

**Факультет физико-математических и естественных наук**  
**Кафедра прикладной информатики и теории вероятностей**

## **Отчет по лабораторной работе №2**

*Дисциплина: Операционные системы*

Студент: Латыпова Диана

Группа: НФИбд-02-21

**Москва**

2022 г.

## Управление версиями.

**Цель работы:** изучить идеологию и применение средств контроля версий, освоить умения по работе с git.

**Ход работы:**

Сначала я создала учетную запись на <https://github.com>.

При создании аккаунта в гит на моем компьютере произошла ошибка, поэтому я создала учетную запись на чужом компьютере (Рис. 2.1 а). Затем вошла в аккаунт со своего компьютера и заполнила основную информацию (Рис. 2.1 б).

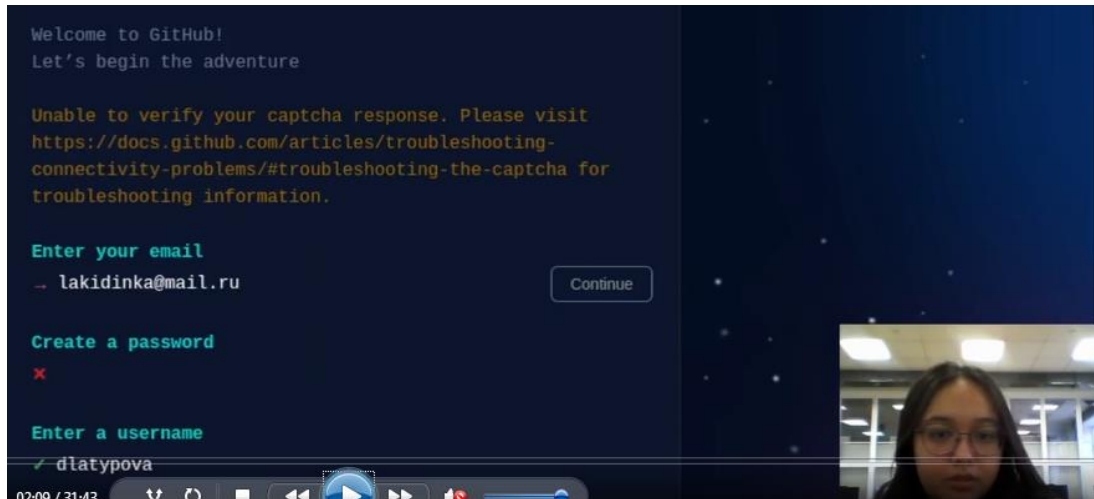


Рис. 2.1 а. Ошибка создания учетной записи в git

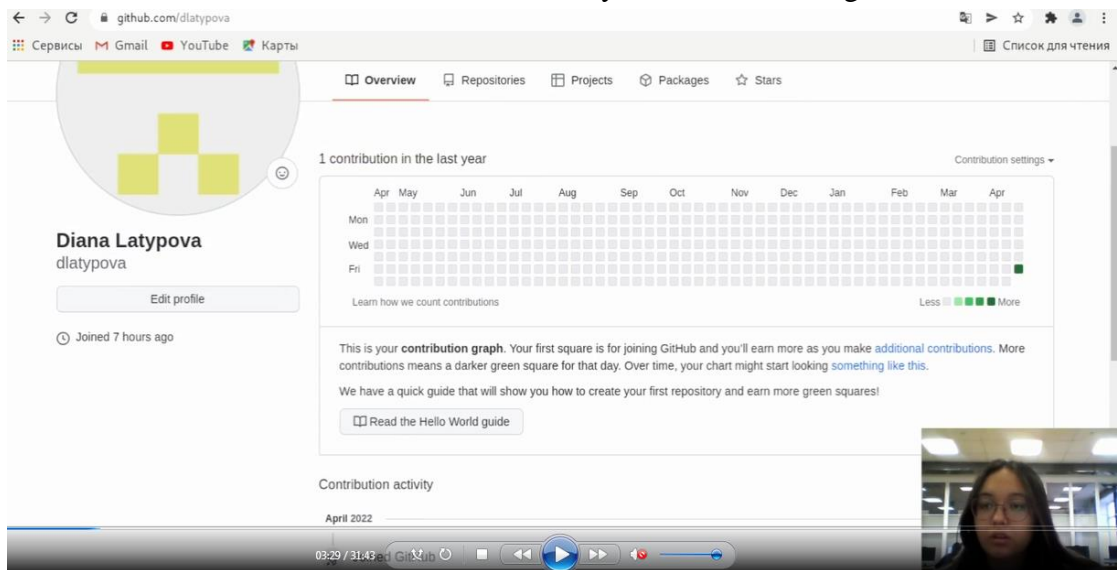


Рис. 2.1 б. Вход в аккаунт dlatypova, заполнение основной информации.

Далее я установила вручную программное обеспечение git-flow в терминале с помощью команд:

1 `cd /tmp`

2 `wget --no-check-certificate -q`

`https://raw.githubusercontent.com/petervanderdoes/gitflow/develop/contrib/gitflow-installer.sh`

3 `chmod +x gitflow-installer.sh`

4 `sudo ./gitflow-installer.sh install stable`

И ввела пароль (Рис. 2.2).

```
[dlatihpova@fedora ~]$ cd /tmp
[dlatihpova@fedora tmp]$ wget --no-check-certificate -q https://raw.githubusercontent.com/petervanderdoes/gitflow/develop/contrib/gitflow-installer.sh
[dlatihpova@fedora tmp]$ chmod +x gitflow-installer.sh
[dlatihpova@fedora tmp]$ sudo ./gitflow-installer.sh install stable
[sudo] пароль для dlatihpova:
### git-flow no-make installer ###
Installing git-flow to /usr/local/bin
Cloning repo from GitHub to gitflow
Клонирование в «gitflow»...
remote: Enumerating objects: 4270, done.
Получение объектов: 55% (2349/4270), 940.01 КиБ | 299.00 КиБ/с
```

Рис. 2.2. Установка программного обеспечения git-flow

Далее установила gh (Рис.2.3.).

```
'gitflow/hooks/pre-flow-release-track' -> '/usr/local/share/doc/gitflow/hooks/pre-f
low-release-track'
[dlatihpova@fedora tmp]$ sudo dnf install gh
google-chrome 5.3 kB/s | 3.6 kB 00:00
Зависимости разрешены.
=====
Пакет Архитектура Версия Репозиторий Размер
=====
Установка:
gh x86_64 2.7.0-1.fc35 updates 6.8 М
Результат транзакции
=====
Установка 1 Пакет
Объем загрузки: 6.8 М
Объем изменений: 32 М
Продолжить? [д/н]: д
Загрузка пакетов:
gh-2.7.0-1.fc35.x86_64.rpm 949
-----
Общий размер 592
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверка транзакции
Тест транзакции проведен успешно.
```

Рис.2.3. Установка gh

После чего занялась базовой настройкой git (Рис. 2.4.). Задала имя и email владельца репозитория:

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

Настроила utf-8 в выводе сообщений git:

**git config --global core.quotePath false**

Настроила верификацию и подписание коммитов git и задала имя начальной ветки:

**git config --global init.defaultBranch master**

– Параметр autocrlf:

**git config --global core.autocrlf input**

– Параметр safecrlf:

**git config --global core.safecrlf warn**

```

gh-2.7.0-1.fc35.x86_64
Выполнено!
[dlatihpova@fedora tmp]$ git config --global user.name "Diana Latypova"
[dlatihpova@fedora tmp]$ 2 git config --global user.email "lakidinka@mail.ru"
bash: 2: command not found...
[dlatihpova@fedora tmp]$ 2 git config --global user.email "lakidinka@mail"
bash: 2: command not found...
[dlatihpova@fedora tmp]$ git config --global user.email "lakidinka@mail.ru"
[dlatihpova@fedora tmp]$ git config --global core.quotePath false
[dlatihpova@fedora tmp]$ git config --global init.defaultBranch master
[dlatihpova@fedora tmp]$ 1 git config --global core.autocrlf input
bash: 1: command not found...
[dlatihpova@fedora tmp]$ git config --global core.autocrlf input
[dlatihpova@fedora tmp]$ git config --global core.safecrlf warn
[dlatihpova@fedora tmp]$ ssh-keygen -t rsa -b 4096

```

Рис. 2.4. Базовая настройка git

Создала ssh ключ (Рис. 2.5.):

--по алгоритму rsa с ключём размером 4096 бит:

**ssh-keygen -t rsa -b 4096**

--по алгоритму ed25519:

**ssh-keygen -t ed25519**



```

[dlatihpova@fedora tmp]$ git config --global core.safecrlf warn
[dlatihpova@fedora tmp]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/dlatihpova/.ssh/id_rsa):
Created directory '/home/dlatihpova/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/dlatihpova/.ssh/id_rsa
Your public key has been saved in /home/dlatihpova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:g6SF8mVLVZie05BCnKrZL89J1mL259SEM9ZFDIaSVUs dlatihpova@fedora
The key's randomart image is:
+---[RSA 4096]-----+
|  o.. .oo.o.oEo. |
| + ooo.o .o o. |
| ...oo+o . . |
| . + ++. o . |
|.o = ..S = o |
|.o .. . o = |
| .o = . o . |
| .+. . + |
| o+ . |
+---[SHA256]-----+
[dlatihpova@fedora tmp]$
[dlatihpova@fedora tmp]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/dlatihpova/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/dlatihpova/.ssh/id_ed25519
Your public key has been saved in /home/dlatihpova/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:XtLXRIyHF2rLgVY0m0dBipB13PaNmpnPZkQJfauZ/xE dlatihpova@fedora
The key's randomart image is:
+--[ED25519 256]--+
| .+0. .+ |
| ..+0... |
| ..0oB=.. |
| . o X=o. |
| S o ++0E |
| . o .=@. . |
| . o ++. |
| . =.. |
| o o |
+-----+

```

Рис. 2.5. Создание ssh ключа

Создала ключи pgp(Рис.2.6. а-в):

Генерировала ключ

**gpg --full-generate-key**

Из предложенных опций выбрала:

- тип RSA and RSA;
- размер 4096;
- срок действия; значение по умолчанию— 0 (срок действия не истекает никогда).
- GPG запросил личную информацию, которая сохранилась в ключе:

- Имя (Diana Latypova).
- Адрес моей электронной почты.
- Комментарий (оставила пустым).

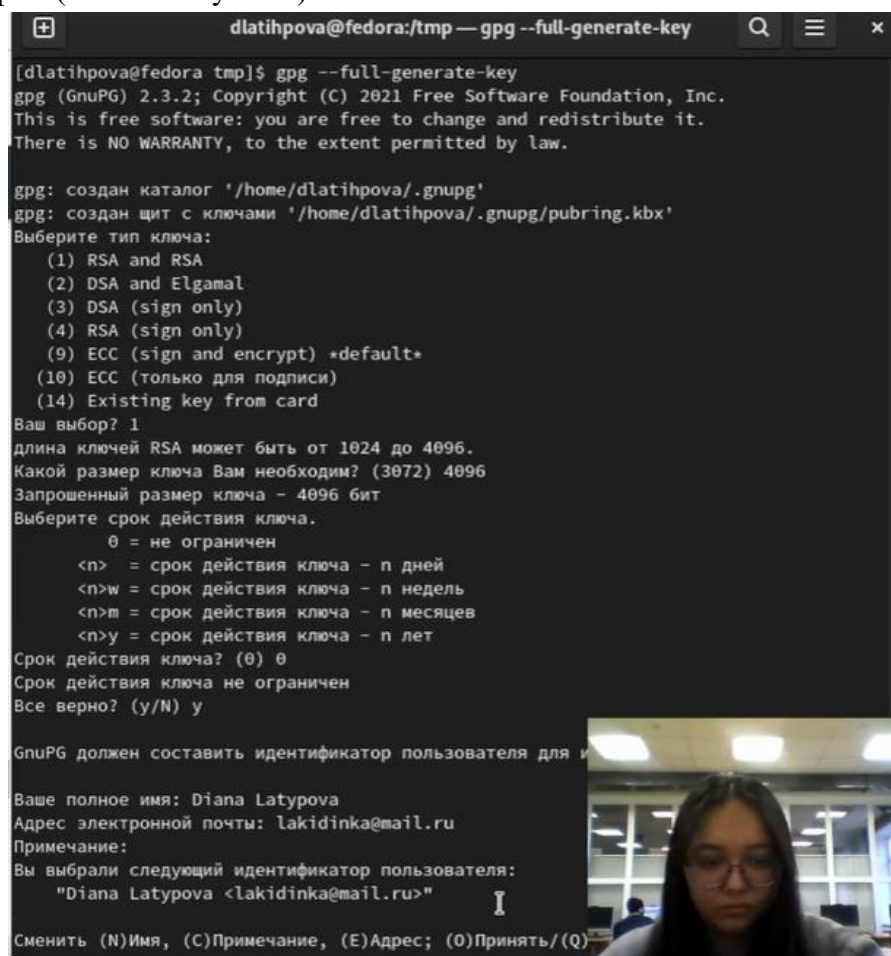


Рис. 2.6 а. Создание ключа ргр

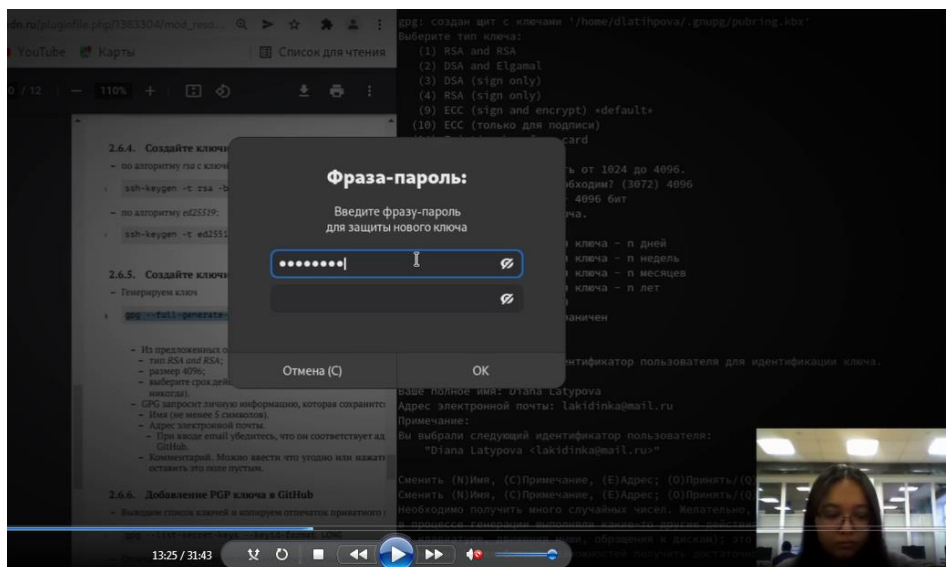


Рис. 2.6 б. Ввод фразы-пароля



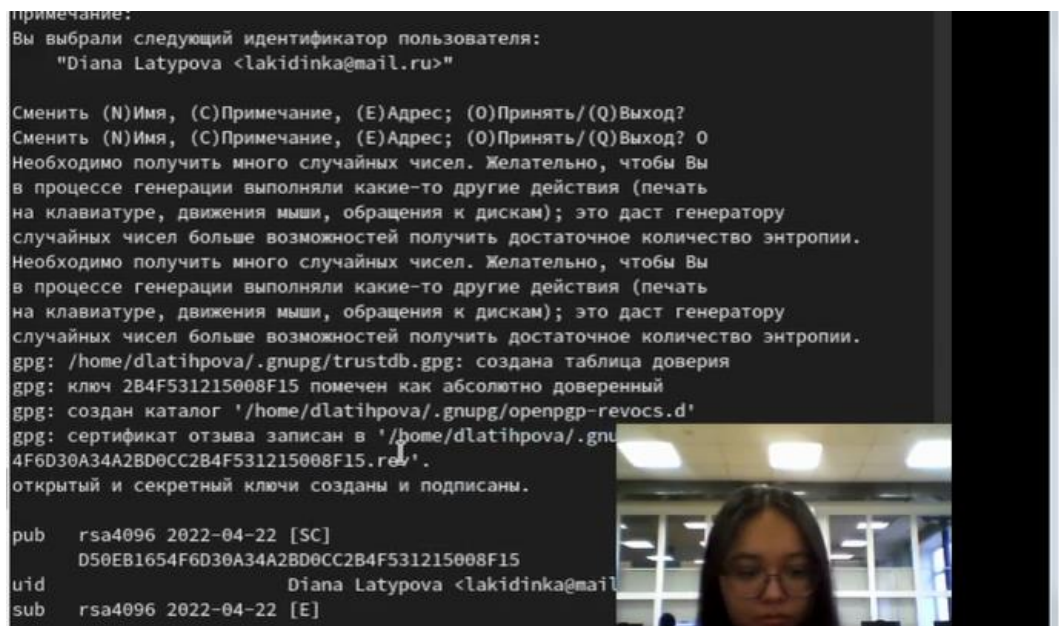


Рис. 2.6 в. Продолжение создания ключа gpg

Далее нужно было добавить gpg ключ в GitHub (Рис.2.7а.). Вывела список ключей и скопировала отпечаток приватного ключа:

**gpg --list-secret-keys --keyid-format LONG**

Скопировала сгенерированный PGP ключ в буфер обмена:

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

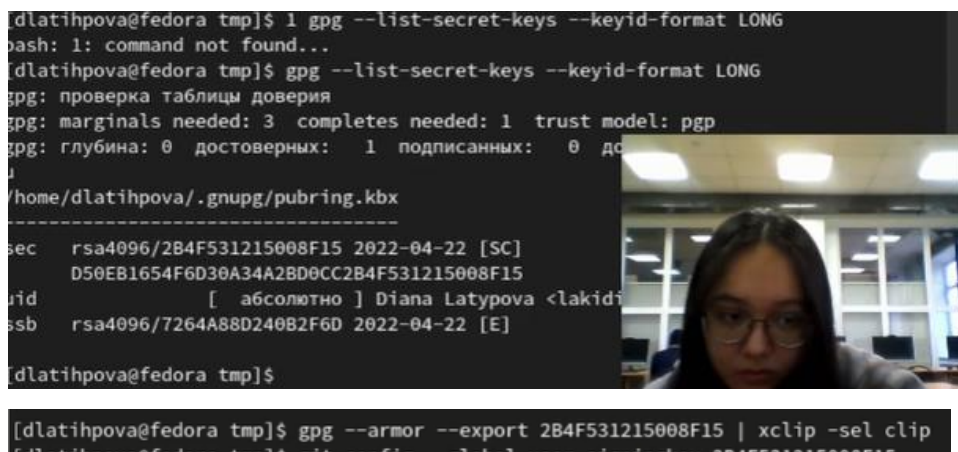


Рис.2.7а. Вывод списка ключей и копирование сгенерированного PGP ключа в буфер обмена

После чего перешла в настройки GitHub (<https://github.com/settings/keys>), нажала на кнопку New GPG key и вставила полученный ключ в поле ввода (Рис.2.7б).

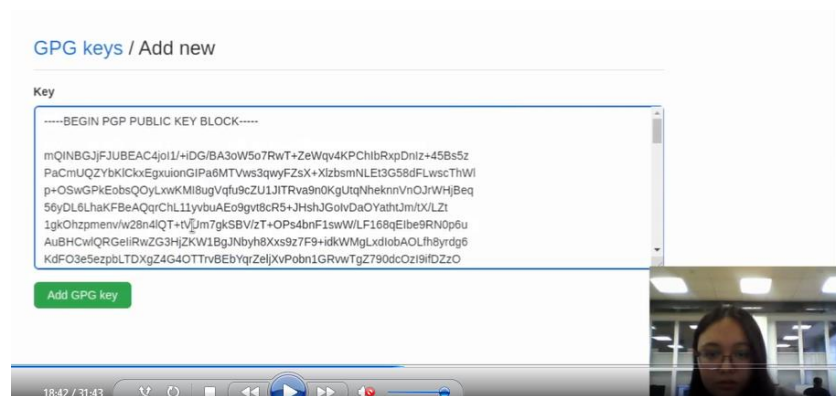


Рис.2.7. б. Добавление полученного ключа gpg в поле ввода

Далее я настроила автоматические подписи коммитов git. С помощью следующих коммитов (Рис.2.8.):

- 1 **git config --global user.signingkey <PGP Fingerprint>**
- 2 **git config --global commit.gpgsign true**
- 3 **git config --global gpg.program \$(which gpg2)**

```
[dlatihpova@fedora tmp]$ gpg --armor --export 2B4F531215008F15 | xclip -sel clip
[dlatihpova@fedora tmp]$ git config --global user.signingkey 2B4F531215008F15
[dlatihpova@fedora tmp]$ git config --global commit.gpgsign true
[dlatihpova@fedora tmp]$ git config --global gpg.program $(which gpg2)
[dlatihpova@fedora tmp]$ gh auth login
```

Рис.2.8. Настройка автоматических подписей коммитов git

Настроила gh с помощью **gh auth login** (Рис.2.9.)

```
[dlatihpova@fedora tmp]$ gpg --armor --export 2B4F531215008F15 | xclip -sel clip
[dlatihpova@fedora tmp]$ git config --global user.signingkey 2B4F531215008F15
[dlatihpova@fedora tmp]$ git config --global commit.gpgsign true
[dlatihpova@fedora tmp]$ git config --global gpg.program $(which gpg2)
[dlatihpova@fedora tmp]$ gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations? HTTPS
? Authenticate Git with your GitHub credentials? Yes
? How would you like to authenticate GitHub CLI? Login with a web browser

! First copy your one-time code: 1A3C-17A9
Press Enter to open github.com in your browser...
Окно или вкладка откроется в текущем сеансе браузера.
libva error: vaGetDriverNameByIndex() failed with unknown libva error, driver_name
= (null)
[8835:8835:0100/000000.358077:ERROR:sandbox_linux.cc(377)] InitializeSandbox() call
ed with multiple threads in process gpu-process.
✓ Authentication complete.
- gh config set -h github.com git_protocol https
✓ Configured git protocol
✓ Logged in as dlatypova
[dlatihpova@fedora tmp]$ mkdir -p ~/work/study/2021-2022
```

Рис.2.9. Настройка gh

Создала шаблон рабочего пространства (Рис.2.10.):

- 1 **mkdir -p ~/work/study/2021-2022/"Операционные системы"**
- 2 **cd ~/work/study/2021-2022/"Операционные системы"**
- 3 **gh repo create study\_2021-2022\_os-intro--template=yamadharm/course-directory-student-template --public**
- 4 **cat ~/.ssh/id\_rsa.pub | xclip -sel clip**
- 5 **git clone --recursive git@github.com:<owner>/study\_2021-2022\_os-intro.git os-intro**

```
[dlatihpova@fedora tmp]$ mkdir -p ~/work/study/2021-2022/"Операционные системы"
[dlatihpova@fedora tmp]$ cd ~/work/study/2021-2022/"Операционные системы"
[dlatihpova@fedora Операционные системы]$ gh repo create study_2021-2022_os-intro --
-template=yamadharm/course-directory-student-template --public
✓ Created repository dlatypova/study_2021-2022_os-intro on GitHub

[dlatihpova@fedora Операционные системы]$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
[dlatihpova@fedora Операционные системы]$ git clone --recursive git@github.com:<own
er>/study_2021-2022_os-intro.git os-intro
bash: owner: Нет такого файла или каталога
[dlatihpova@fedora Операционные системы]$ git clone --recursive git@github.com:<own
er>/study_2021-2022_os-intro.git os-intro
bash: owner: Нет такого файла или каталога
[dlatihpova@fedora Операционные системы]$ git clone --recursive git@github.com:dlat
ypova/study_2021-2022_os-intro.git os-intro
Клонирование в «os-intro»...
The authenticity of host 'github.com (140.82.121.4)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvC0qU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Host key verification failed.
fatal: Не удалось прочитать из внешнего репозитория.

Удостоверьтесь, что у вас есть необходимые права доступ
```

Рис.2.10. Создание репозитория курса на основе шаблона

И наконец, настроила каталог курса (Рис.2.11. а-в):

**cd ~/work/study/2021-2022/"Операционные системы"/os-intro**

– Удалила лишние файлы:

**rm package.json**

– Создайте необходимые каталоги:

**1 make COURSE=os-intro**

– Отправила файлы на сервер:

**1 git add .**

**2 git commit -am 'feat(main): make course structure'**

**3 git push**

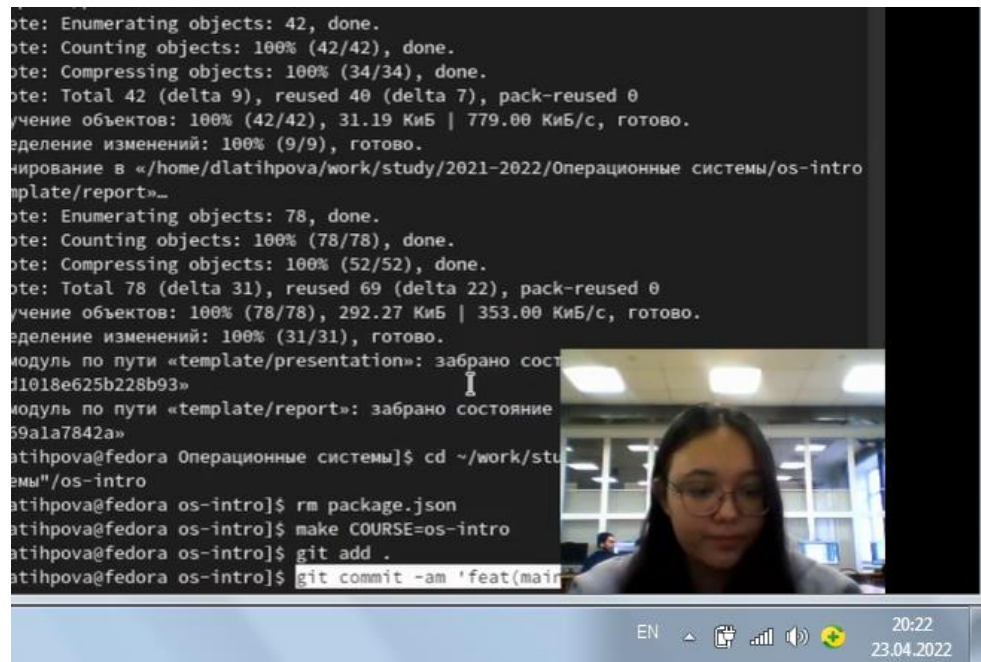


Рис.2.11а. Настройка каталога курса

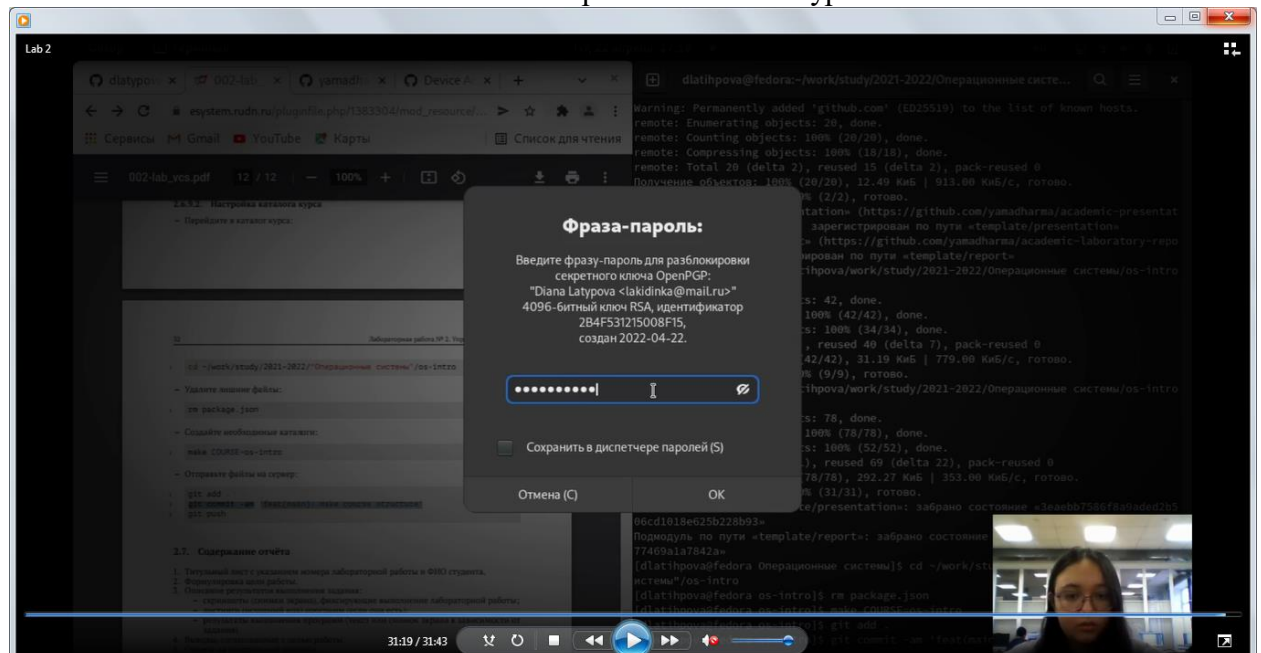


Рис.2.11б. Ввод фразы-пароля



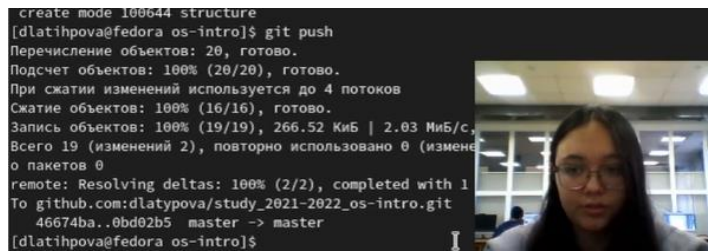


Рис.2.11в. Продолжение настройки каталога курса

### **Контрольные вопросы:**

1. Система контроля версий — это система, записывающая изменения в файл или набор файлов в течение времени и позволяющая вернуться позже к определённой версии.

В свете усложнения сред разработки они помогают командам разработчиков работать быстрее и эффективнее. Системы контроля версий наиболее полезны командам DevOps, поскольку помогают сократить время разработки и увеличить количество успешных развертываний.

Программное обеспечение контроля версий отслеживает все вносимые в код изменения в специальной базе данных. При обнаружении ошибки разработчики могут вернуться назад и выполнить сравнение с более ранними версиями кода для исправления ошибок, сводя к минимуму проблемы для всех участников команды.

2. - Хранилище –репозиторий - место хранения всех версий и служебной информации.

- Команда `git commit` делает для проекта снимок текущего состояния изменений, добавленных в раздел проиндексированных файлов
- История – место, где сохраняются все коммиты, по которым можно посмотреть данные о коммитах.
- Рабочая копия –текущее состояние файлов проекта, основанное на версии, загруженной из хранилища.

3. Централизованные системы контроля версий представляют собой приложения типа клиент-сервер, когда репозиторий проекта существует в единственном экземпляре и хранится на сервере. Доступ к нему осуществляется через специальное клиентское приложение. В качестве примеров таких программных продуктов можно привести CVS, Subversion.

Децентрализованные VCS позволяют хранить репозиторий (его копию) у каждого разработчика, работающего с данной системой. При этом можно выделить центральный репозиторий (условно), в который будут отправляться изменения из локальных и, с ним же эти локальные репозитории будут синхронизироваться. При работе с такой системой, пользователи периодически синхронизируют свои локальные репозитории с центральным и работают непосредственно со своей локальной копией.

6. У Git две основных задачи: первая — хранить информацию обо всех изменениях в вашем коде, начиная с самой первой строчки, а вторая — обеспечение удобства командной работы над кодом.

7. Основные команды `git`:

- Команда `git add` добавляет содержимое рабочего каталога в индекс (staging area) для последующего коммита.
- Команда `git status` показывает состояния файлов в рабочем каталоге и индексе: какие файлы изменены, но не добавлены в индекс; какие ожидают коммита в индексе.
- Команда `git diff` используется для вычисления разницы между любыми двумя Git деревьями.
- Команда `git difftool` просто запускает внешнюю утилиту сравнения для показа различий в двух деревьях, на случай если вы хотите использовать что-либо отличное от встроенного просмотрщика `git diff`.
- Команда `git commit` берёт все данные, добавленные в индекс с помощью `git add`, и сохраняет их слепок во внутренней базе данных, а затем сдвигает указатель текущей ветки на этот слепок.
- Команда `git rm` используется в Git для удаления файлов из индекса и рабочей копии.
- Команда `git mv` — это всего лишь удобный способ переместить файл, а затем выполнить `git add` для нового файла и `git rm` для старого.
- Команда `git clean` используется для удаления мусора из рабочего каталога.

**Вывод:** я изучила идеологию и применение средств контроля версий, а также освоила умения по работе с `git`.