
Указание кодировки

1. Задаём имя ветки - **master**, а также параметры для простоты работы с будущим проектом.



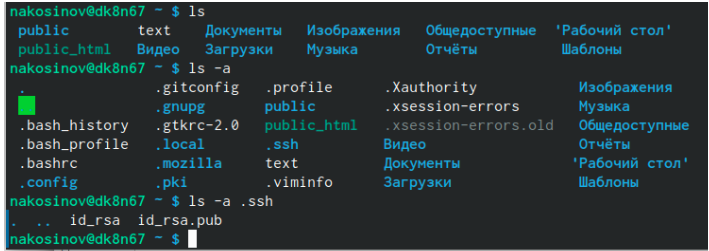
Настройка параметров

1. Создаём **SSH** ключ для идентификации. Подтверждаем место сохранения в подпапке **.ssh** домашнего каталога. Парольную фразу оставляем пустой.



Идентифицирующий ключ

1. Проверяем наличие созданных файлов. Заметим, что они скрытые, а значит, их можно увидеть с помощью ключа **-a**.

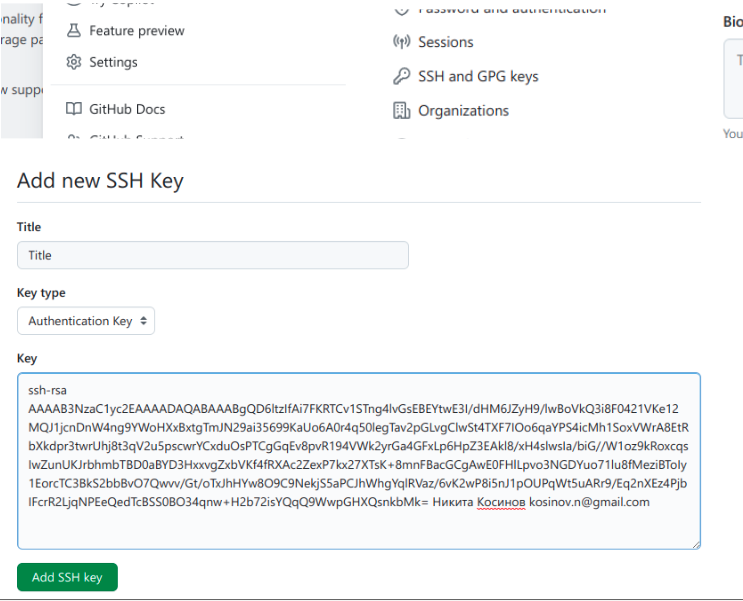


Файлы

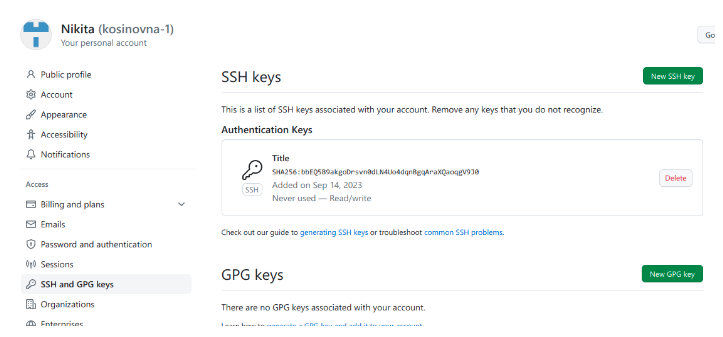
1. Загружаем публичный ключ в основной репозиторий. Для этого копируем его в буфер обмена и далее указываем в настройках на **github**: *Settings -> SSH and GPG keys -> New SSH key*.

Загрузка ключа

Загрузка ключа



Запись ключа

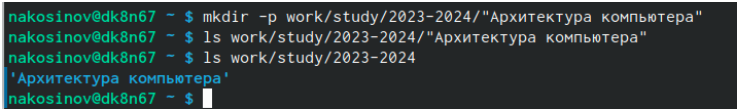


Учётная запись с ключом

# 4 Создание структурированного пространства для работы

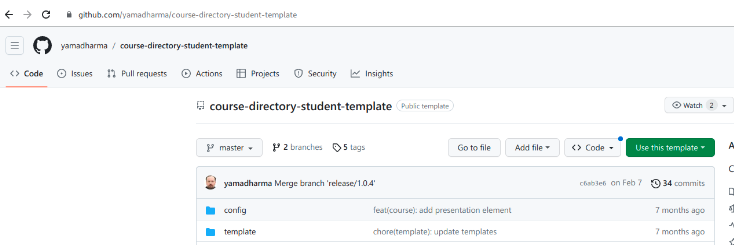
С помощью терминала и интерфейса **github** создаём удобную для чтения и работы файловую систему будущих лабораторных работ.

1. Создаём папку для предмета “Архитектура компьютера” командой **mkdir** с использованием ключа **-p** для одновременного создания вложенных папок.



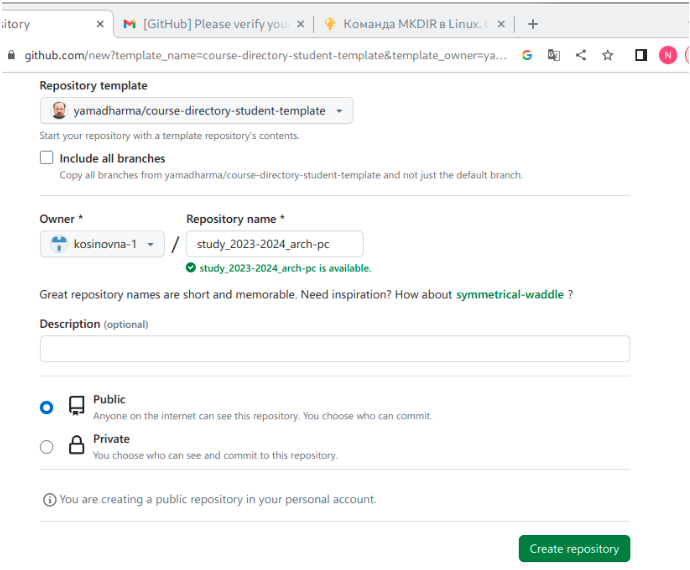
Создание папки

1. Переходим в репозиторий для студентов пользователя **yamadharma**.

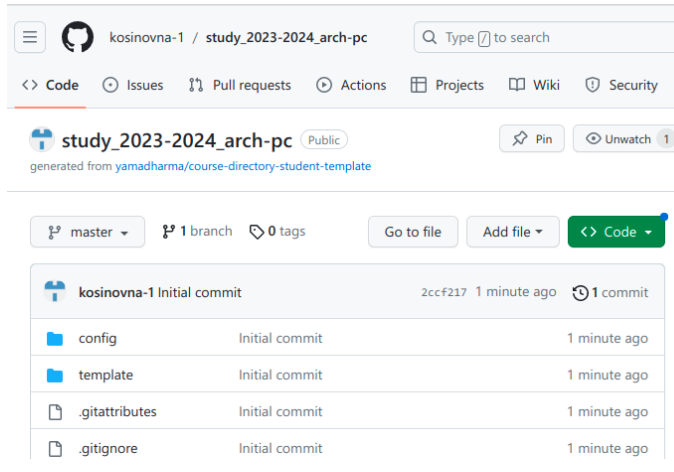


Репозиторий yamadharma

1. Нажимаем *Use this template -> Create a new repository* и создаём свой репозиторий. Оставляем его публичным.

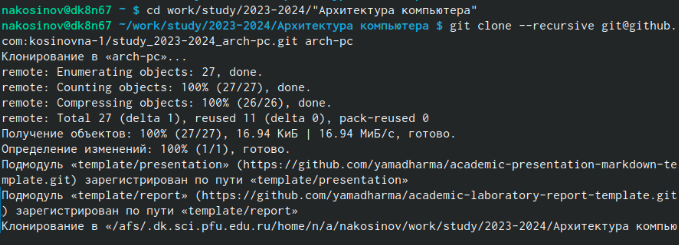


Создание репозитория

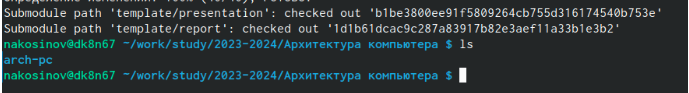


Новый репозиторий

1. Клонируем созданный репозиторий на локальный компьютер командой **git clone**.

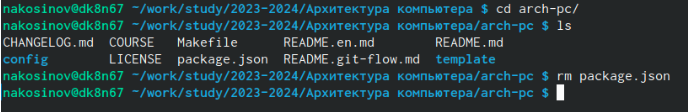


Клонирование



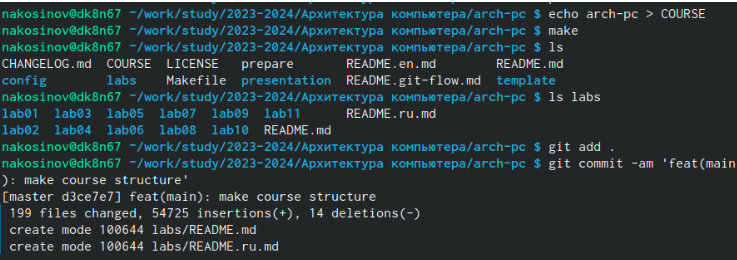
Клонирование

1. Настраиваем рабочий каталог. Переходим в папку **arch-pc** и удаляем лишний файл.

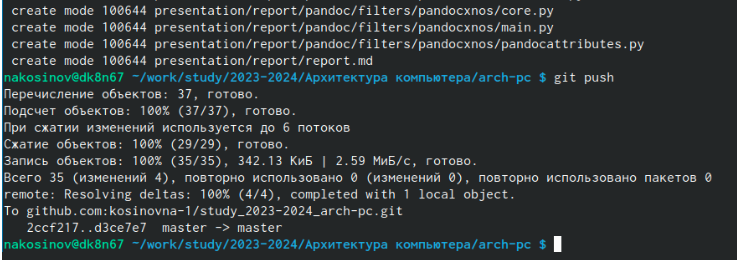


Очистка папки

1. Создаём рабочие папки, добавляем их на сервер и добавляем комментарий о проделанной работе.

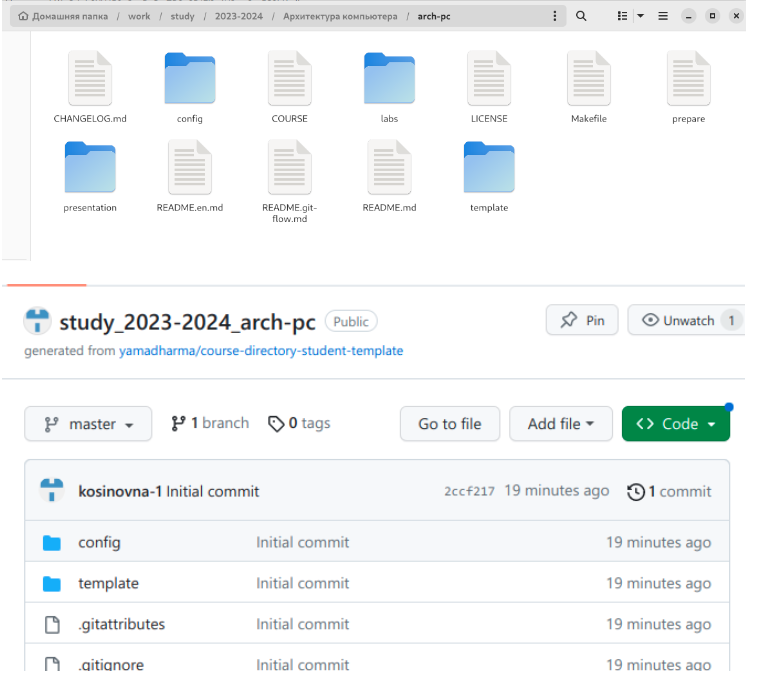


Создание каталогов

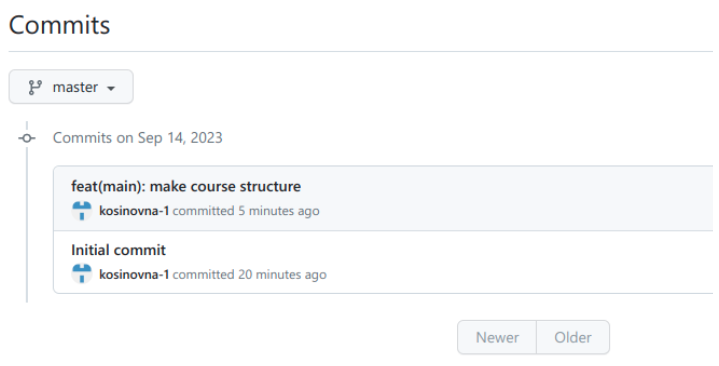


Загрузка каталогов

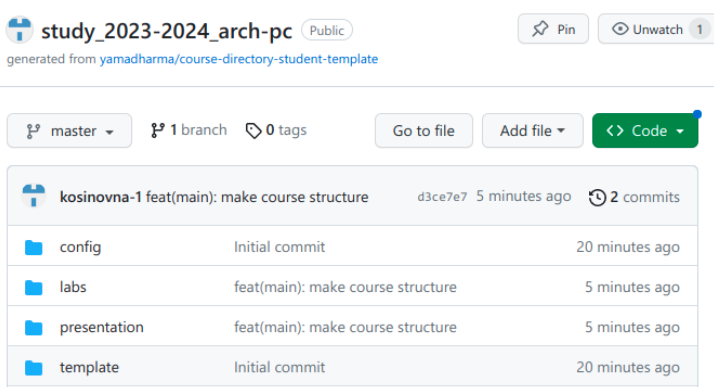
1. Проверяем, совпадает ли созданная файловая система на компьютере и на хостинге.



Каталог на локальном и глобальном репозитории совпадает



Новый коммит

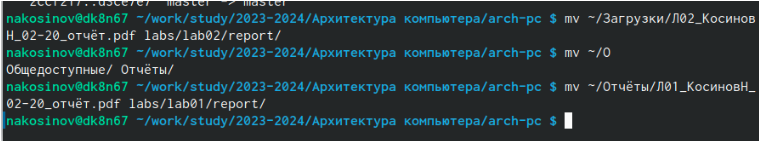


Новый коммит

# 5 Самостоятельная работа

Закрепляем полученные знания по работе с системой контроля версий.

1. Создаём отчёт о выполненной лабораторной работе №2 в подпапке **report** соответствующего каталога. Переносим отчёт о выполненной работе №1 в её **report**.



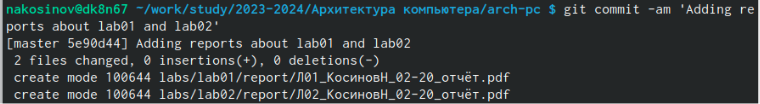
Создание отчёта

1. Подготавливаем отчёты для загрузки на **github**.

Добавление на отправку

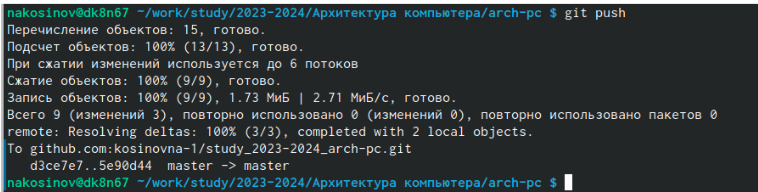
Добавление на отправку

1. Комментируем проделанные изменения.



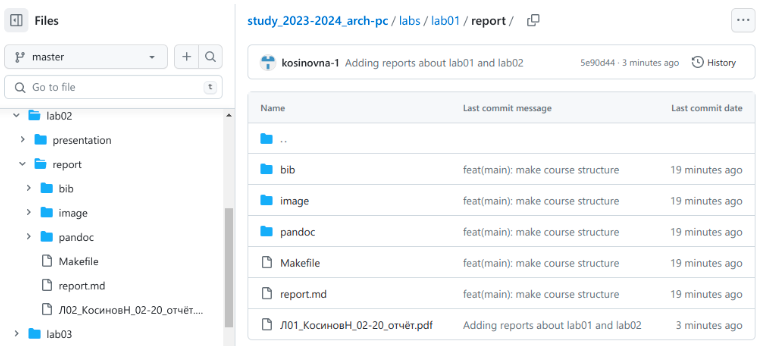
Новый коммит

1. Загружаем файлы в репозиторий.



Загрузка

1. Проверяем, что всё сработало корректно.



Репозиторий на github

1. По завершению Отчёта планируем загрузить окончательный вариант.

# 6 Выводы

В данной лабораторной работе мы узнали о способе бесконфликтного одновременного изменения проекта с использованием средств контроля версий на примере **VCS Git**. Поняли, как связывать глобальный репозиторий с локальными его копиями, с каждой из которых работает отдельный разработчик, загружать изменения и указывать на них.