РУДН. Операционные системы

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 Git.

1. У нас уже есть учётная запись на сайте **github**, так что сразу переходим к настройке репозитория и связи его с глобальным. Указываем имя пользователя и свой **e-mail**. Настраиваем также **utf-8** и верификайцию.

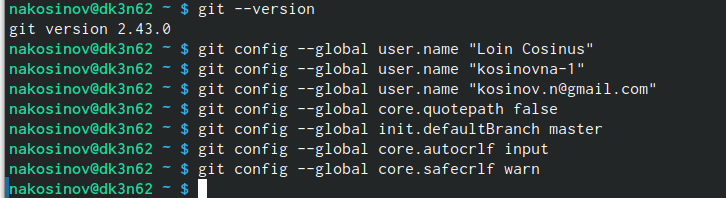


Рис. 1: Настройка git

1. Создаём пару **SSH**-ключей: публичный и приватный. Публичный - чтобы другие узлы видели нашу работу, приватный - чтобы определить узел, из которого мы будем работать.

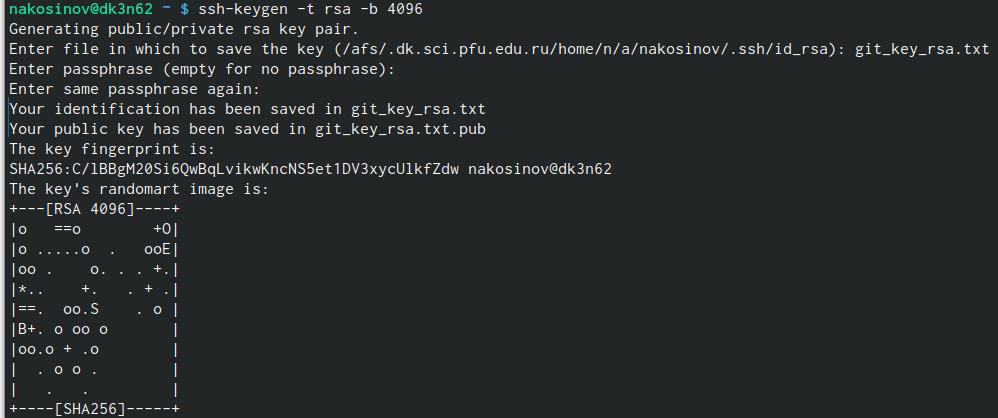


Рис. 2: Создание SSH-ключа по алгоритму RSA

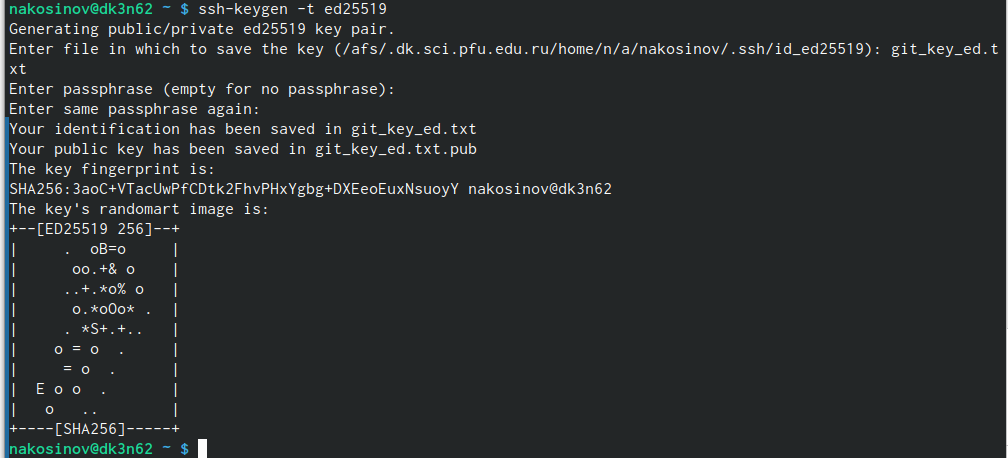


Рис. 3: Создание SSH-ключа по алгоритму ed25519

1. Привязываем созданный по алгоритму **ed25519** ключ к своему аккаунту на **GitHub**

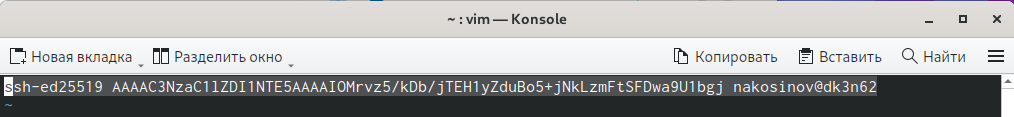


Рис. 4: Запись SSH ключа

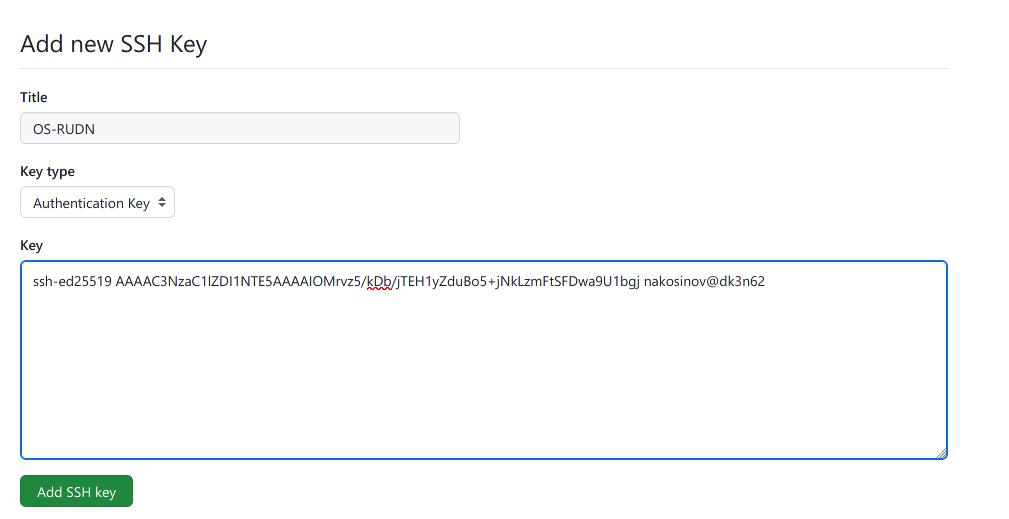


Рис. 5: Запись SSH ключа

1. Генерируем **GPG**-ключ для корректной работы коммитов. Указываем необходимую конфигурацию, своё полное имя и почту. Дополнительный пароль для защиты не ставим.

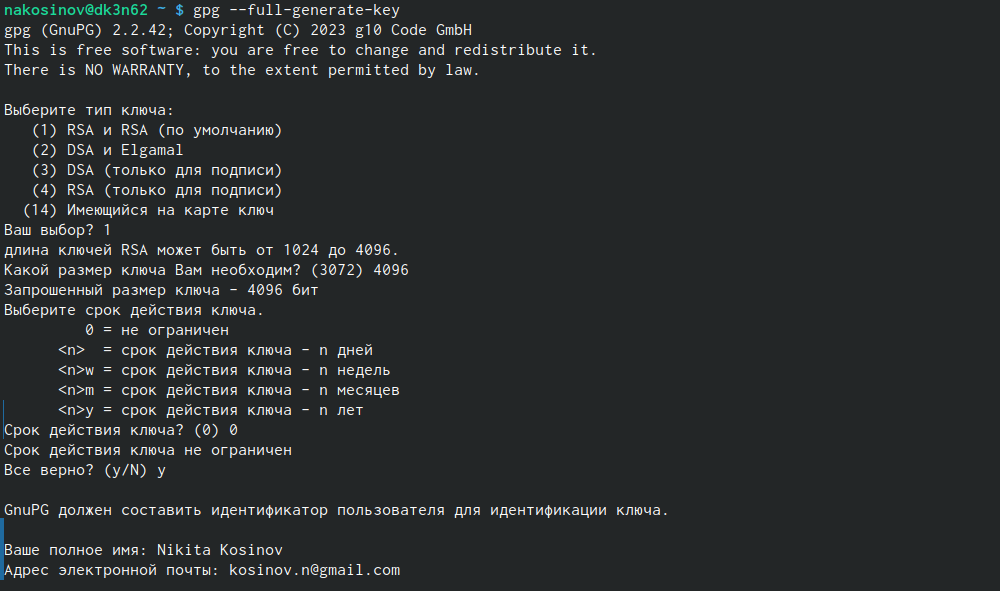


Рис. 6: Генерация PGP-ключа

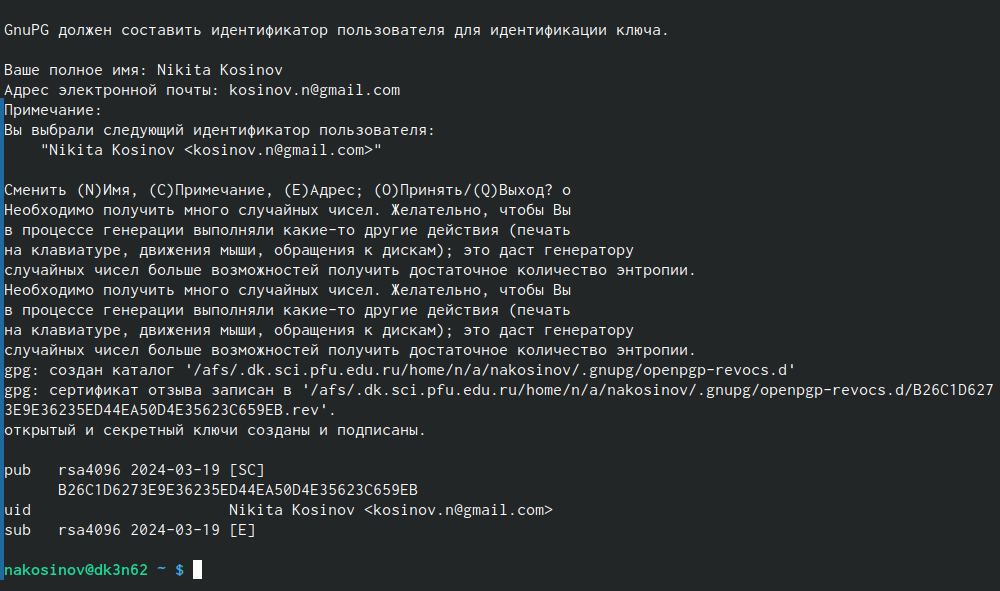


Рис. 7: Генерация PGP-ключа

1. Выводим отпечаток ключа. По нему копируем сам ключ для дальнейшей привязке к своему аккаунту на **GitHub**.

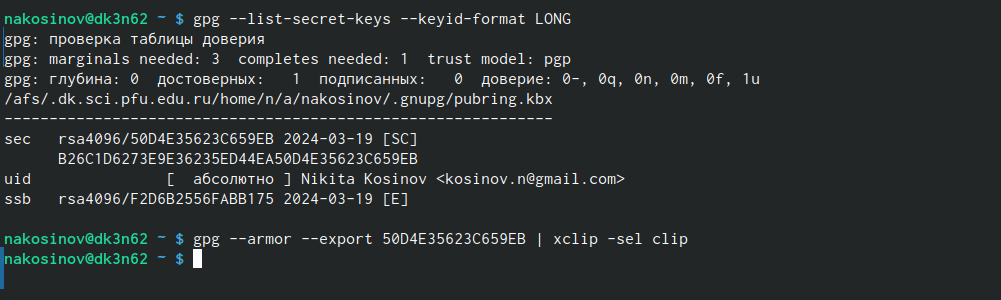


Рис. 8: Запись GPG ключа

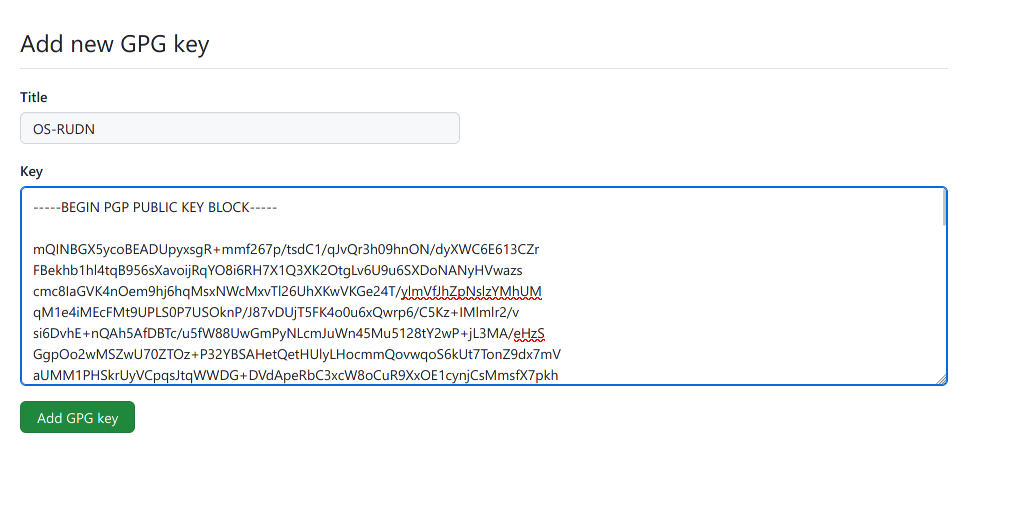


Рис. 9: Запись GPG ключа

1. Включаем режим бдительности для отслеживания неподписанных комминтов

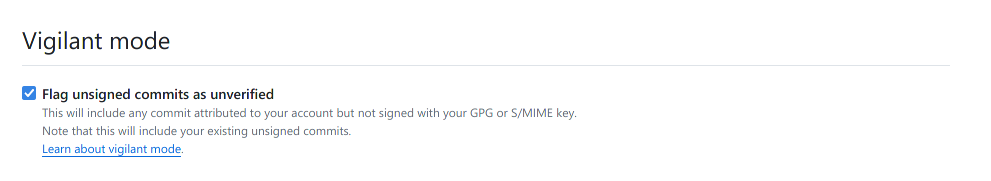


Рис. 10: Режим бдительности

1. Настраиваем автоматические подписи коммитов и настраиваем **gh**.

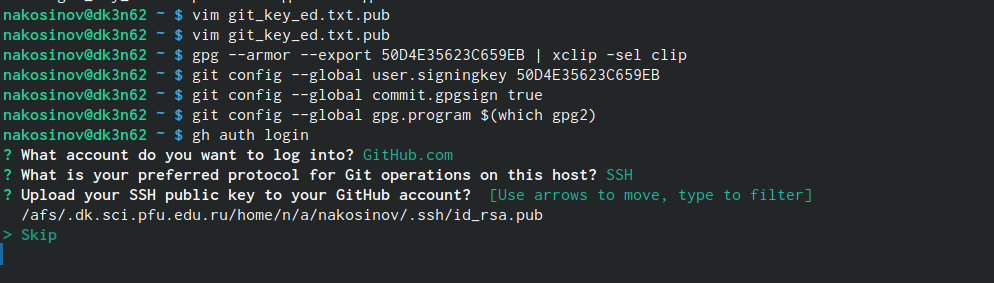


Рис. 11: Финальная настройка

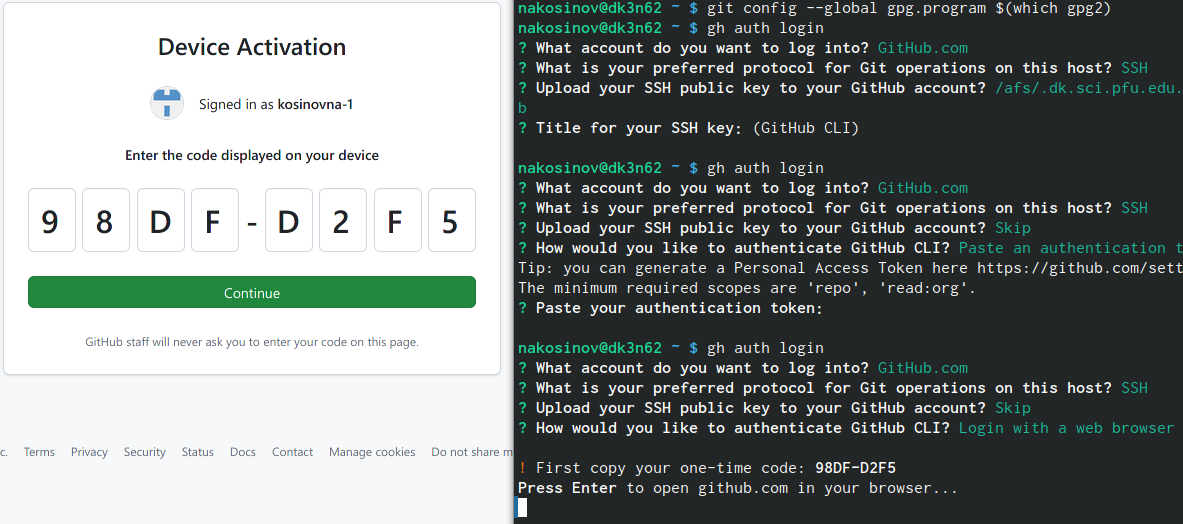


Рис. 12: Финальная настройка



Рис. 13: Финальная настройка

# 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 “Операционные системы” командой **mkdir** с использованием ключа **p** для одновременного создания вложенных папок. Переходим в созданную папку и копируем шаблон репозитория пользователя **yamadharma**.

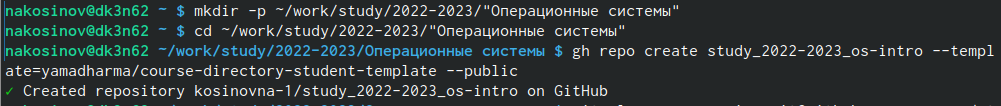


Рис. 14: Создание папки

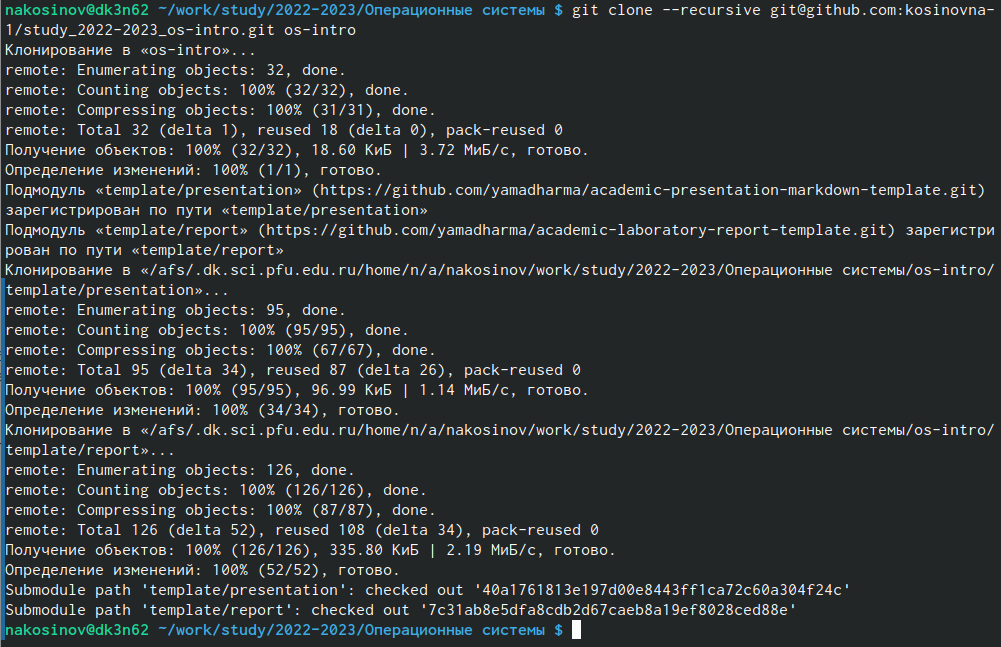


Рис. 15: Копирование шаблона

1. Проверяем, что всё заработало успешно

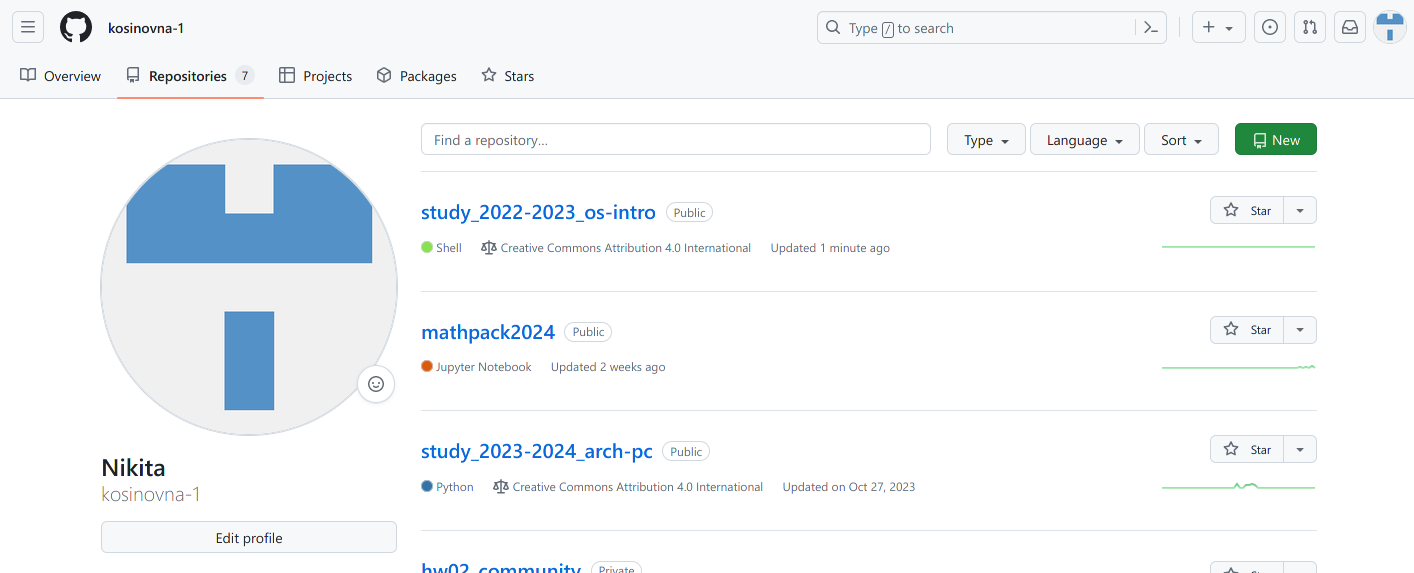


Рис. 16: Репозиторий yamadharma

1. Удаляем лишние папки из шаблона, создаём необходимые, а также заливаем изменения на репозиторий, сдела коммит **make course structure**.

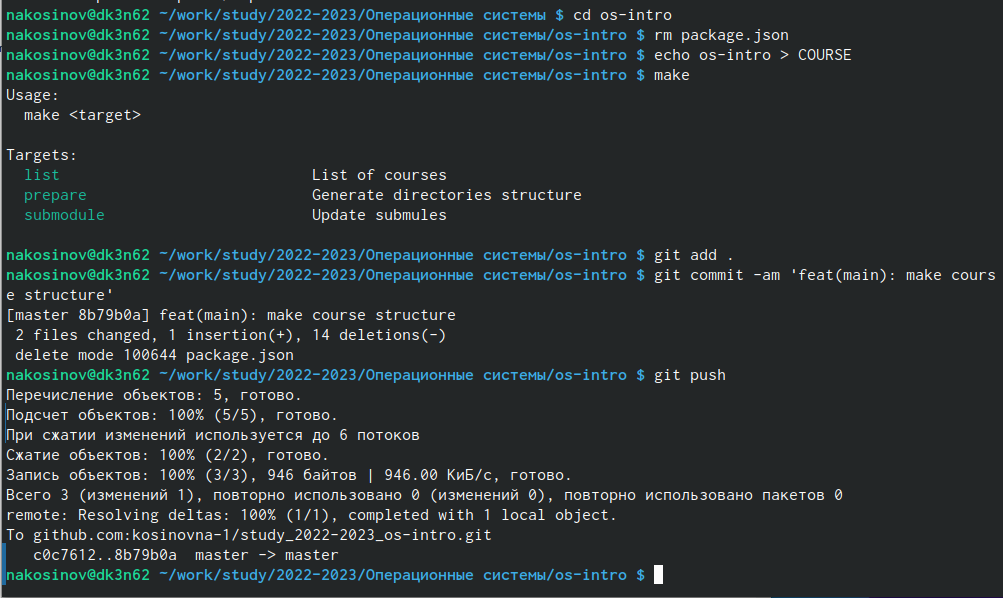


Рис. 17: Внесение изменений

1. После завершения отчёта плнируем также загрузить его на репозиторий

# 5 Контрольные вопросы

1. Система контроля версий (**Version Control System, VCS**) - это программное обеспечение для отслеживания изменений в документах, что обеспечивает удобную работу с нескольких устройств и, возможно, несколькими разработчиками. **VCS** также позволяет фиксировать версии, делать откаты на более ранние, разрешать конфликты изменений, следить, кто именно внёс и какие исправления и т.п.
2. Хранилище - место, где располагается проект. Commit - это способ сохранять изменения. История в Git - это место хранения всех сделанных commit-ов. Рабочая копия - это “снимок” одной из версий проекта, с которым можно работать на локальном компьютере.
3. Централизованная VCS - это клиент-серверское приложение, когда проект существует в единственном экземпляре и хранится в общедоступном месте - сервере. Примеры: CVS, SVN. Децентрализованная (распределённая) VCS - приложение, позволяющее хранить копию проекта у каждого разработчика, работающего с ним. При этом есть место хранения всех изменений проекта. Примеры: Mercurial, Git.

# 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.

[1].

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.