Отчёта по лабораторной работе №2

Ханина Людмила Константиновна

Содержание

# Цель работы

Изучить идеологию и применение средств контроля версий, научиться управляться с git.

# Задание

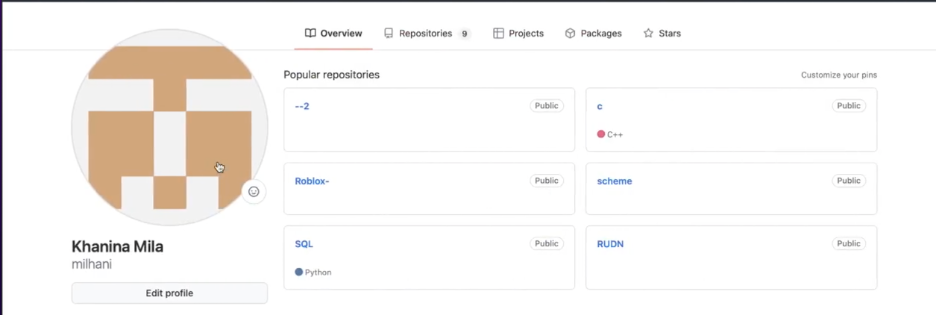
* Создать базовую конфигурацию для работы с git.
* Создать ключ SSH.
* Создать ключ PGP.
* Настроить подписи git.
* Зарегистрироваться на Github.
* Создать локальный каталог для выполнения заданий по предмету.

# Теоретическое введение

| Команда | Описание команды |
| --- | --- |
| sudo dnf install gh | Установка gh |
| git config --global user.name "Name Surname" | Обозначения имени владельца репозитория |
| git config --global user.email "work@mail" | Обозначения почты владельца репозитория |
| git config –-global core.quotepath false | Настройка utf-8 в выводе сообщений в git |
| ssh-keygen -t rsa -b 4096 | Создание SSH ключа |
| gpg --full-generate-key | Создание PGP ключа |
| gpg --list-secret-keys --keyid-format LONG | Вывод списка ключей |
| gh auth login | Авторизация в gh |
| rm package.json | Удаление лишних файлов из репозитория |

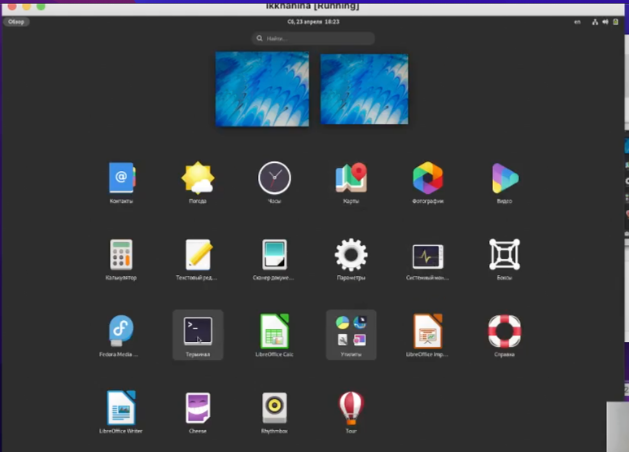
# Выполнение лабораторной работы

1. Создаем аккаунт на Github (у меня уже был).

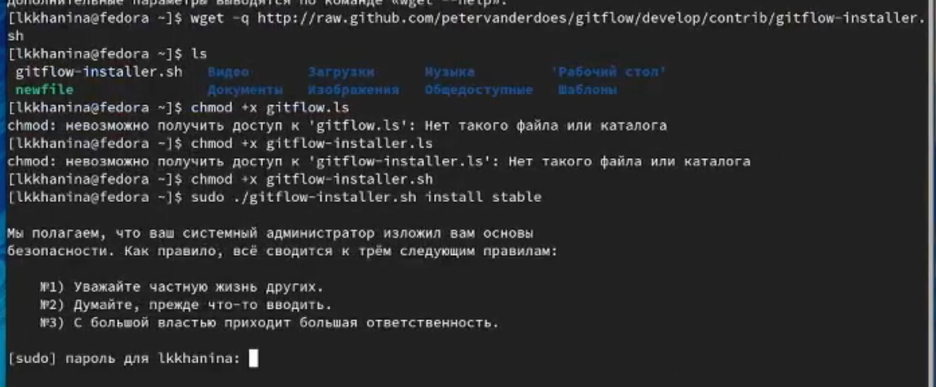


Аккаунт на Github

1. Запускаем виртуальную машину, заходим в терминал. Устанавливаем git-flow в Fedora Linux с помощью команд ниже.



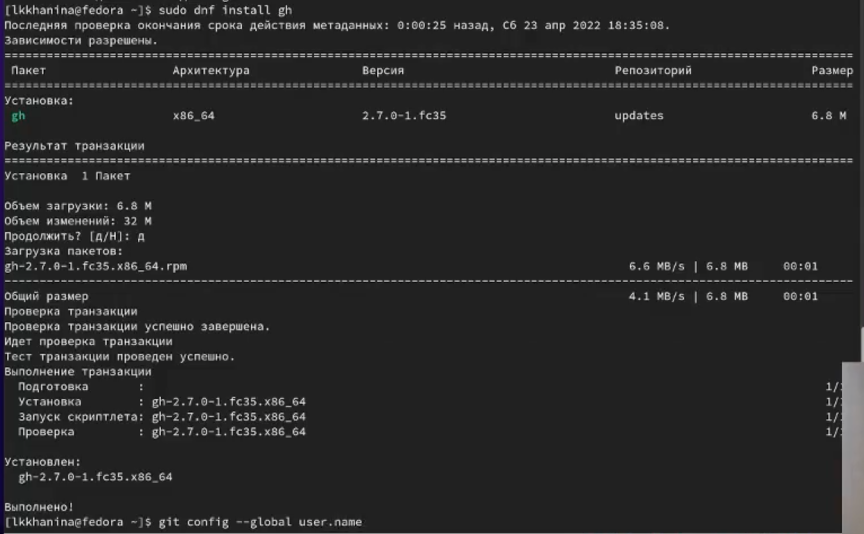
Запускаем виртуальную машину



Устанавливаем git-flow

1. Далее устанавливаем gh с помощью команды

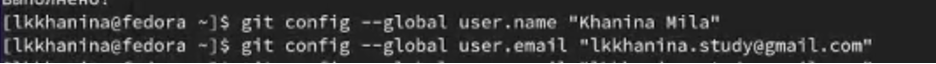
sudo dnf install gh



Устанавливаем gh

1. Начинаем работу с git. Сделаем базовые действия. Начнем с обозначения имени и почты владельца репозитория:

git config --global user.name "Name Surname"  
git config --global user.email "work@mail"



Обозначения имени и почты владельца репозитория

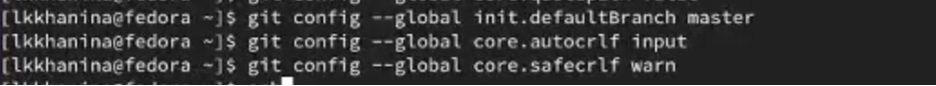
1. Далее настроим utf-8 в выводе сообщений в git с помощью команды:

git config –-global core.quotepath false

Настройка utf-8 в выводе сообщений в git

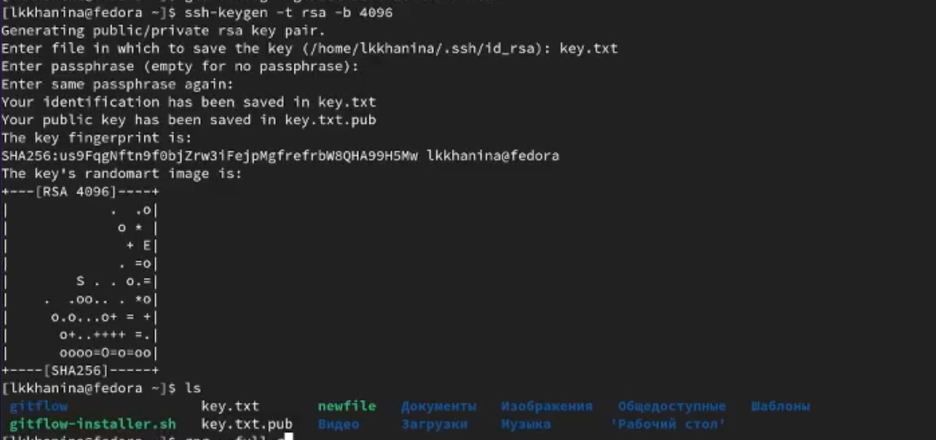
Настройка utf-8 в выводе сообщений в git

1. Теперь поработаем с ветками. Зададим имя начальной ветки и изменим параметры autocrlf и safecrlf.

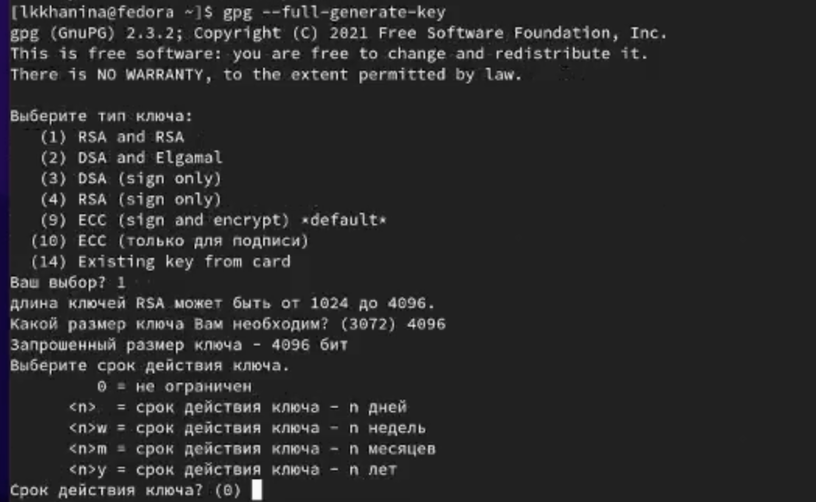


Имя начальной ветки и изменение параметров autocrlf и safecrlf

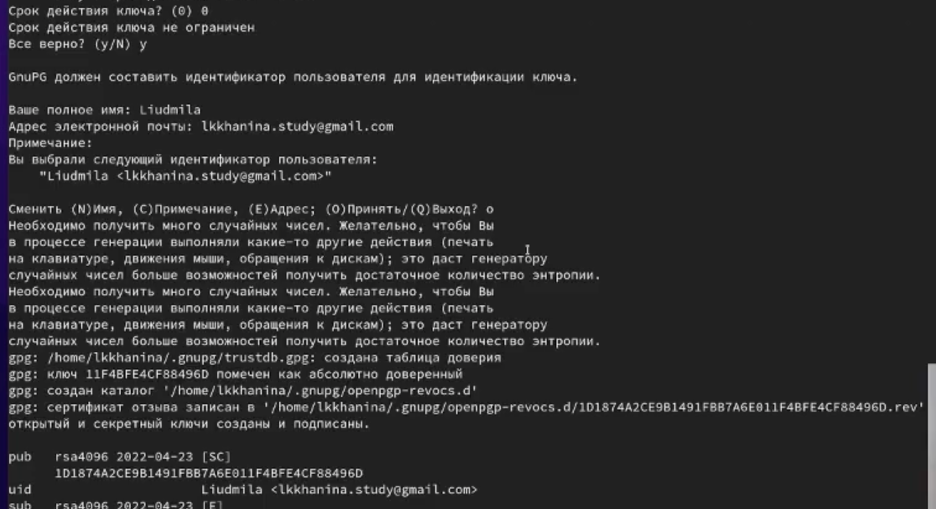
1. Следующий шаг — создание ключей SSH и PGP.



Создание ключа SSH



Создание ключа PGP #1

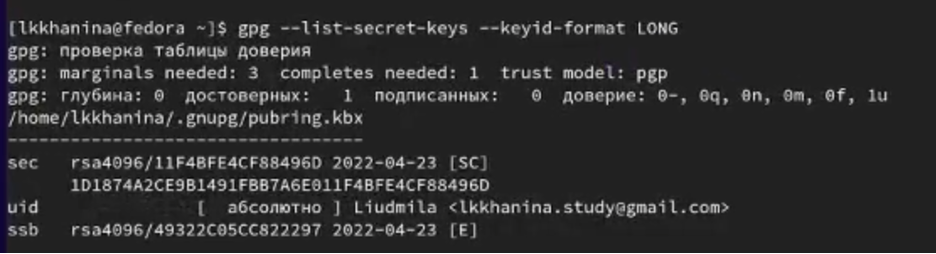


Создание ключа PGP #2

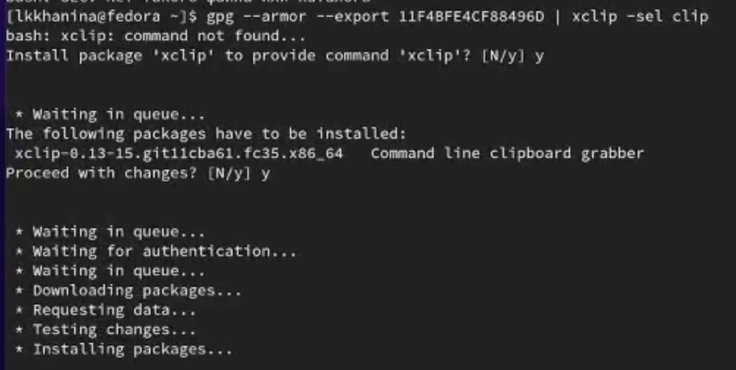
1. Далее необходимо добавить ключ PGP в Github. Для этого нужно вывести все ключи, найти Отпечаток ключа, скопировать его, а потом с помощью команды

gpg –-armor –-export <PGP Fingerprint> | xclip -sel clip

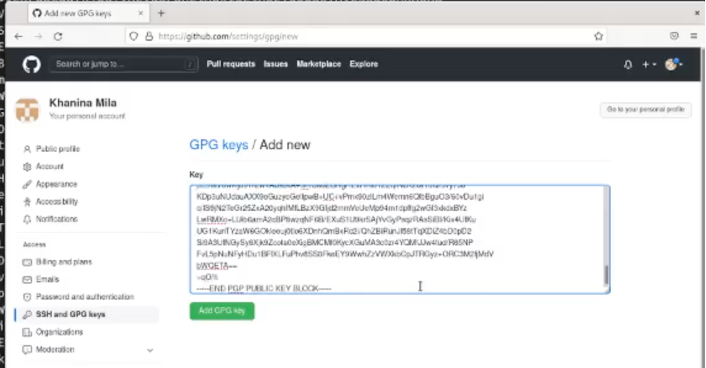
скопировать ключ, чтобы вставить его в Github.



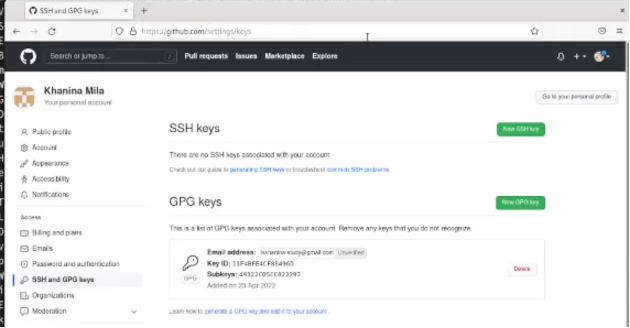
Вывод ключей



Копирование ключа

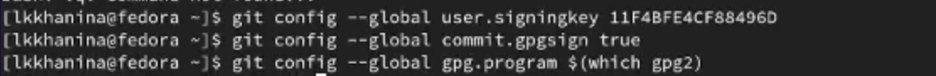


Добавление ключа в Github #1



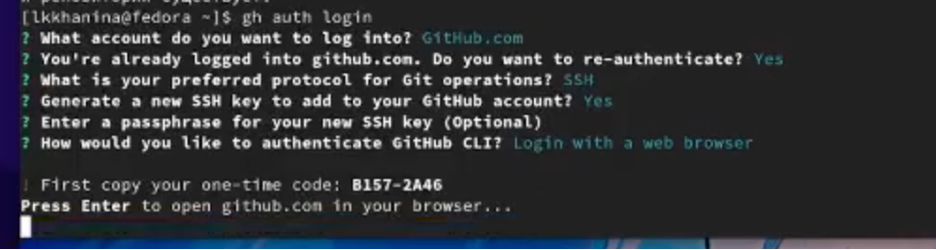
Добавление ключа в Github #2

1. Теперь можем настроить подпись коммитов.

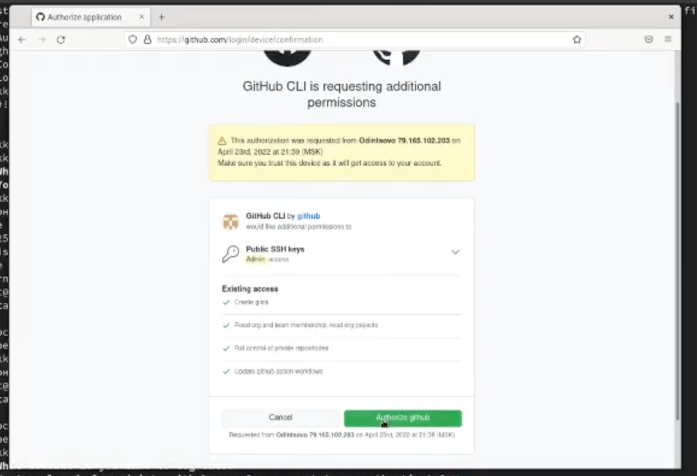


Настройка подписей коммитов

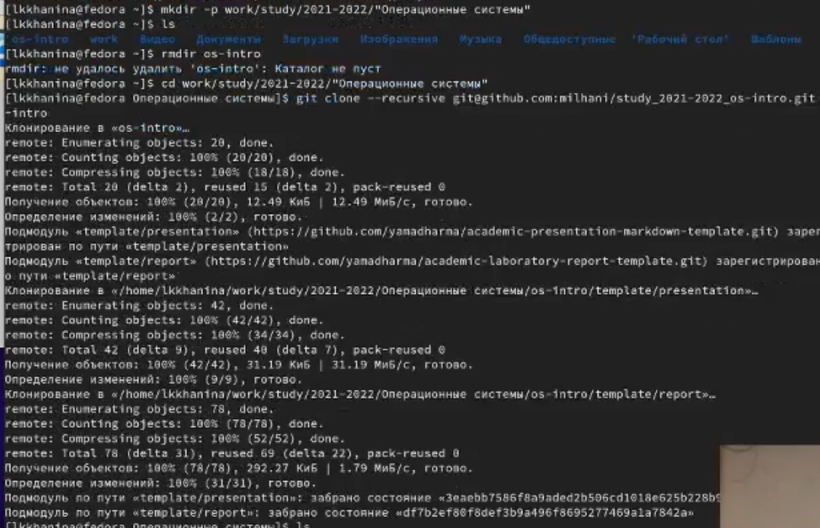
1. Авторизовываемся в gh и создаем шаблон для рабочего пространства.



Авторизация в gh #1



Авторизация в gh #2



Создание шаблона

1. Переходим к настройкам каталога курса. Удаляем лишние файлы с помощью команды

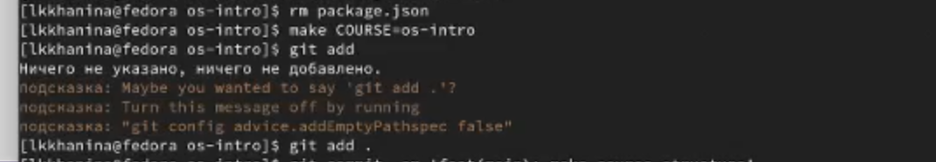
rm package.json

Затем создаем нужные каталоги с помощью команды

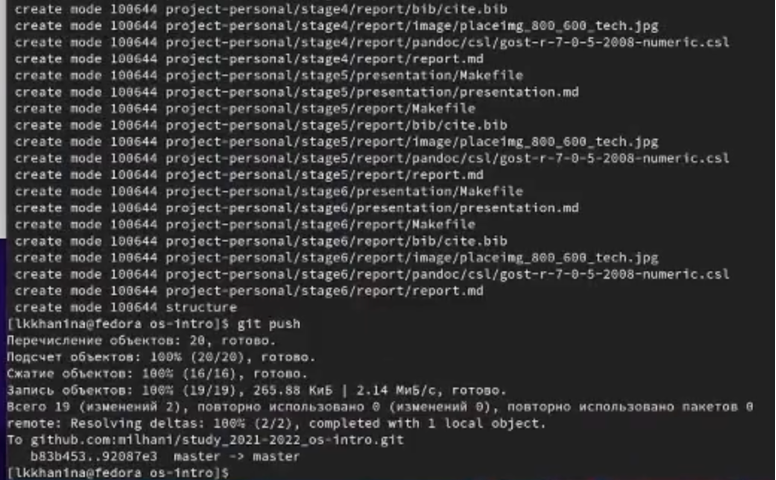
make COURSE=os-intro

И отправляем все файлы на сервер:

git add .  
git commit -am ‘feat(main): make course structure’  
git push



Настройка каталога курса #1



Настройка каталога курса #2

# Выводы

Создала рабочее пространство для дальнейшей работы. Разобралась в работе с git.

# Контрольные вопросы

1. Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. 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.
2. В классических системах контроля версий используется централизованная модель, предполагающая наличие единого 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 с помощью команды git commit.
3. Централизованная система — система, где один или несколько клиентских узлов напрямую подключены к центральному серверу. Пример — википедия. Децентрализованная — система, где каждый узел подключается к любому другому серверу. Пример — Bitcoin.
4. Создадим локальный репозиторий. Сначала сделаем предварительную конфигурацию, указав имя и email владельца репозитория:

git config --global user.name "Имя Фамилия"  
git config --global user.email "work@mail"

Для инициализации локального репозитория, расположенного, например, в каталоге ~/tutorial, необходимо ввести в командной строке:

cd   
mkdir tutorial  
cd tutorial  
git init

1. Основные задачи — хранение информации и обеспечение удобной командной работы над кодом.
2. Ветви нужны для хранения различных версий состояния хранилища. С помощью них довольно удобно перемещаться между версиями.