**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**

**по лабораторной работе №**

*дисциплина: Операцинные системы*

Студент: Сунгурова Мариян Мухсиновна

Группа: НКНбд-01-21

**МОСКВА**

2022 г.

**Цель работы**

1. Изучить идеологию и применение средств контроля версий.
2. Освоить умения по работе с git.

**Результаты выполнения работы**

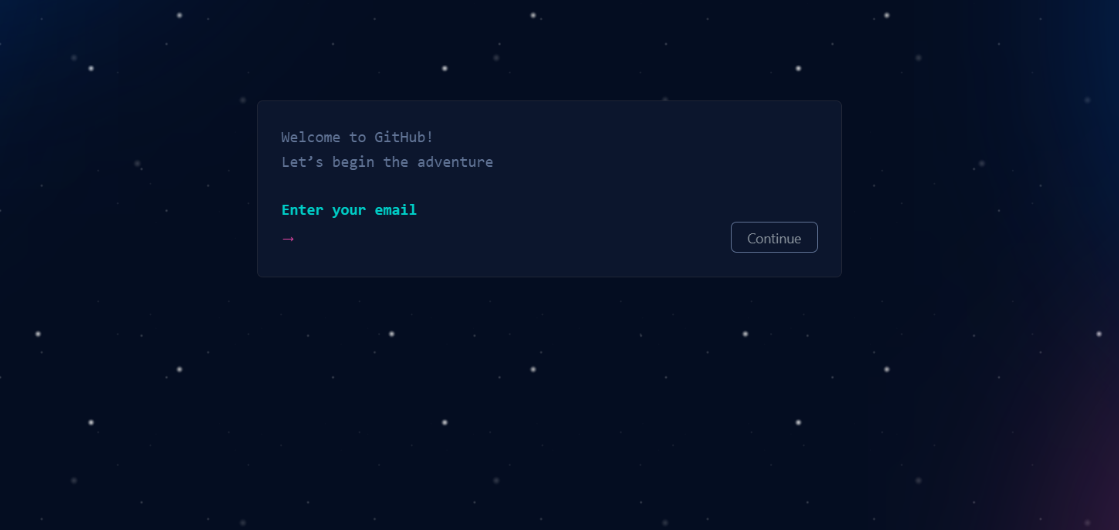


Рисунок 1. Создала учётную запись на <https://github.com>. Заполнила основные данные на <https://github.com>.

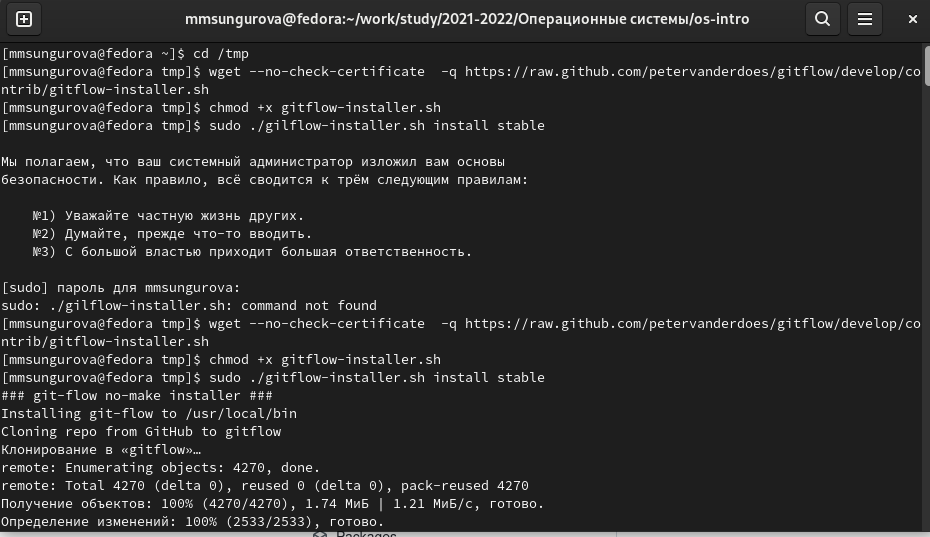


Рисунок 2. Установила git-flow в Fedora Linux

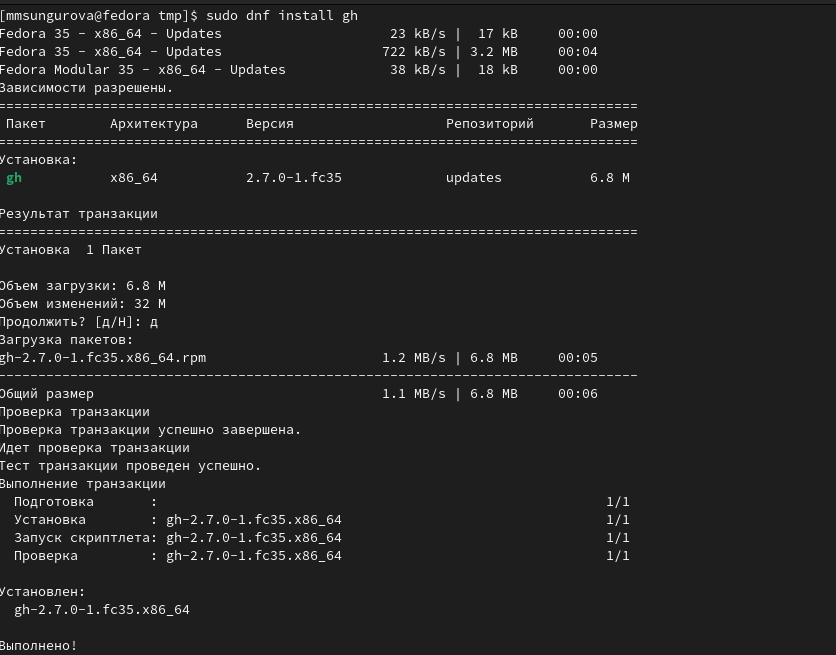


Рисунок 3. Установила gh в Fedora Linux

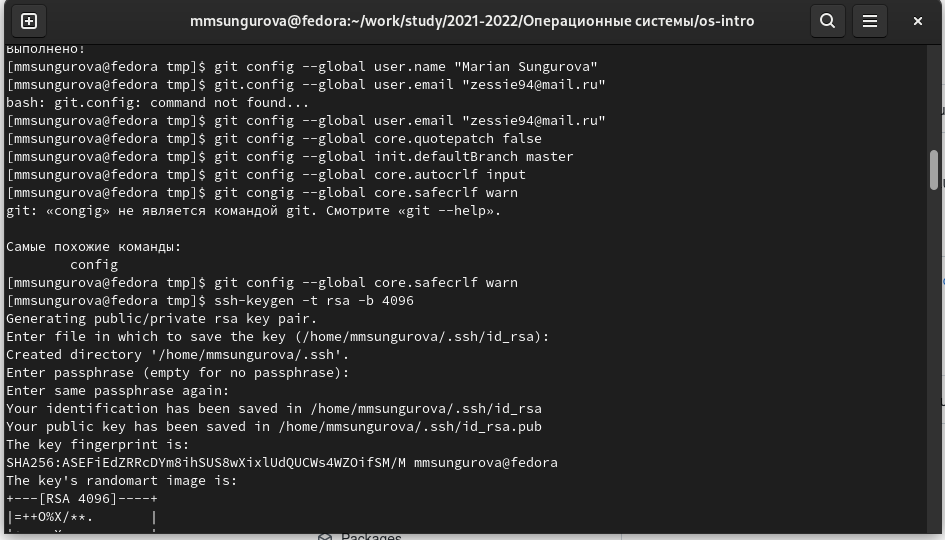


Рисунок 4. Произвела базовую настройку git

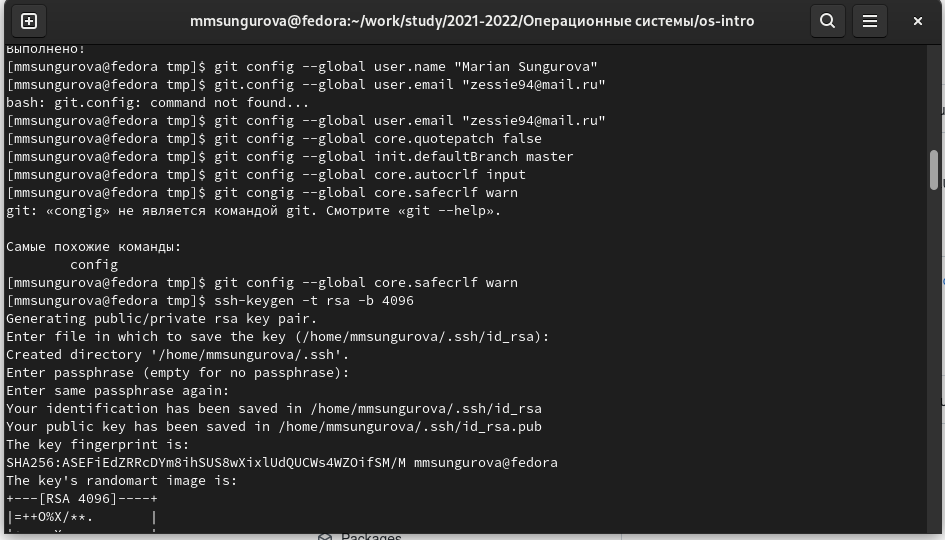


Рисунок 5. Создала ключ ssh по алгоритму rsa размером 4096 мб

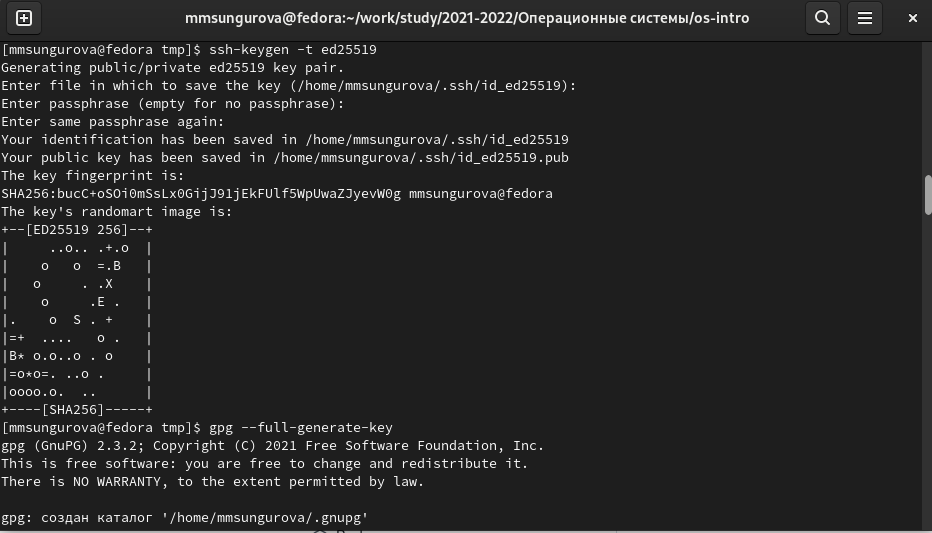


Рисунок 6. по алгоритму ed25519

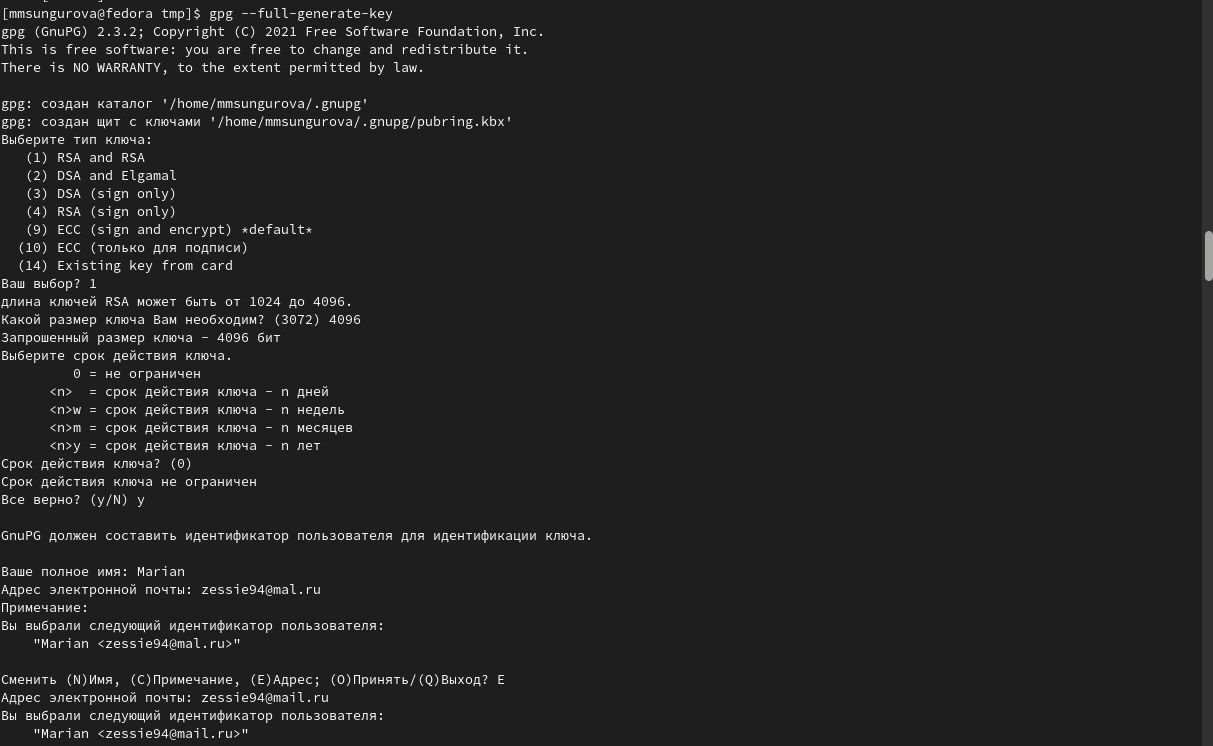


Рисунок 7. Создала ключи pgp.

Из предложенных опций выбрала:

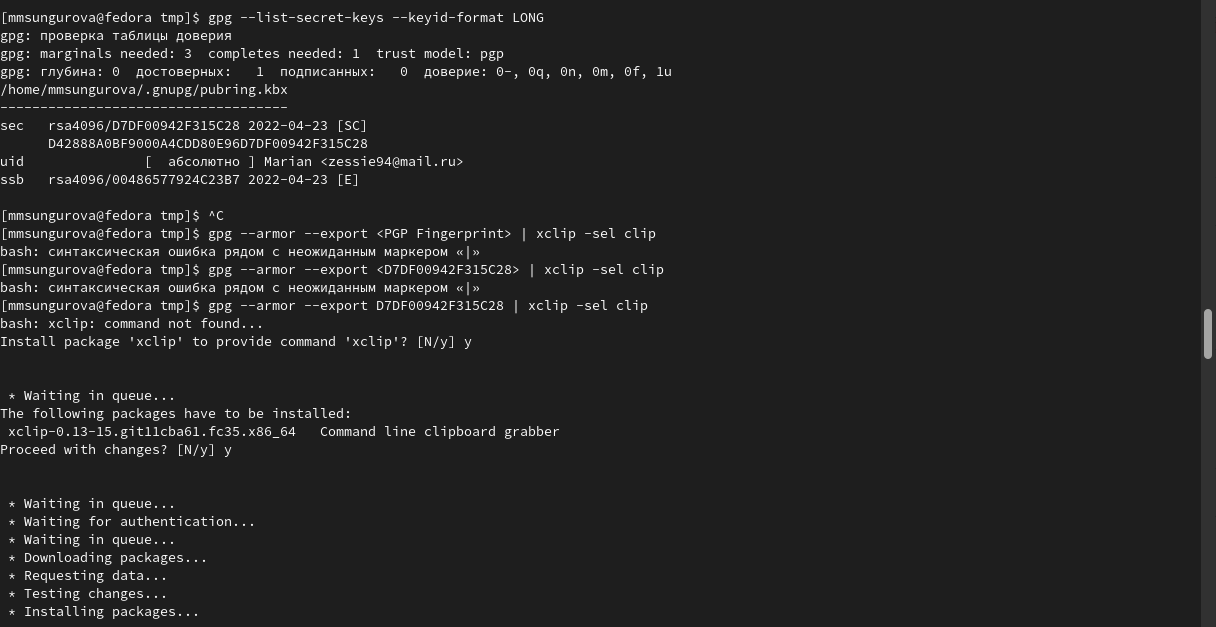
– тип RSA and RSA;

– размер 4096;

– выбрала срок действия по умолчанию— 0 (срок действия не истекает никогда).

– Имя.

– Адрес электронной почты.



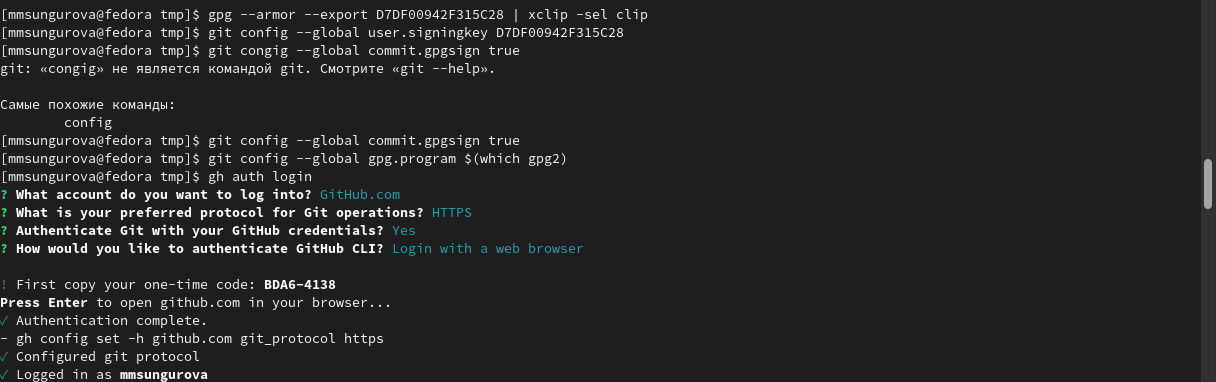


Рисунок 8. Добавила PGP ключ в GitHub

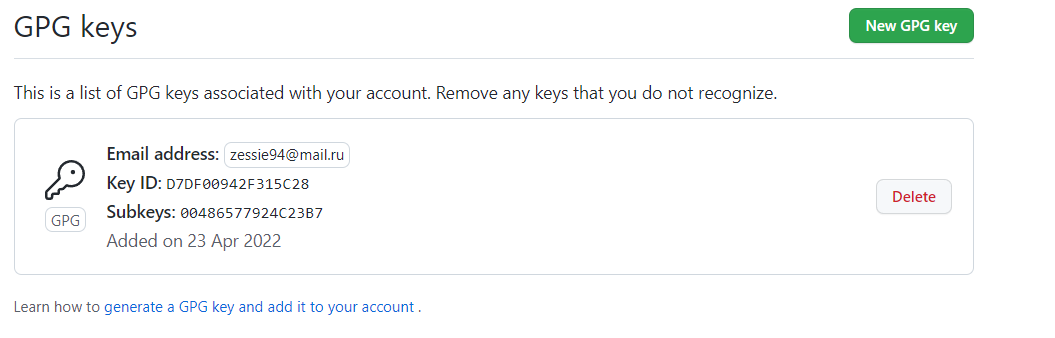


Рисунок 9.Перешла в настройки GitHub (https://github.com/settings/keys), нажала на кнопку New GPG key и вставила полученный ключ в поле ввода.

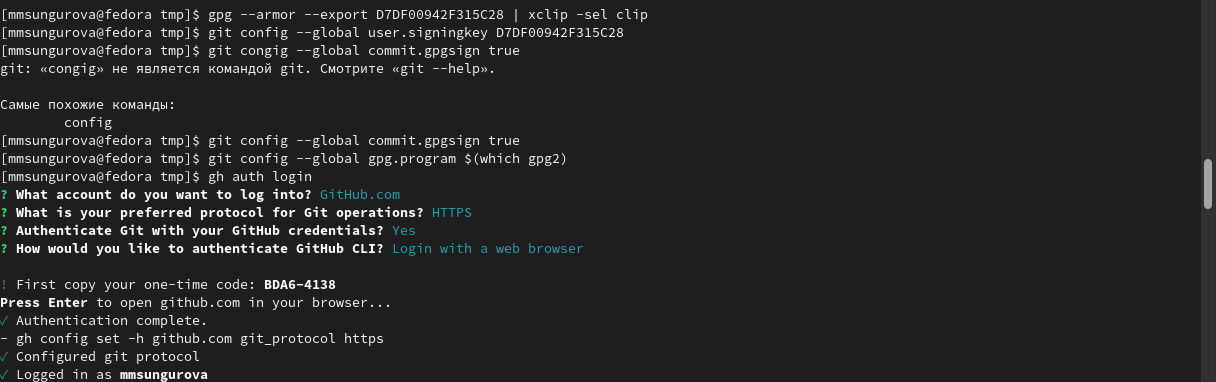


Рисунок 10. Произвела настройку автоматических подписей коммитов git

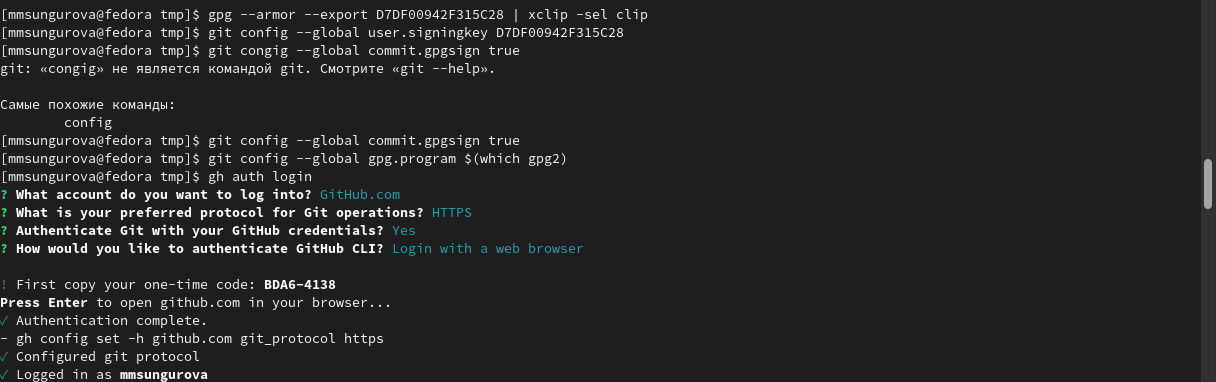
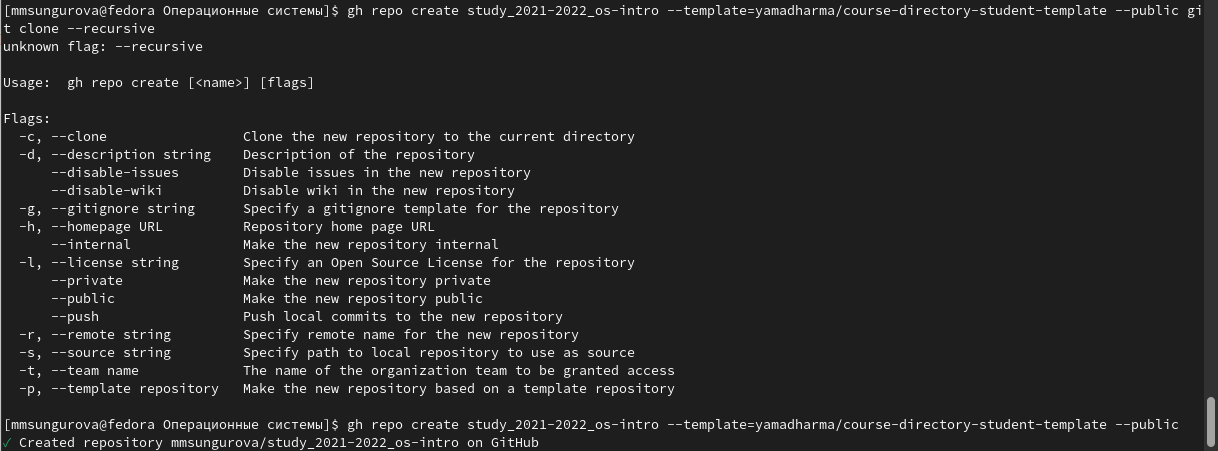
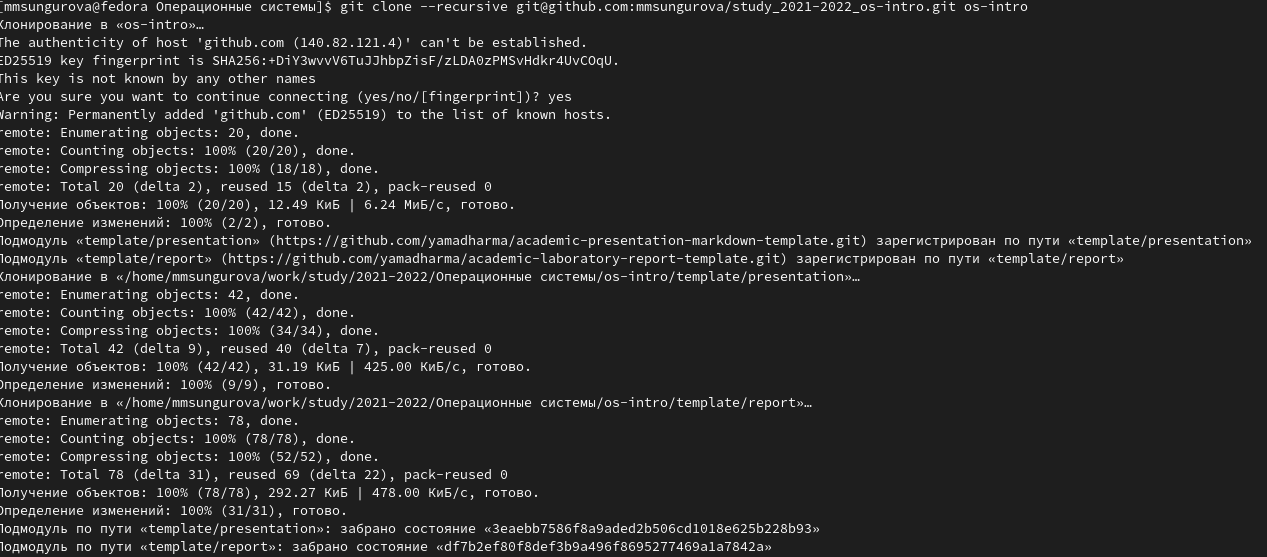
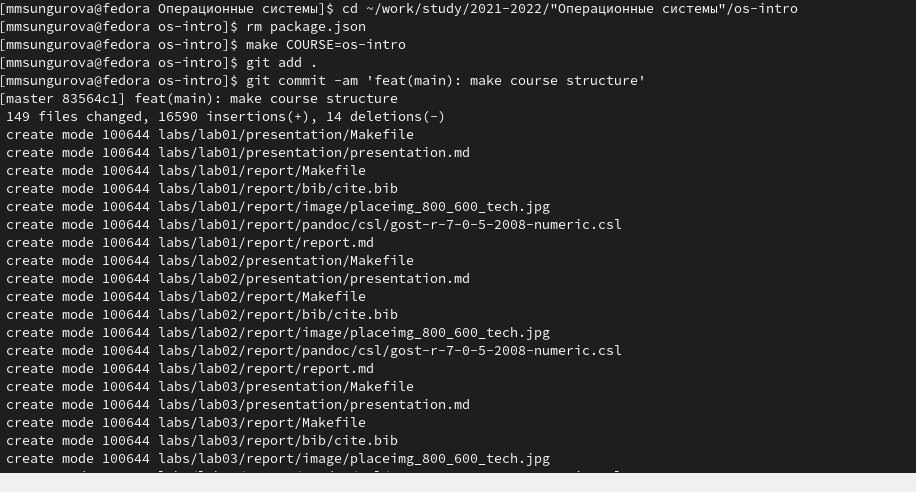


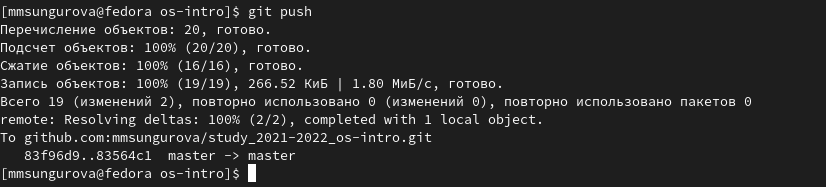
Рисунок 11. Настроила gh





Рисунки 12-13 Создала репозиторий курса на основе шаблона





Рисунки 14-15 Настроила каталог курса

**Вывод**

В этой лабораторной работе я научилась работать с git и подключать удаленные репозитории, добавлять и удалять необходимые файлы. Я также научилась использовать Git Flow, который значительно упрощает разработку проекта и навигацию между ветвями.

**Контрольные вопросы**

1. Системы контроля версий -VCS- это программное обеспечение, которое используется для облегчения работы с изменяющейся информацией, обычно - в проектах. Чаще всего используется при разработке, когда над одним проектом работает большое количество людей. 2.

* Репозиторий в системе контроля версий - Это удаленный репозиторий, в котором хранятся все файлы проекта
* commit - фиксирует изменения перед загрузкой файлов в систему контроля версий
* история хранит все изменения в проекте, и при необходимости позволяет откатиться в желаемое место
* рабочая копия - - это копия проекта на компьютере разработчика. Если другой член команды изменил проект, вам необходимо скачать новую версию проекта на свой компьютер.

1. 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. В децентрализованных системах у каждого из участников проекта есть полная копия проекта на своем компьютере, что делает его менее зависимым от сервера (Git).
2. Для начала необходимо создать и подключить удаленный репозиторий. Затем, поскольку никто, кроме вас, не изменяет проект, по мере изменения проекта отправляйте изменения на сервер, и нет необходимости загружать изменения.
3. 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.
4. Упрощение обмена информацией, ускорение разработки, устранение ошибок и недочетов при разработке. 7.

* git init - инициализирует локальный репозиторий
* git add \* или add. - добавляет файлы в репозиторий
* git commit - версия фиксации
* git pull - загружает текущую версию проекта
* git push - отправляет измененный проект на сервер
* git checkout - позволяет переключаться между ветками
* git status - текущий статус проекта
* git branch - просмотреть доступные ветки
* git remote add - добавить удаленный репозиторий

1. git push --all (push origin master/любой branch)
2. Ветви функций, также иногда называемые ветвями тем, используются для разработки новых функций, которые должны появиться в текущих или будущих выпусках.
3. Существуют временные и системные файлы, которые загромождают проект и не нужны. путь к ним можно добавить в файл .gitignore, тогда они не будут добавлены в проект