

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

дисциплина: Операционные системы

**Студент:** Кондратьев Арсений  
Вячеславович

**Группа:** НПИбд-01-21

**Ст. билет №:** 1132210645

Москва

2022

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

**Ход работы:**

1. Создал учетную запись на Github(рис.1)

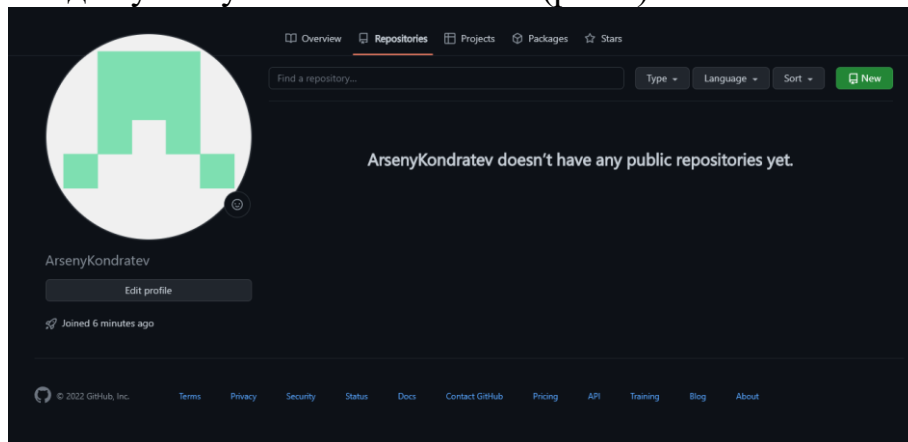


Рис. 1

2. Установи Git-flow(рис. 2)

```
> cd /tmp
> wget --no-check-certificate -q https://raw.githubusercontent.com/petervanderdoes/gitflow/develop/contrib/gitflow-installer.sh
↪ /gitflow/develop/contrib/gitflow-installer.sh
> chmod +x gitflow-installer.sh
> sudo ./gitflow-installer.sh install stable
```

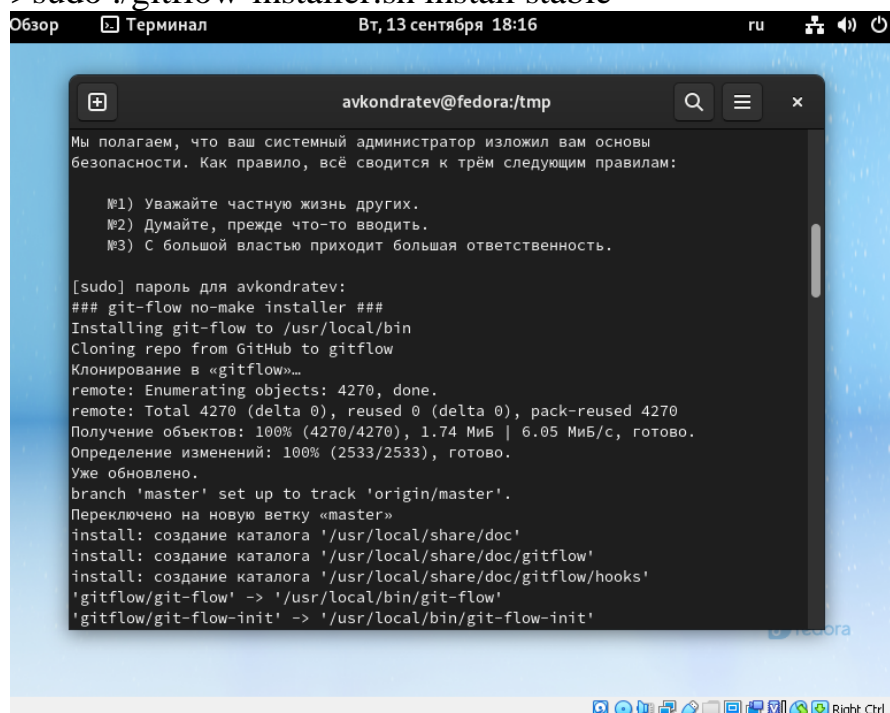


Рис. 2

### 3. Установил gh в Fedora Linux(рис.3)

> sudo dnf install gh

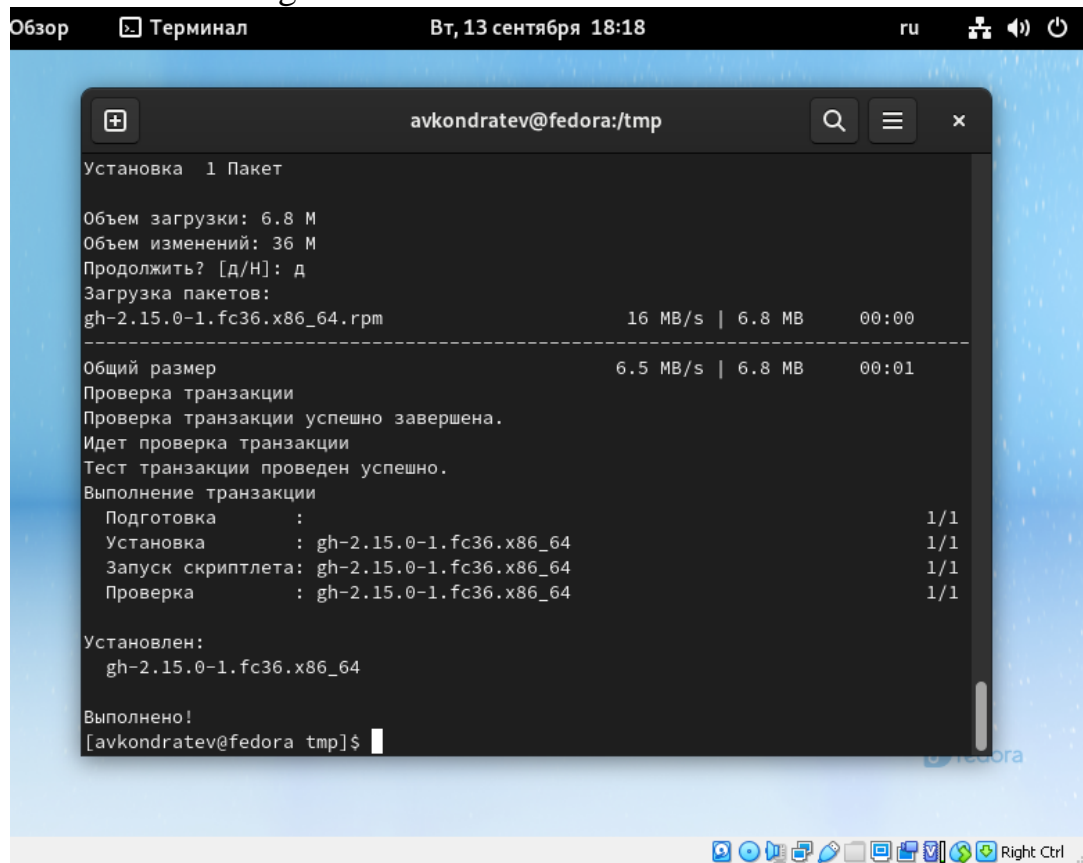


Рис. 3

### 4. Выполнил базовую настройку Git(рис. 4)

> git config --global user.name "KondratevArseny"  
> git config --global user.email avk2200@yandex.ru  
> git config --global core.quotepath false  
> git config --global init.defaultBranch master  
> git config --global core.autocrlf input  
> git config --global core.safecrlf warn

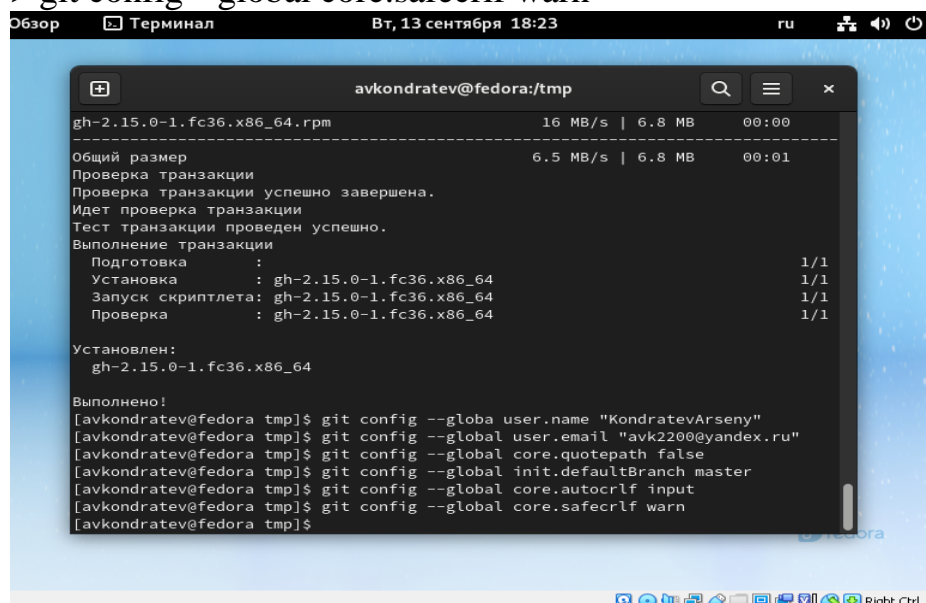
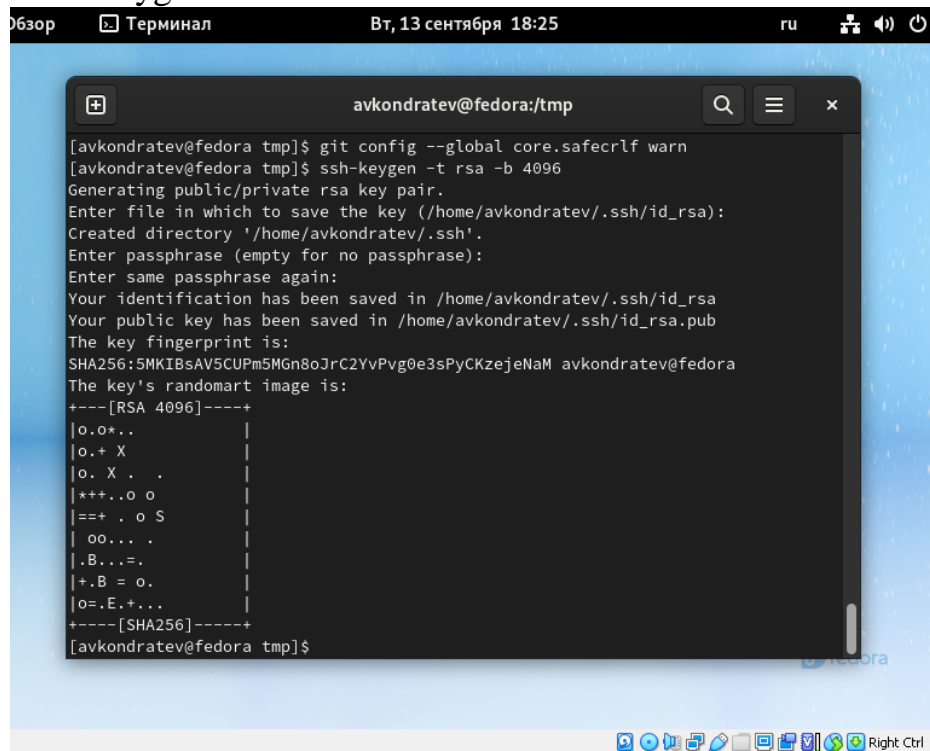


Рис. 4

## 5. Создал ключ SSH по алгоритму rsa(рис. 5)

> ssh-keygen -t rsa -b 4096

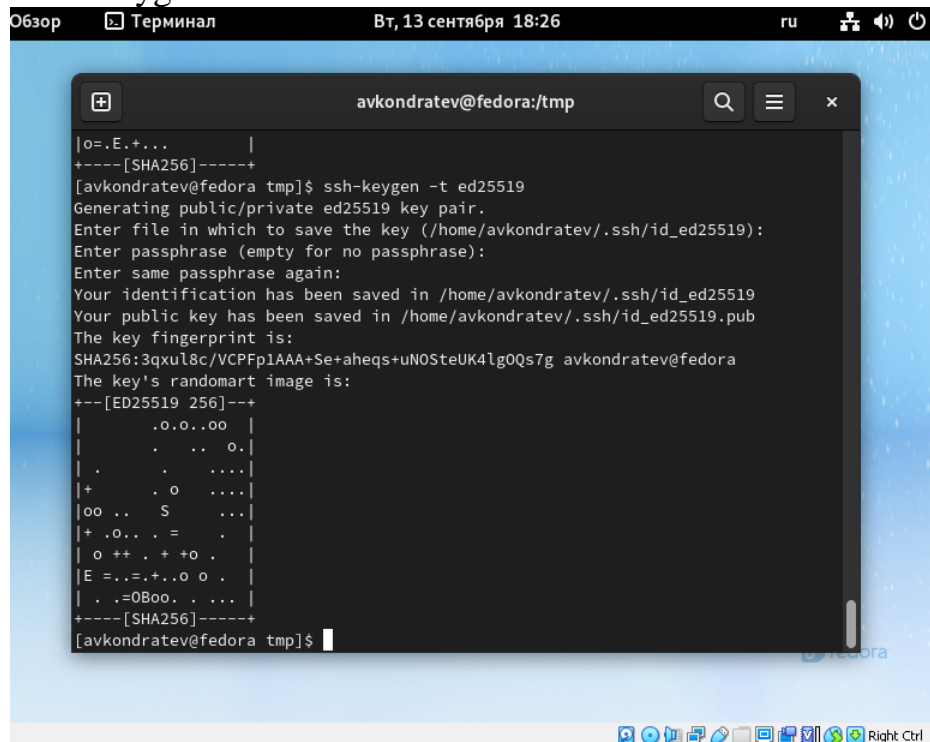


```
Обзор Терминал Вт, 13 сентября 18:25 ru [иконки] [иконка] [иконка]
avkondratev@fedora:tmp
[avkondratev@fedora tmp]$ git config --global core.safecrlf warn
[avkondratev@fedora tmp]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/avkondratev/.ssh/id_rsa):
Created directory '/home/avkondratev/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/avkondratev/.ssh/id_rsa
Your public key has been saved in /home/avkondratev/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:5MKIBsAV5CUPm5MGn8oJrC2YvPvg0e3sPyCKzejeNaM avkondratev@fedora
The key's randomart image is:
+---[RSA 4096]-----+
|o.o*..|
|o.+ X |
|o. X . |
|*+..o o|
|==+ . o S|
| oo... |
|.B...=|
|+.B = o.|
|o=.E.+...|
+----[SHA256]-----+
[avkondratev@fedora tmp]$
```

Рис. 5

## 6. Создал ключ SSH по алгоритму ed25519(рис. 6)

> ssh-keygen -t ed25519

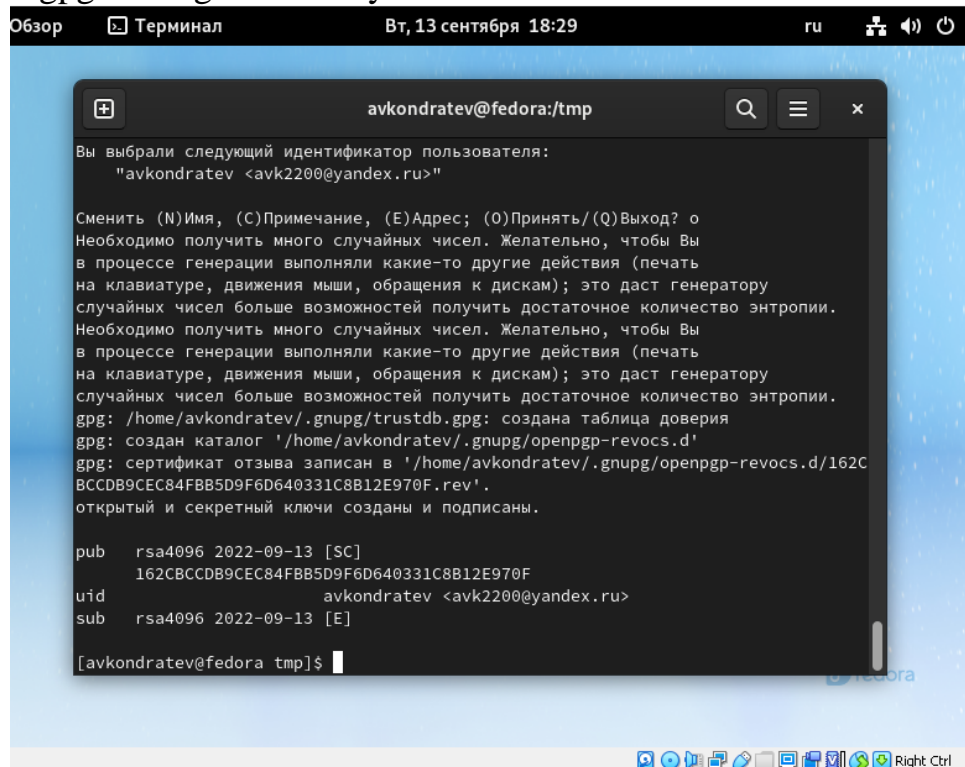


```
Обзор Терминал Вт, 13 сентября 18:26 ru [иконки] [иконка] [иконка]
avkondratev@fedora:tmp
[avkondratev@fedora tmp]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/avkondratev/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/avkondratev/.ssh/id_ed25519
Your public key has been saved in /home/avkondratev/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:3qxul8c/VCPFp1AAA+Se+aeqqs+uN0SteUK4lg0Qs7g avkondratev@fedora
The key's randomart image is:
+--[ED25519 256]--+
| .o.o..oo |
| . . . o |
| . . ....|
|+ . o ....|
|oo .. S ...|
|+ .o.. = .|
| o ++ . +o .|
|E =..=+.o o .|
| . .oBoo. . ...|
+----[SHA256]-----+
[avkondratev@fedora tmp]$
```

Рис. 6

## 7. Создал ключи pgp(рис. 7)

> gpg --full-generate-key



```
Обзор Терминал Вт, 13 сентября 18:29 ru

avkondratev@fedora:tmp

Вы выбрали следующий идентификатор пользователя:
"avkondratev <avk2200@yandex.ru>"

Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? o
Необходимо получить много случайных чисел. Желательно, чтобы Вы
в процессе генерации выполняли какие-то другие действия (печатать
на клавиатуре, движения мыши, обращения к дискам); это даст генератору
случайных чисел больше возможностей получить достаточное количество энтропии.
Необходимо получить много случайных чисел. Желательно, чтобы Вы
в процессе генерации выполняли какие-то другие действия (печатать
на клавиатуре, движения мыши, обращения к дискам); это даст генератору
случайных чисел больше возможностей получить достаточное количество энтропии.
gpg: /home/avkondratev/.gnupg/trustdb.gpg: создана таблица доверия
gpg: создан каталог '/home/avkondratev/.gnupg/openpgp-revocs.d'
gpg: сертификат отзыва записан в '/home/avkondratev/.gnupg/openpgp-revocs.d/162C
BCCDB9CEC84FBB5D9F6D640331C8B12E970F.rev'.
открытый и секретный ключи созданы и подписаны.

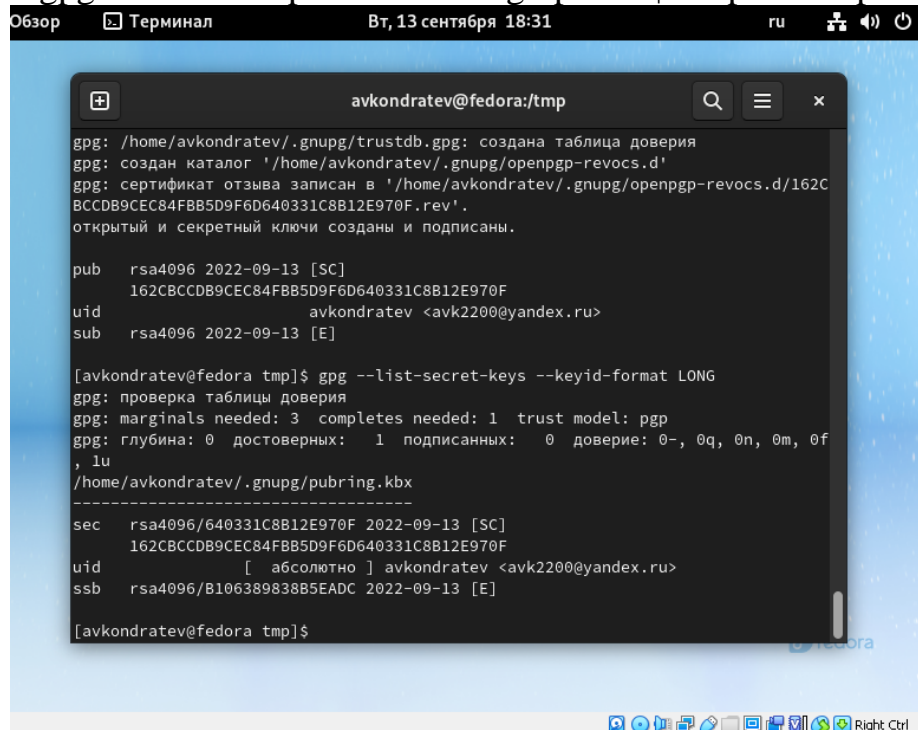
pub  rsa4096 2022-09-13 [SC]
    162CBCCDB9CEC84FBB5D9F6D640331C8B12E970F
uid                          avkondratev <avk2200@yandex.ru>
sub  rsa4096 2022-09-13 [E]

[avkondratev@fedora tmp]$
```

Рис. 7

## 8. Добавил ключ pgp в github(рис. 8, рис. 9)

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



```
Обзор Терминал Вт, 13 сентября 18:31 ru

avkondratev@fedora:tmp

gpg: /home/avkondratev/.gnupg/trustdb.gpg: создана таблица доверия
gpg: создан каталог '/home/avkondratev/.gnupg/openpgp-revocs.d'
gpg: сертификат отзыва записан в '/home/avkondratev/.gnupg/openpgp-revocs.d/162C
BCCDB9CEC84FBB5D9F6D640331C8B12E970F.rev'.
открытый и секретный ключи созданы и подписаны.

pub  rsa4096 2022-09-13 [SC]
    162CBCCDB9CEC84FBB5D9F6D640331C8B12E970F
uid                          avkondratev <avk2200@yandex.ru>
sub  rsa4096 2022-09-13 [E]

[avkondratev@fedora tmp]$ gpg --list-secret-keys --keyid-format LONG
gpg: проверка таблицы доверия
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: глубина: 0 достоверных: 1 подписанных: 0 доверие: 0-, 0q, 0n, 0m, 0f
, 1u
/home/avkondratev/.gnupg/pubring.kbx
-----
sec  rsa4096/640331C8B12E970F 2022-09-13 [SC]
    162CBCCDB9CEC84FBB5D9F6D640331C8B12E970F
uid      [ абсолютно ] avkondratev <avk2200@yandex.ru>
ssb  rsa4096/B106389838B5EADC 2022-09-13 [E]

[avkondratev@fedora tmp]$
```

Рис.8(вывел список ключей и скопировал список приватного ключа)

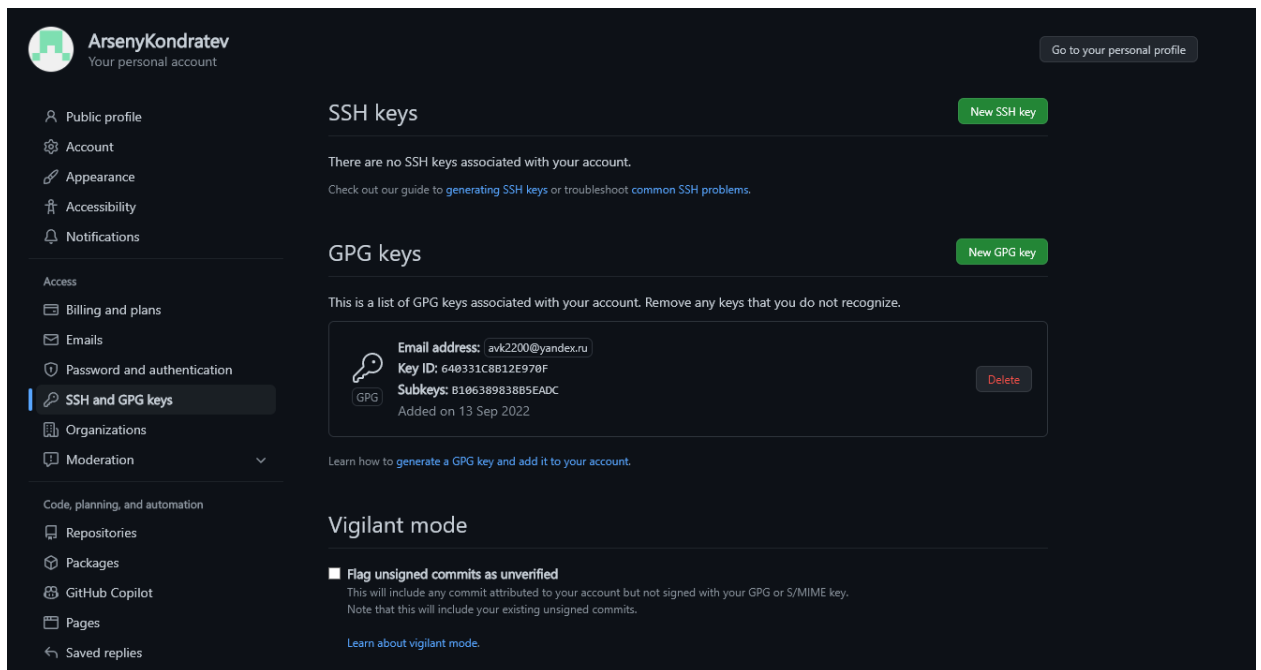


Рис. 9(Добавил ключ в Github)

## 9. Настроил автоматические подписи коммитов git(рис. 10)

> git config --global user.signingkey <PGP Fingerprint>

>git config --global commit.gpgsign true

>git config --global gpg.program \$(which gpg2)

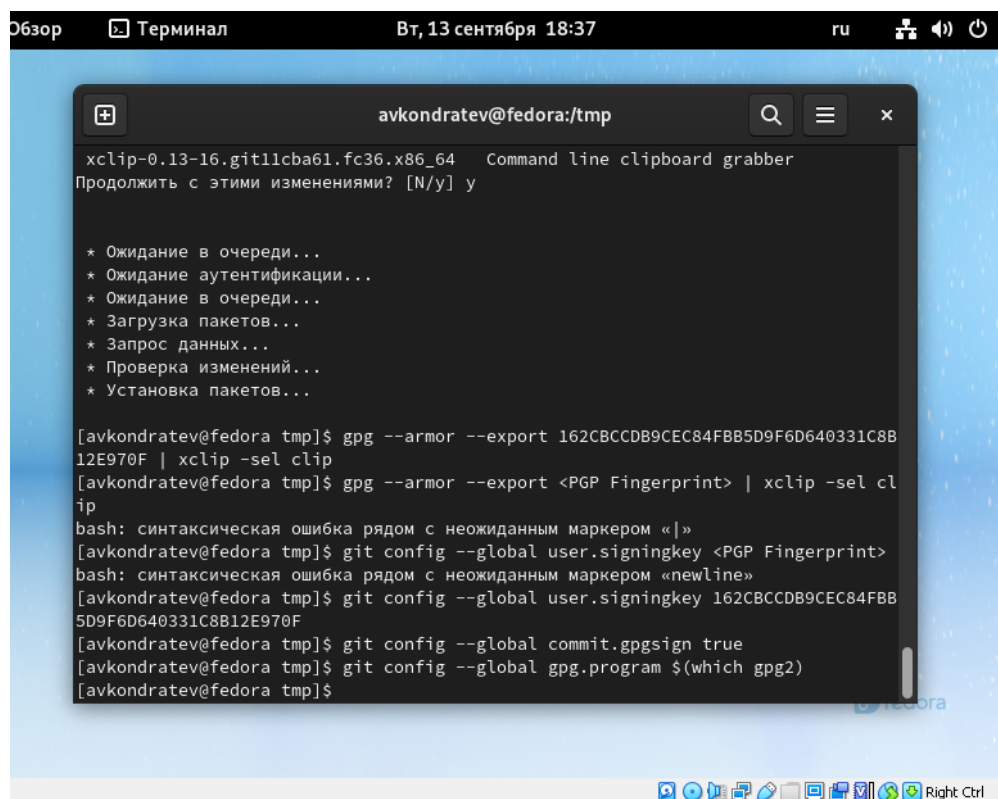


Рис. 10

10. Создал репозиторий курса на основе шаблона(рис. 11)

```
> mkdir -p ~/work/study/2021-2022/"Операционные системы"
```

```
> cd ~/work/study/2021-2022/"Операционные системы"
```

```
> gh repo create study_2021-2022_os-intro
```

```
↪ --template=yamadharma/course-directory-student-template --public
```

```
> git clone --recursive
```

```
↪ git@github.com:ArsenyKondratev study_2021-2022_os-intro.git os-intro
```

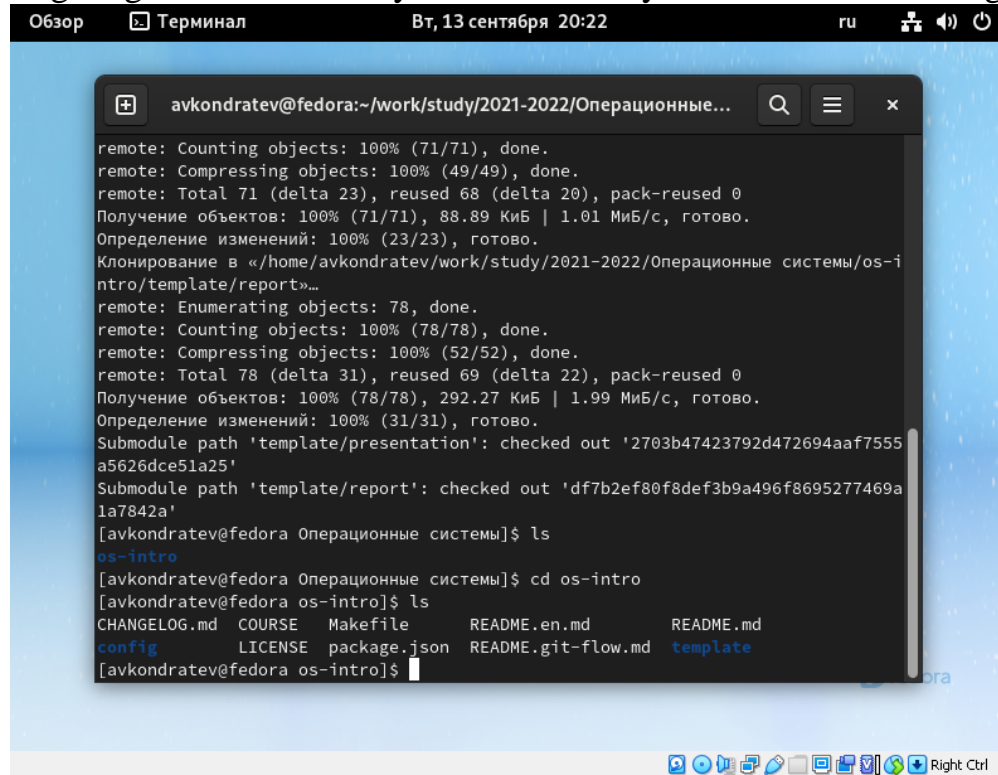


Рис. 11

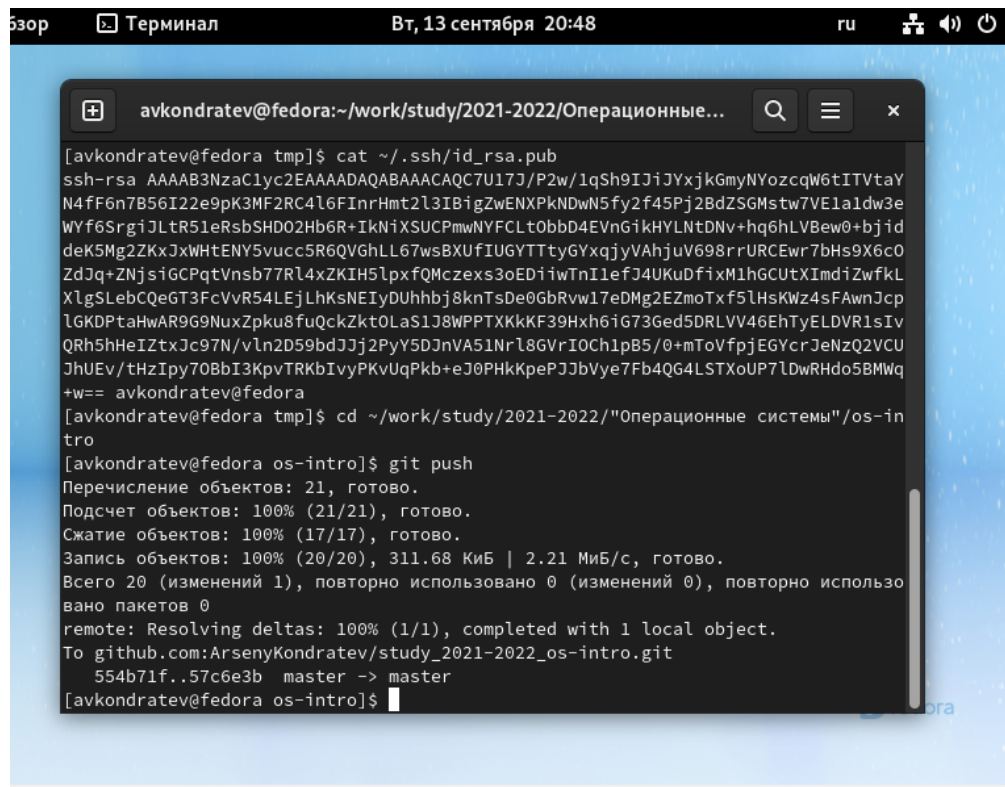
11. Настроил каталог курса(рис. 12)

Удалил лишние файлы и создал необходимый каталог

```
> rm package.json
```

```
> make COURSE=os-intro
```





```
[avkondratev@fedora tmp]$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAQC7U17J/P2w/1qSh9IJiJYxjkGmyNYozcqW6tITVtaY
N4ff6n7B56I22e9pK3MF2RC4l6FInrHmt2l3IBigZwENXPkNDwN5fy2f45Pj2BdZSGMstw7VE1a1dw3e
WYf6SrgiJLr51eRsbSHD02Hb6R+IkNiXSUCPmwNYFCLt0bbD4EVnGikHYLNTDNv+hq6hLVBew0+bjid
deK5Mg2ZKxJxWhtENY5vucc5R6QVGHLL67wsBXUfIU6YTTtyGYxqjyVAhjuV698rrURCEwr7bHs9X6c0
ZdJq+ZNjsiGCPqtVnsb77Rl4xZKIH5lpxfQMcexs3oEDiiwTnIlefJ4UKuDfixM1hGCuTXImdiZwfkL
XlgSLebCQeGT3FcVvR54LEjLhKsNEIyDUhhbj8knTsDe0GbRvw17eDMg2EZmoTxf5lHsKWz4sFAwnJcp
lGKDptaHwAR9G9NuxZpku8fuQckZkt0LaS1J8WPPTXKKKF39Hxh6iG73Ged5DRLVV46EhTyELDVr1sIv
QRh5hHeIZtxJc97N/vln2D59bdJJj2PyY5DJnVA51Nr18GVrIOChlpB5/0+mToVfpjEGYcrJeNzQ2VCU
JhUEv/tHzIpy70BbI3KpvTRKbIvyPKvUqPkb+eJ0PHkKpePJ3bVye7Fb4QG4LSTxoUP7LDwRHdo5BMWq
+w== avkondratev@fedora
[avkondratev@fedora tmp]$ cd ~/work/study/2021-2022/"Операционные системы"/os-in
tro
[avkondratev@fedora os-intro]$ git push
Перечисление объектов: 21, готово.
Подсчет объектов: 100% (21/21), готово.
Сжатие объектов: 100% (17/17), готово.
Запись объектов: 100% (20/20), 311.68 КиБ | 2.21 МиБ/с, готово.
Всего 20 (изменений 1), повторно использовано 0 (изменений 0), повторно использо
вано пакетов 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:ArsenyKondratev/study_2021-2022_os-intro.git
554b71f..57c6e3b master -> master
[avkondratev@fedora os-intro]$
```

Рис. 12

**Вывод:** Я изучил идеологию и применение средств контроля версий и освоил умения по работе с git.

### Контрольные вопросы:

1. Система контроля версий (Version Control System, VCS) — программное обеспечение для облегчения работы с изменяющейся информацией.  
VCS позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение.
2. Хранилище — место, где хранятся изменения кода.  
Commit - снимок состояния проекта на текущий момент времени.  
История — список снимков состояния проекта к которым можно при необходимости откатиться.  
Рабочая копия - Рабочая копия является снимком одной версии проекта.
3. Централизованные системы контроля версий представляют собой приложения типа клиент-сервер, когда репозиторий проекта существует в единственном экземпляре и хранится на сервере.(CVS, Subversion)  
Децентрализованные системы контроля версий (Distributed Version Control System, DVCS) позволяют хранить репозиторий (его копию) у каждого разработчика, работающего с данной системой.(Git, Mercurial)



4. Создаем свою ветку, базирующуюся на главной(`git checkout -b имя_ветки`), вносим изменения, делаем снимок(`git commit`) и затем вносим эти изменения в свою ветку(`git push`)
5. Отдельные ветки разработчиков внедряются в общую master ветку
6. Git позволяет нескольким разработчикам с удобством работать над одним проектом. Возможность получать изменения, внесенные другим человеком и откатываться на прошлые версии в случае ошибок.
7.
  - a. создание основного дерева репозитория – `git init`
  - b. получение обновлений (изменений) текущего дерева из центрального репозитория – `git pull`
  - c. отправка всех произведённых изменений локального дерева в центральный репозиторий – `git push`
  - d. просмотр списка изменённых файлов в текущей директории – `git status`
  - e. просмотр текущих изменений – `git diff`
  - f. добавить все изменённые и/или созданные файлы и/или каталоги – `git add`
  - g. сохранить все добавленные изменения и все изменённые файлы – `git commit`
  - h. создание новой ветки, базирующейся на текущей: - `git checkout -b имя_ветки`
  - i. переключение на некоторую ветку - `git checkout имя_ветки`
  - j. слияние ветки с текущим деревом - `git merge --no-ff имя_ветки`
  - k. удаление локальной уже слитой с основным деревом ветки - `git branch -d имя_ветки`
  - l. принудительное удаление локальной ветки - `git branch -D имя_ветки`
8. С локальным: `commit`(снимок состояния проекта)  
С удаленным: `push`(отправляем изменения) `pull`(загружаем изменения)
9. Это простой перемещаемый указатель на один из таких коммитов. Они нужны для того, чтобы разделять код. Например одна ветка у нас может быть основная для разработки. Если мы делаем новый функционал, то мы создаем новую ветку под него, а после окончания работы сливаем то, что мы сделали в основную ветку.
10. Во время работы могут появляться временные файлы, не несущие смысла для проекта. Их лучше не отправлять при использовании `commit`.