**­­­РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**

**по лабораторной работе № 2**

*дисциплина: Архитектура компьютеров и операционных систем*

Студент: Плугин Никита Сергеевич

Группа: НБИбд-02-23

**МОСКВА**

2023 г

**Содержание**

1. [Цель работы](file:///C:\Users\Huawei-PC\Downloads\Telegram%20Desktop\Л02_Акмурадов_отчет.docx#_bookmark0) 5
2. [Задание](file:///C:\Users\Huawei-PC\Downloads\Telegram%20Desktop\Л02_Акмурадов_отчет.docx#_bookmark1) 6
3. [Теоретическое введение](file:///C:\Users\Huawei-PC\Downloads\Telegram%20Desktop\Л02_Акмурадов_отчет.docx#_bookmark2) 7
4. [Выполнение лабораторной работы](file:///C:\Users\Huawei-PC\Downloads\Telegram%20Desktop\Л02_Акмурадов_отчет.docx#_bookmark3) 9

[4.1 Настройка GitHub](file:///C:\Users\Huawei-PC\Downloads\Telegram%20Desktop\Л02_Акмурадов_отчет.docx#_bookmark4) . . . . . . . . . . . . . . . . . . . . . . . . . . 9

* 1. [Базовая настройка Git](file:///C:\Users\Huawei-PC\Downloads\Telegram%20Desktop\Л02_Акмурадов_отчет.docx#_bookmark7) 10
  2. [Создание SSH-ключа](file:///C:\Users\Huawei-PC\Downloads\Telegram%20Desktop\Л02_Акмурадов_отчет.docx#_bookmark13) 11
  3. [Создание рабочего пространства и репозитория курса на основе](file:///C:\Users\Huawei-PC\Downloads\Telegram%20Desktop\Л02_Акмурадов_отчет.docx#_bookmark19) [шаблона](file:///C:\Users\Huawei-PC\Downloads\Telegram%20Desktop\Л02_Акмурадов_отчет.docx#_bookmark19) 14
  4. [Создание репозитория курса на основе шаблона](file:///C:\Users\Huawei-PC\Downloads\Telegram%20Desktop\Л02_Акмурадов_отчет.docx#_bookmark21) 15
  5. [Настройка каталога курса](file:///C:\Users\Huawei-PC\Downloads\Telegram%20Desktop\Л02_Акмурадов_отчет.docx#_bookmark28) 17
  6. [Выполнение заданий для самостоятельной работы](file:///C:\Users\Huawei-PC\Downloads\Telegram%20Desktop\Л02_Акмурадов_отчет.docx#_bookmark35) 20

1. [Выводы](file:///C:\Users\Huawei-PC\Downloads\Telegram%20Desktop\Л02_Акмурадов_отчет.docx#_bookmark54) 27
2. [Список литературы](file:///C:\Users\Huawei-PC\Downloads\Telegram%20Desktop\Л02_Акмурадов_отчет.docx#_bookmark55) 28

# Цель работы

Целью данной работы является изучить идеологию и применение средств контроля версий, а также приобрести практические навыки по работе с системой git.

# Задание

* 1. Настройка GitHub.
  2. Базовая настройка Git.
  3. Создание SSH-ключа.
  4. Создание рабочего пространства и репозитория курса на основе шаблона.
  5. Создание репозитория курса на основе шаблона.
  6. Настройка каталога курса.
  7. Выполнение заданий для самостоятельной работ

# 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 еди- ного 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 изменения, сделанные разными участниками, 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. Системы 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. 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. Система контроля версий Git представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды git с различными опциями. Благодаря тому, что Git является распределённой системой контроля версий, резервную копию локального хранилища можно сделать простым копированием или архивацией. Работа пользователя со своей веткой начинается с проверки и получения изменений из центрального репозитория (при этом в локальное дерево до начала этой процедуры не должно было вноситься изменений). Затем можно вносить изменения в локальном дереве и/или ветке. После завершения внесения какого-то изменения в файлы и/или каталоги проекта необходимо разместить их в центральном репозитории

# Выполнение лабораторной работы

## Настройка GitHub

Создаю учетную запись на сайте GitHub, далее я заполняю основные данные учетной записи и регистрирую аккаунт.

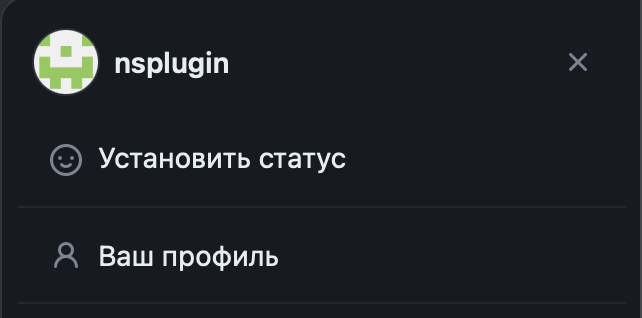


Рис. 4.2: Аккаунт GitHub

## Базовая настройка Git

Запускаю виртуальную машину, затем в терминале задаю предварительную 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

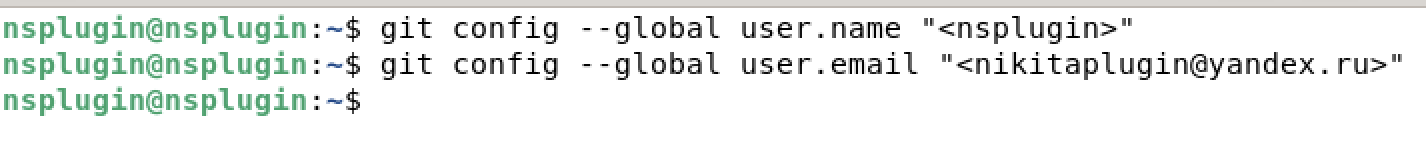


Рис. 4.3: Предварительная конфигурация git

Настраиваю utf-8 в выводе сообщений git для корректного отображения символов (рис. [4.4).](file:///C:\Users\Huawei-PC\Downloads\Telegram%20Desktop\Л02_Акмурадов_отчет.docx#_bookmark9)

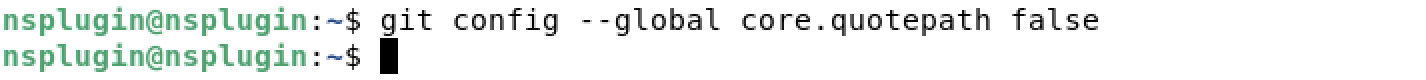


Рис. 4.4: Настройка кодировки

Задаю имя «master» для начальной ветки (рис.[4.5).](file:///C:\\Users\\Huawei-PC\\Downloads\\Telegram%20Desktop\\Л02_Акмурадов_отчет.docx" \l "_bookmark10)

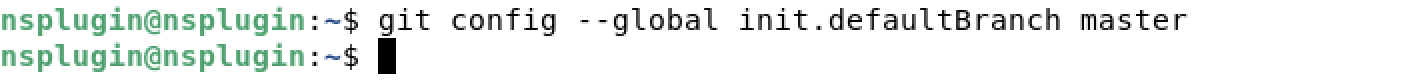


Рис. 4.5: Создание имени для начальной ветки

Задаю параметр autocrlf со значением input, так как я работаю в системе Linux, чтобы конвертировать CRLF в LF только при коммитах (рис. [4.6).](file:///C:\Users\Huawei-PC\Downloads\Telegram%20Desktop\Л02_Акмурадов_отчет.docx#_bookmark11) CR и LF – это символы, которые можно использовать для обозначения разрыва строки в текстовых файлах.

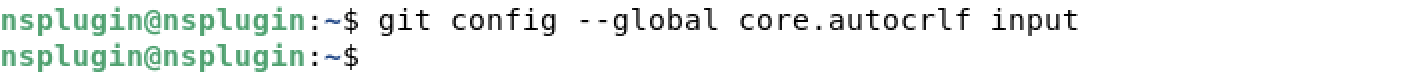


Рис. 4.6: Параметр autocrlf

Задаю параметр safecrlf со значением warn, так Git будет проверять преобразование на обратимость (рис. [4.7).](file:///C:\Users\Huawei-PC\Downloads\Telegram%20Desktop\Л02_Акмурадов_отчет.docx#_bookmark12) При значении warn Git только выведет предупреждение, но будет принимать необратимые конвертации.

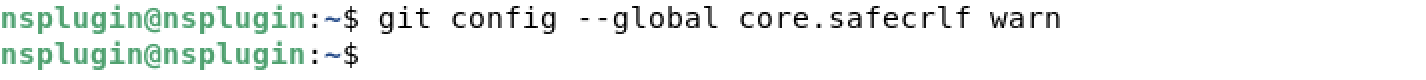


Рис. 4.7: Параметр safecrlf

## Создание SSH-ключа

Для последующей идентификации пользователя на сервере репозиториев необходимо сгенерировать пару ключей (приватный и открытый). Для этого ввожу команду ssh-keygen -C “Имя Фамилия, work@email”, указывая имя владельца и электронную почту владельца (рис. [4.8).](file:///C:\Users\Huawei-PC\Downloads\Telegram%20Desktop\Л02_Акмурадов_отчет.docx#_bookmark14) Ключ автоматически сохранится в каталоге ~/.ssh/.

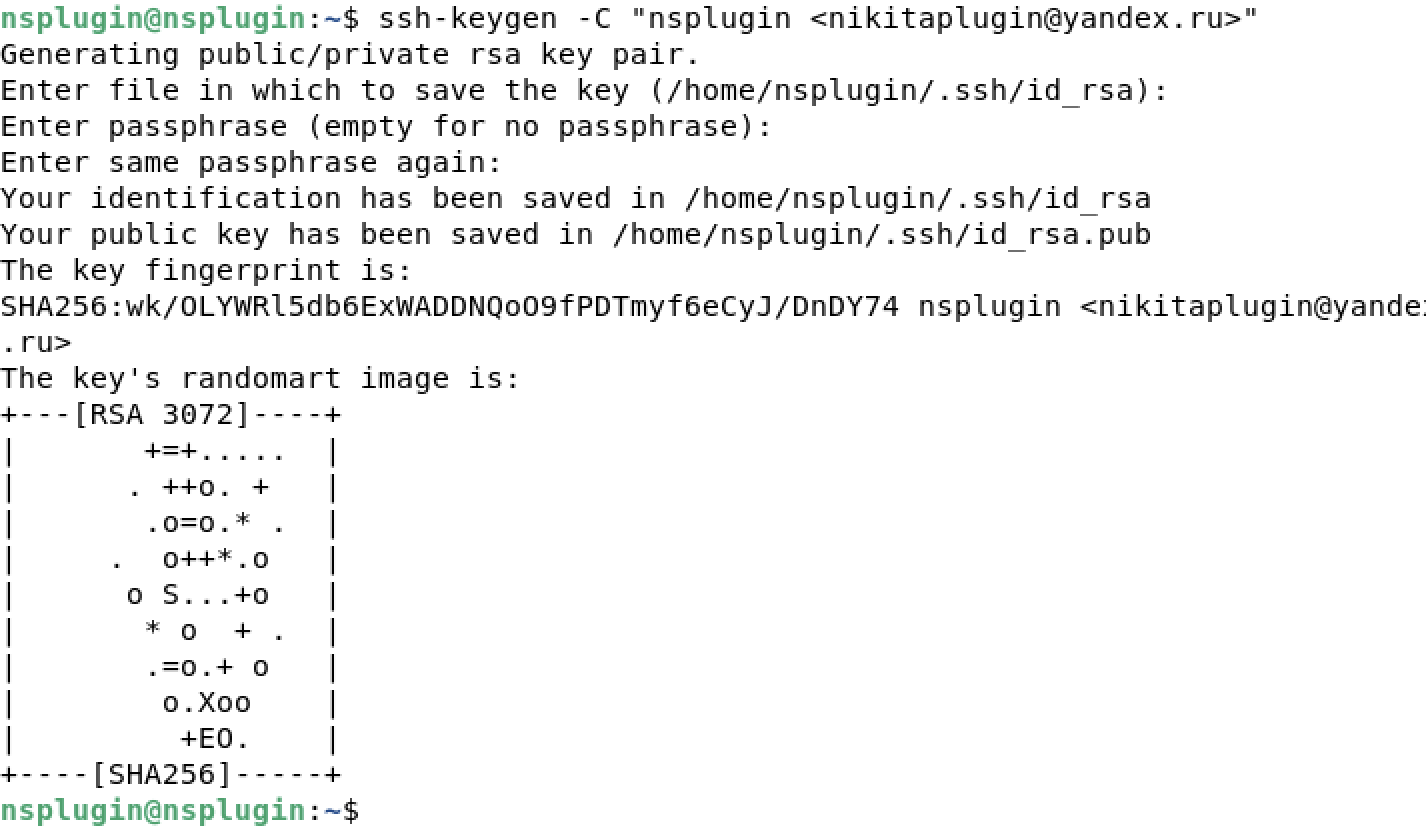


Рис. 4.8: Генерация SSH-ключа

Xclip – утилита, позволяющая скопировать любой текст через терминал. Проверяю ее наличие с помощью команды apt-get install с ключом -y от имени суперпользователя, введя в начале команды sudo (рис. [4.9).](file:///C:\Users\Huawei-PC\Downloads\Telegram%20Desktop\Л02_Акмурадов_отчет.docx#_bookmark15)

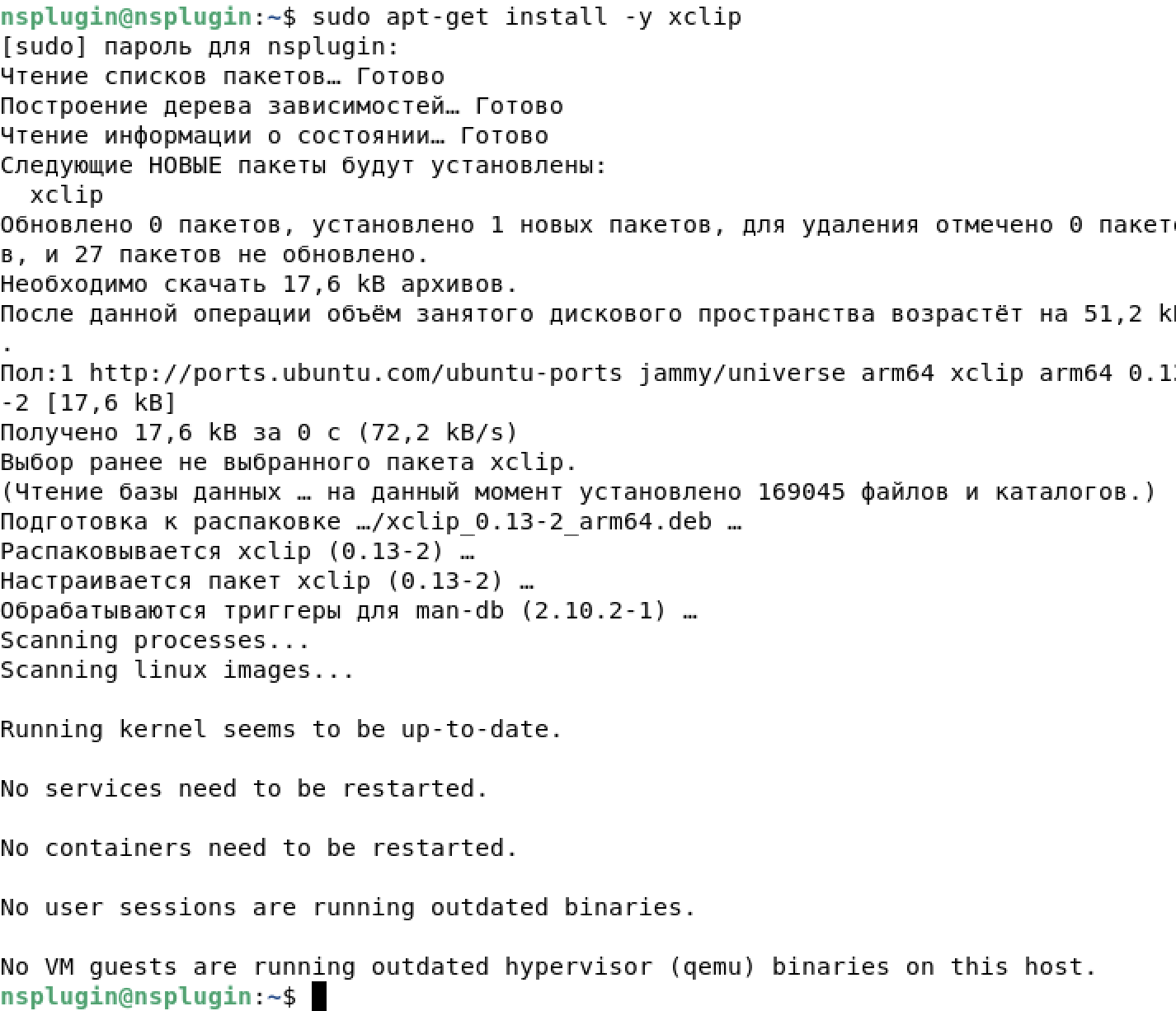


Рис. 4.9: Установка утилиты xclip

Копирую открытый ключ из директории, в которой он был сохранен, с помощью утилиты xclip (рис. [4.10).](file:///C:\Users\Huawei-PC\Downloads\Telegram%20Desktop\Л02_Акмурадов_отчет.docx#_bookmark16)

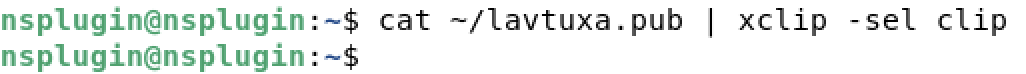


Рис. 4.10: Копирование содержимого файла

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

Вставляю скопированный ключ в поле «Key». В поле Title указываю имя для ключа. Нажимаю «Add SSH-key», чтобы завершить добавление ключа (рис. [4.11).](file:///C:\Users\Huawei-PC\Downloads\Telegram%20Desktop\Л02_Акмурадов_отчет.docx#_bookmark18)

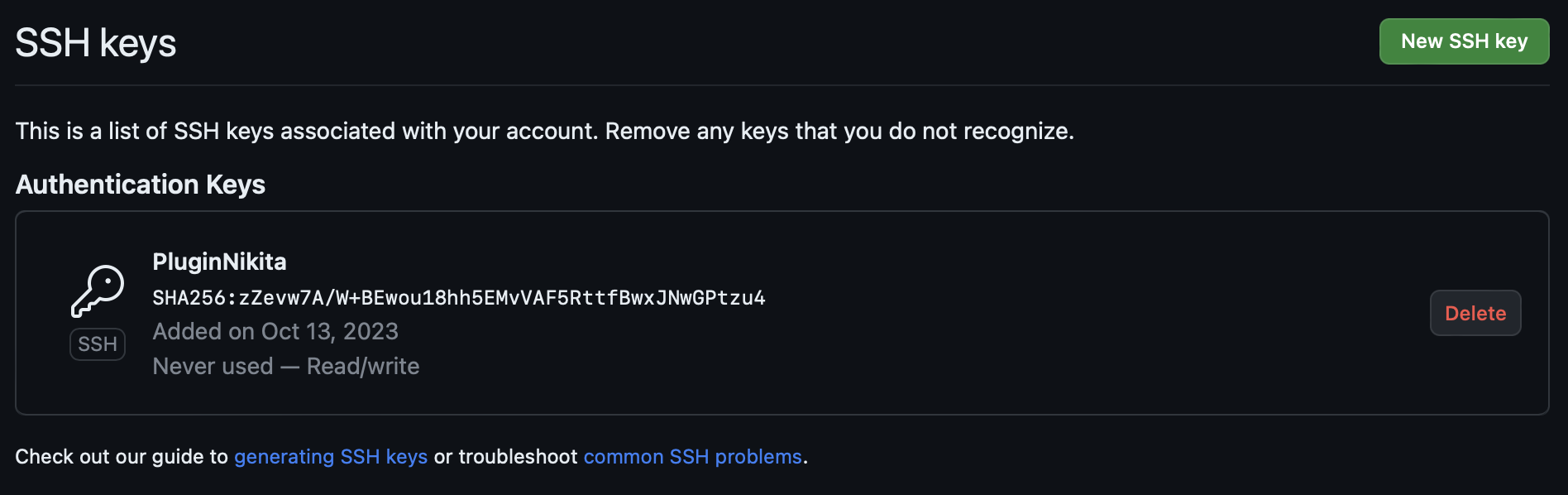


Рис. 4.11: Добавление ключа

## Создание рабочего пространства и репозитория курса на основе шаблона

Закрываю браузер, открываю терминал. Создаю директорию, рабочее пространство, с помощью утилиты mkdir, благодаря ключу -p создаю все директории после домашней ~/work/study/2023-2024/“Computer architecture” рекурсивно. Далее проверяю с помощью ls, действительно ли были созданы необходимые мне каталоги (рис. [4.12).](file:///C:\Users\Huawei-PC\Downloads\Telegram%20Desktop\Л02_Акмурадов_отчет.docx#_bookmark20)

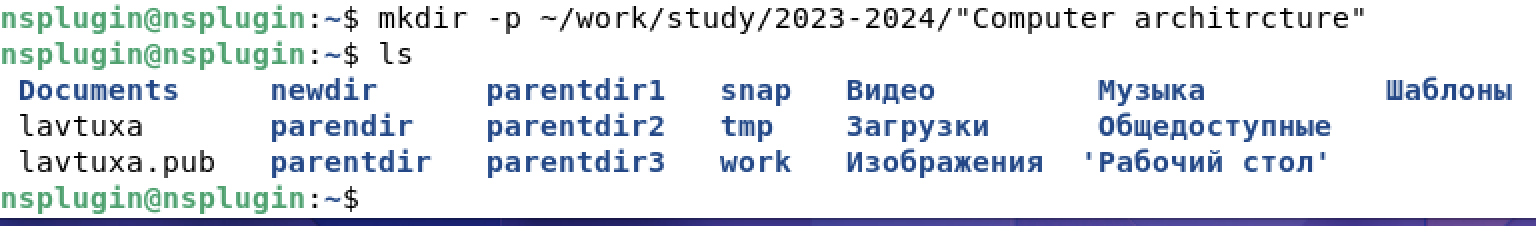


Рис. 4.12: Создание рабочего пространства

**Создание репозитория курса на основе шаблона**

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. [4.13).](file:///C:\Users\Huawei-PC\Downloads\Telegram%20Desktop\Л02_Акмурадов_отчет.docx#_bookmark22)

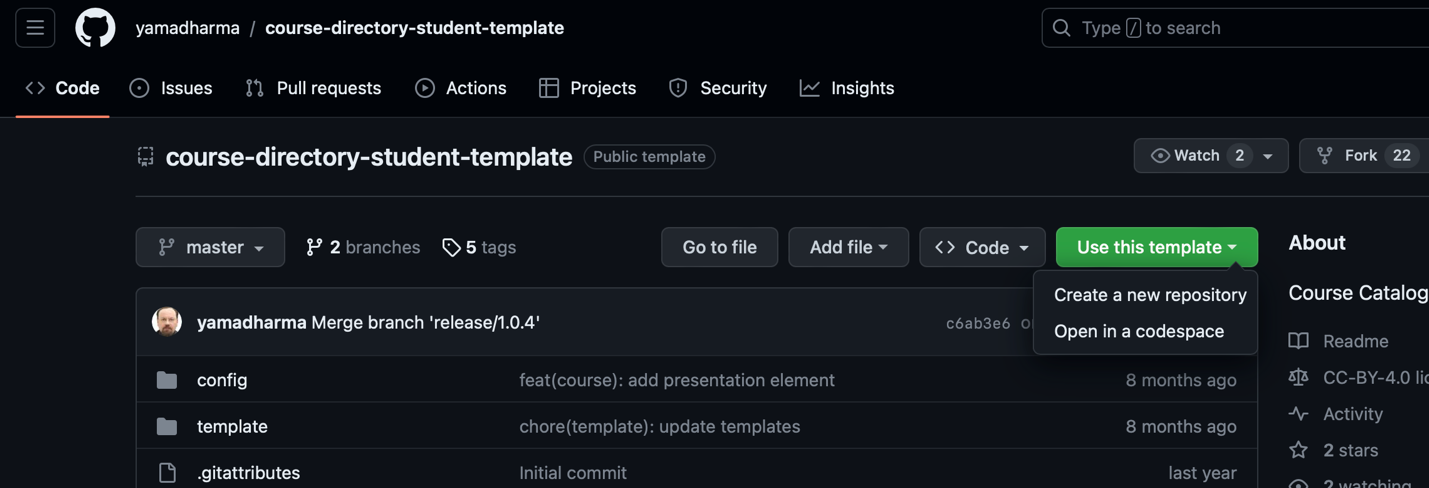


Рис. 4.13: Страница шаблона для репозитория

В открывшемся окне задаю имя репозитория (Repository name): study\_2022–2023\_arch- pc и создаю репозиторий, нажимаю на кнопку «Create repository»

(рис. [4.14).](file:///C:\Users\Huawei-PC\Downloads\Telegram%20Desktop\Л02_Акмурадов_отчет.docx#_bookmark23)

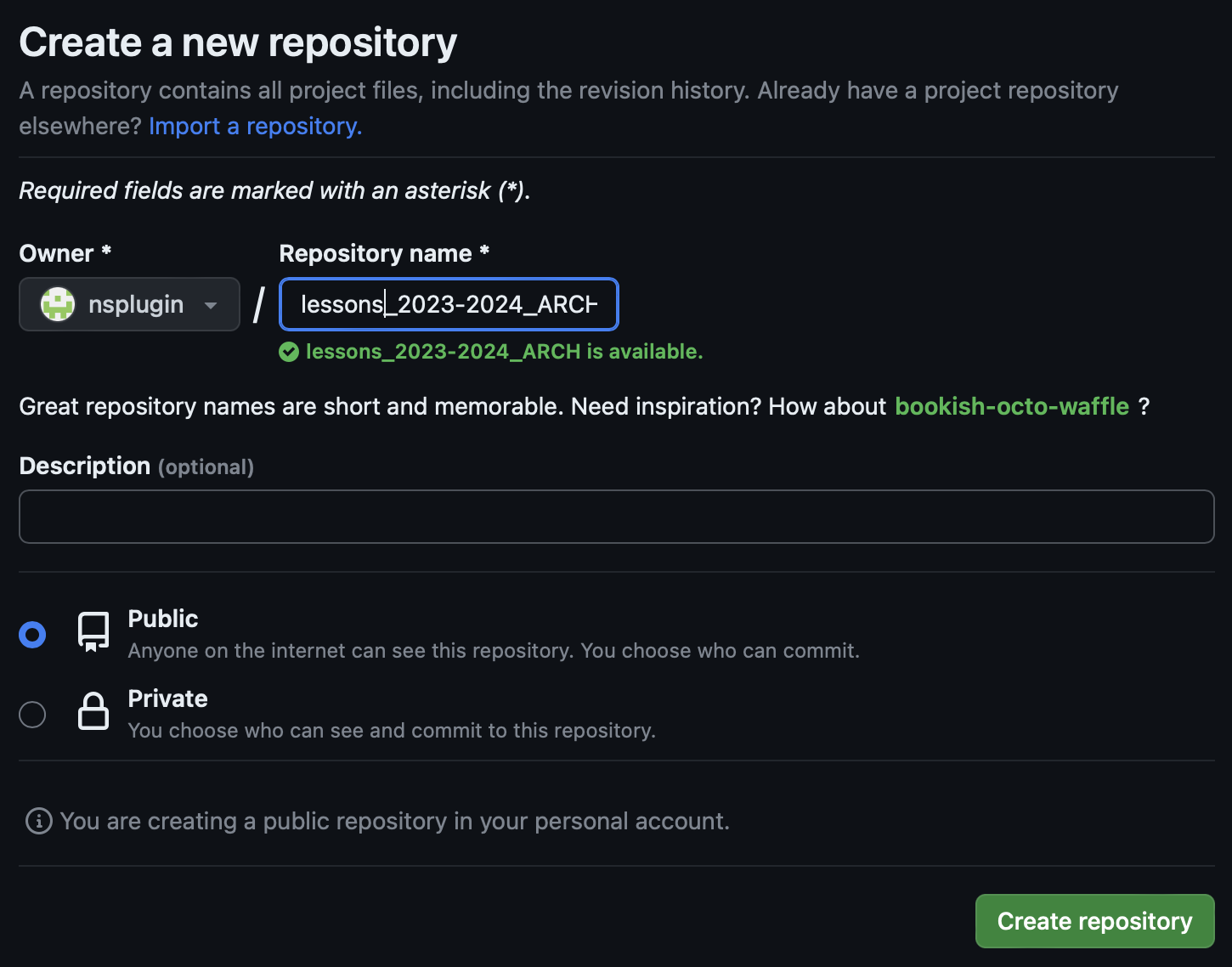


Рис. 4.14: Окно создания репозитория

Через терминал перехожу в созданный каталог курса с помощью утилиты cd (рис. [4.15).](file:///C:\Users\Huawei-PC\Downloads\Telegram%20Desktop\Л02_Акмурадов_отчет.docx#_bookmark25)

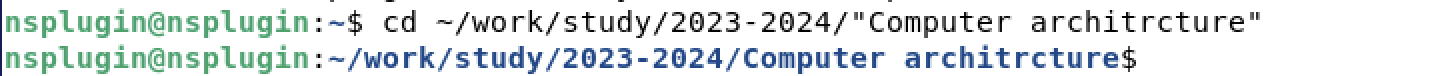


Рис. 4.15: Перемещение между директориями

Копирую ссылку для клонирования на странице созданного репозитория, сначала перейдя в окно «code», далее выбрав в окне вкладку «SSH»

Клонирую созданный репозиторий с помощью команды

git clone –recursive git@github.com:/study\_2023–2024\_arh-pc.git arch-pc (рис. [4.16).](file:///C:\Users\Huawei-PC\Downloads\Telegram%20Desktop\Л02_Акмурадов_отчет.docx#_bookmark26)

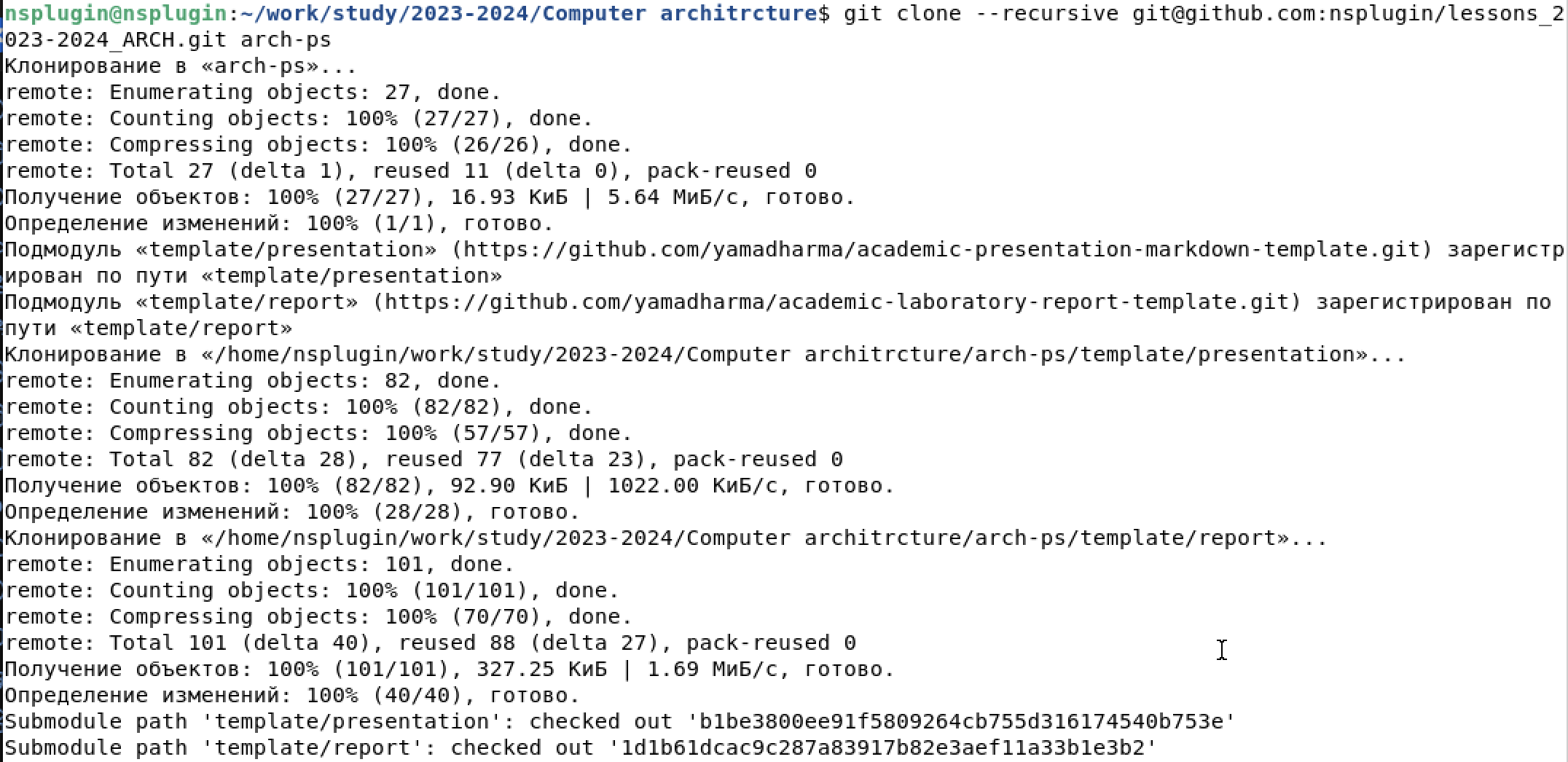


Рис. 4.16: Клонирование репозитория

## Настройка каталога курса

Удаляю лишние файлы с помощью утилиты rm (рис. [4.17).](file:///C:\Users\Huawei-PC\Downloads\Telegram%20Desktop\Л02_Акмурадов_отчет.docx#_bookmark30)



Рис. 4.17: Удаление файлов Создаю необходимые каталоги (рис. [4.18).](file:///C:\Users\Huawei-PC\Downloads\Telegram%20Desktop\Л02_Акмурадов_отчет.docx#_bookmark31)

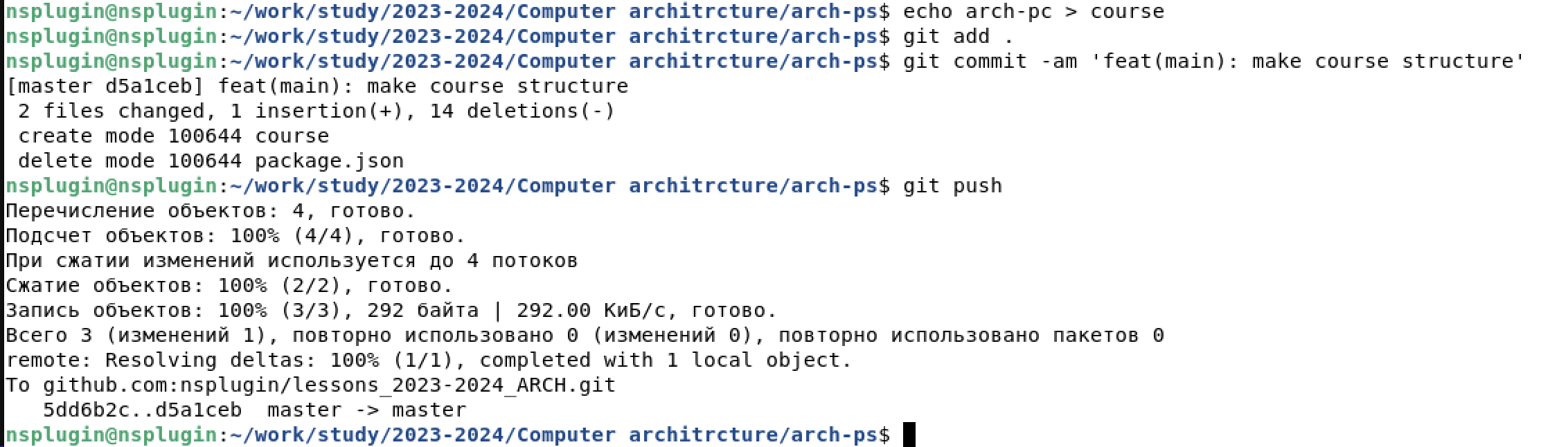


Рис. 4.18: Создание каталогов

# Самостоятельная работа:

# Создадим отчёт по выполнению работы в каталоге рабочего пространства в lab01 и lab02. Загрузим файлы на github

# Вывод:

# Была изучена идеология и применение средств контроля версий, были приобретены практические навыки по работе с системой git, а также по работе сайте https://github.com/

# Список литературы

* 1. [Архитектура ЭВМ](https://esystem.rudn.ru/pluginfile.php/1584622/mod_resource/content/1/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%963.pdf)
  2. [Git - gitattributes Документация](https://git-scm.com/docs/gitattributes)