

# Отчет по лабораторной работе №3

## Операционная система

Старикова Евгения Дмитриевна

### Содержание

Цель работы .....	1
Задание.....	1
Выполнение лабораторной работы .....	1
Выводы .....	12

### Цель работы

Целью данной работы является изучение идеологии и применение средств контроля версий.

### Задание

– Сделайте отчёт по предыдущей лабораторной работе в формате Markdown. –  
В качестве отчёта просьба предоставить отчёты в 3 форматах: pdf, docx и imd (вархиве,  
поскольку он должен содержать скриншоты, Makefile ит.д.)

### Выполнение лабораторной работы

1. Создайте учётную запись на <https://github.com> (рис. [-@fig:001])

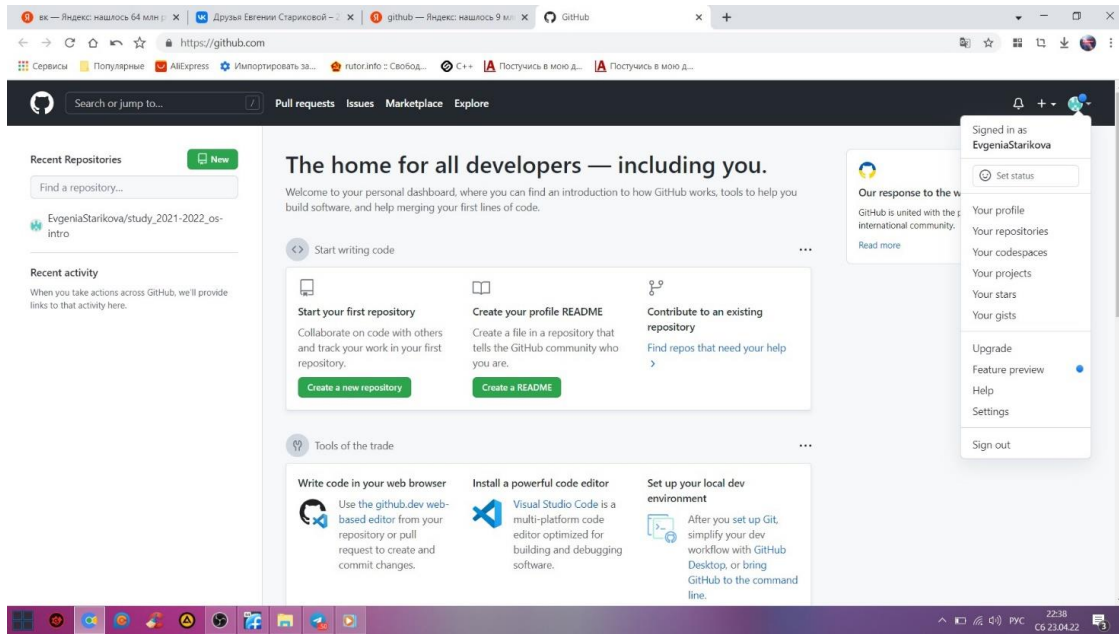
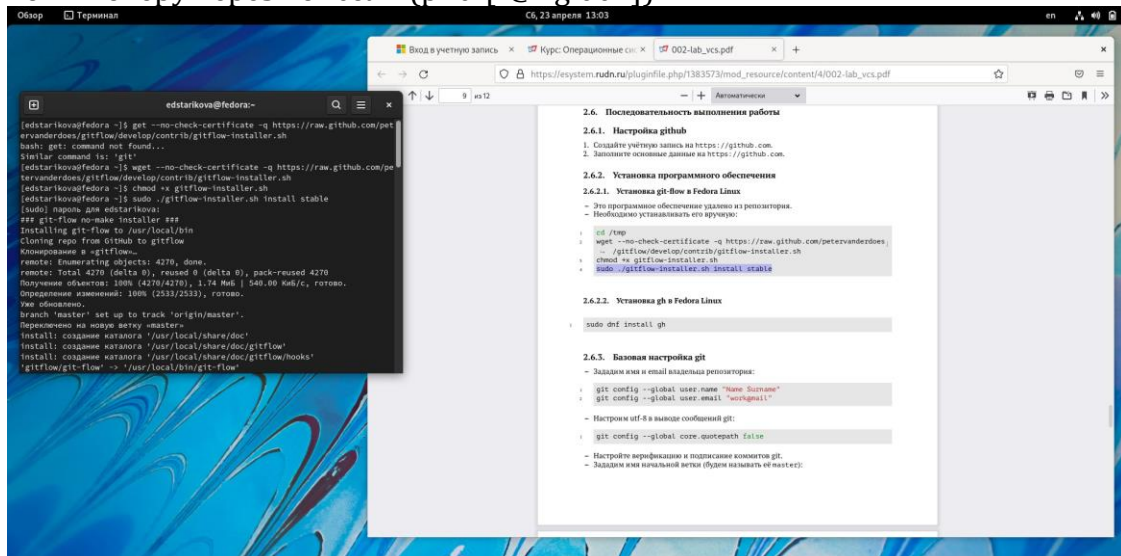
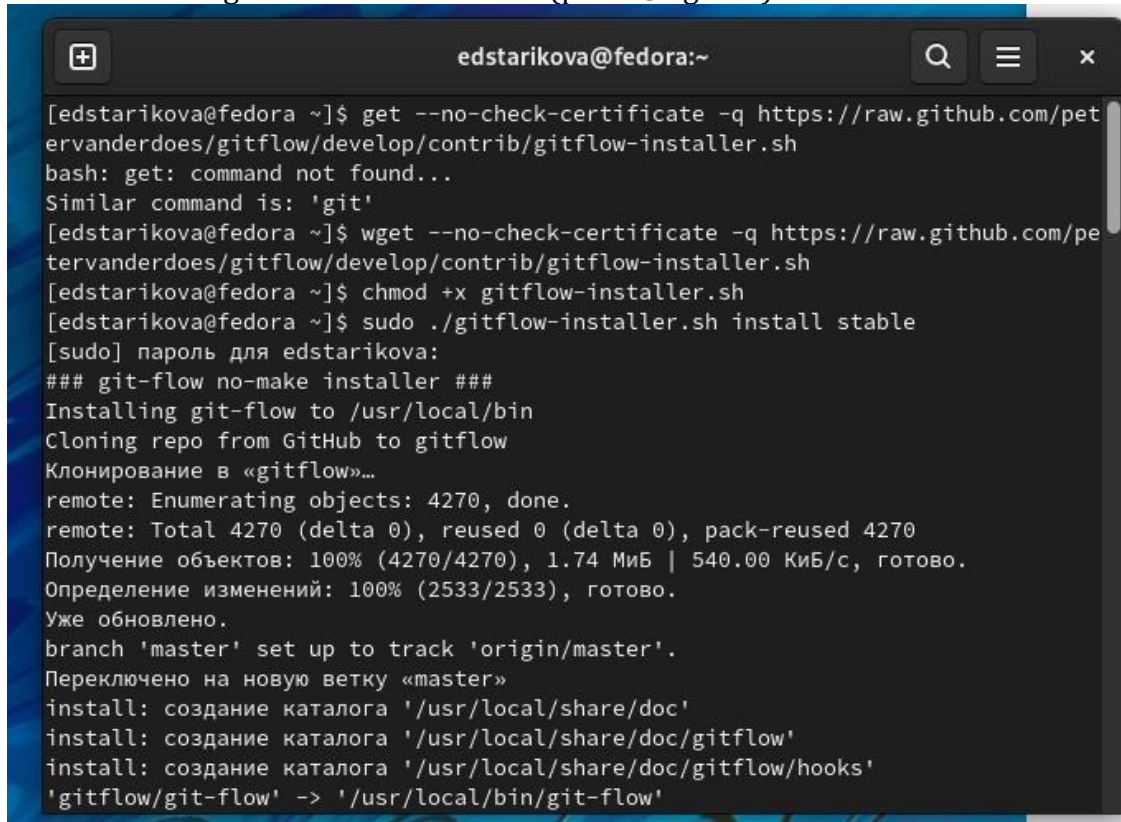


рис.1

2. Настраиваем систему контроля версий git. Синхронизируем учётную запись github с компьютером `git config --global user.name "EvgeniaStarikova"` `git config --global user.email "jenya.stari960687@gmail.com"` Затем создаём новый ключ на github `ssh-keygen -C "jenya.stari960687@gmail.com"` и привязываем его к компьютеру через консоль (рис. [-@fig:002])

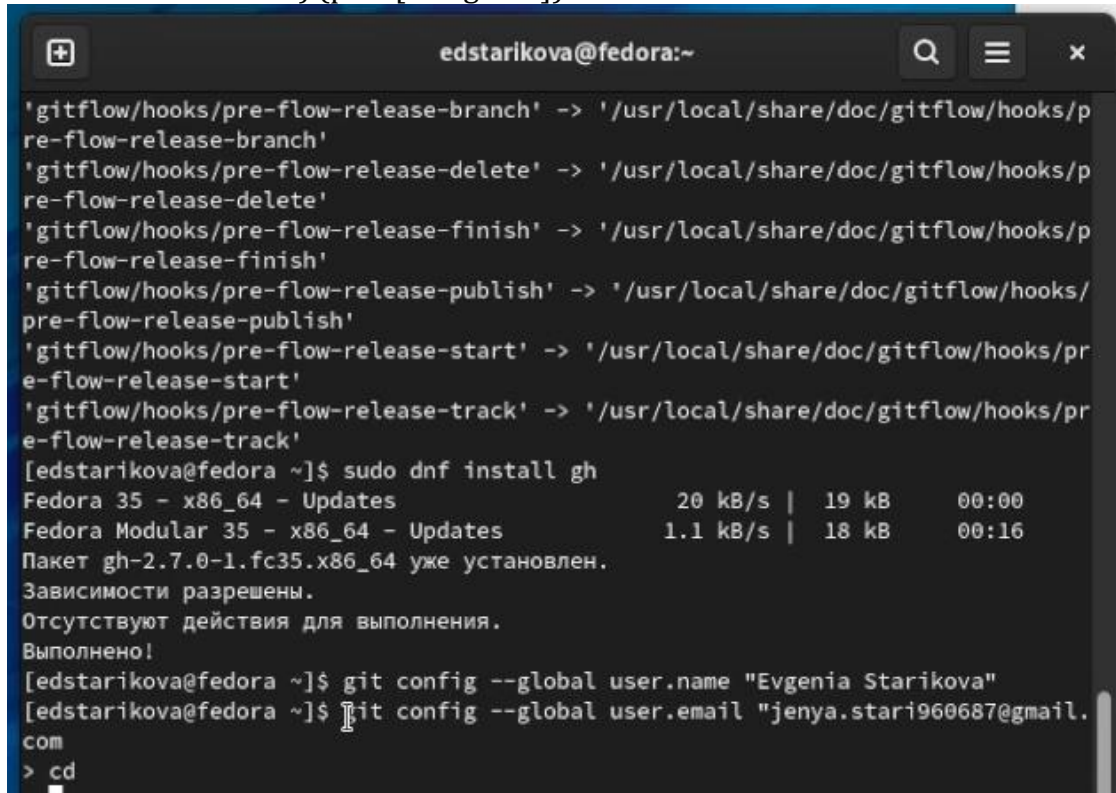


3. Устанавливаю git-flow в Fedora Linux (рис. -@fig:003)



```
edstarikova@fedora:~  
[edstarikova@fedora ~]$ get --no-check-certificate -q https://raw.githubusercontent.com/petervanderdoes/gitflow/develop/contrib/gitflow-installer.sh  
bash: get: command not found...  
Similar command is: 'git'  
[edstarikova@fedora ~]$ wget --no-check-certificate -q https://raw.githubusercontent.com/petervanderdoes/gitflow/develop/contrib/gitflow-installer.sh  
[edstarikova@fedora ~]$ chmod +x gitflow-installer.sh  
[edstarikova@fedora ~]$ sudo ./gitflow-installer.sh install stable  
[sudo] пароль для edstarikova:  
### 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.  
remote: Total 4270 (delta 0), reused 0 (delta 0), pack-reused 4270  
Получение объектов: 100% (4270/4270), 1.74 МиБ | 540.00 КиБ/с, готово.  
Определение изменений: 100% (2533/2533), готово.  
Уже обновлено.  
branch 'master' set up to track 'origin/master'.  
Переключено на новую ветку «master»  
install: создание каталога '/usr/local/share/doc'  
install: создание каталога '/usr/local/share/doc/gitflow'  
install: создание каталога '/usr/local/share/doc/gitflow/hooks'  
'gitflow/git-flow' -> '/usr/local/bin/git-flow'
```

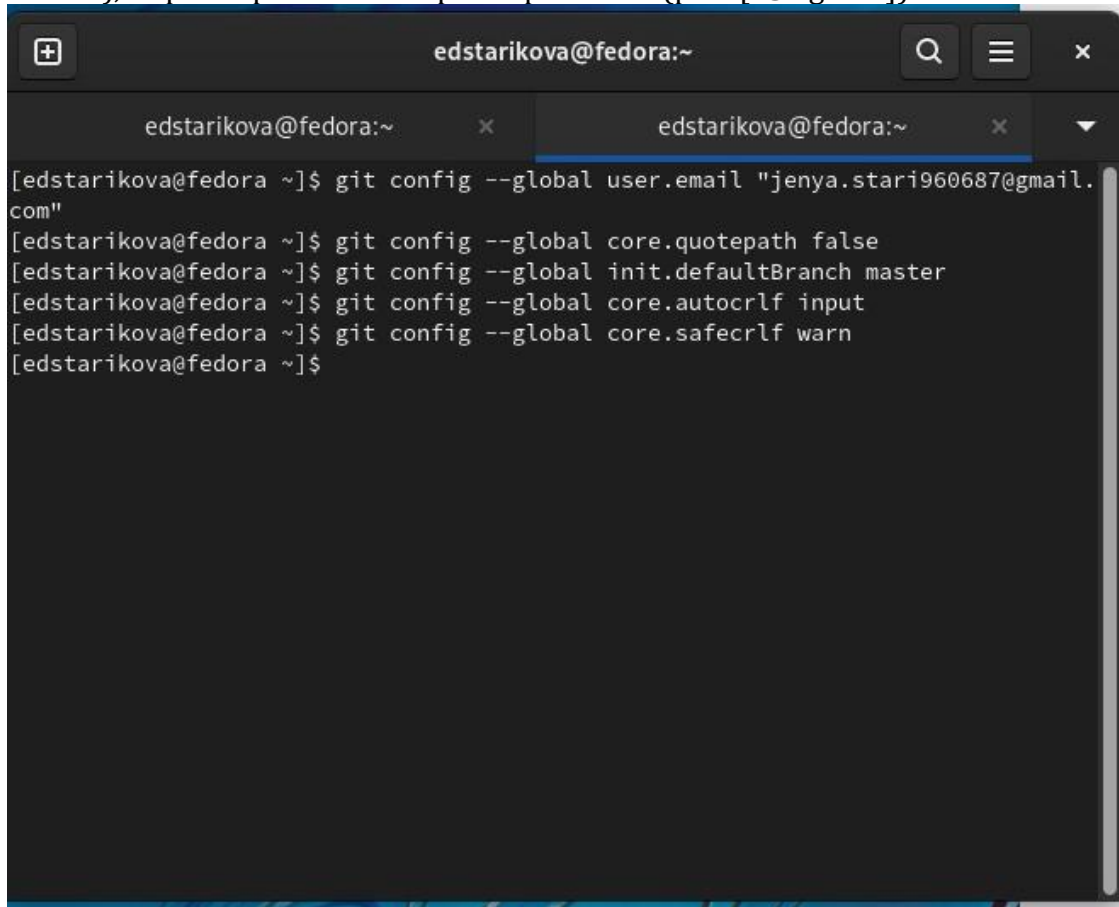
4. Устанавливаю gh в Fedora Linux. Также задаю базовые настройки Git (имя и почта пользователя) (рис. [-@fig:004])



```
edstarikova@fedora:~  
'gitflow/hooks/pre-flow-release-branch' -> '/usr/local/share/doc/gitflow/hooks/p  
re-flow-release-branch'  
'gitflow/hooks/pre-flow-release-delete' -> '/usr/local/share/doc/gitflow/hooks/p  
re-flow-release-delete'  
'gitflow/hooks/pre-flow-release-finish' -> '/usr/local/share/doc/gitflow/hooks/p  
re-flow-release-finish'  
'gitflow/hooks/pre-flow-release-publish' -> '/usr/local/share/doc/gitflow/hooks/  
pre-flow-release-publish'  
'gitflow/hooks/pre-flow-release-start' -> '/usr/local/share/doc/gitflow/hooks/pr  
e-flow-release-start'  
'gitflow/hooks/pre-flow-release-track' -> '/usr/local/share/doc/gitflow/hooks/pr  
e-flow-release-track'  
[edstarikova@fedora ~]$ sudo dnf install gh  
Fedora 35 - x86_64 - Updates                20 kB/s | 19 kB      00:00  
Fedora Modular 35 - x86_64 - Updates        1.1 kB/s | 18 kB     00:16  
Пакет gh-2.7.0-1.fc35.x86_64 уже установлен.  
Зависимости разрешены.  
Отсутствуют действия для выполнения.  
Выполнено!  
[edstarikova@fedora ~]$ git config --global user.name "Evgenia Starikova"  
[edstarikova@fedora ~]$ git config --global user.email "jenya.stari960687@gmail.  
com  
> cd
```

5. Настроим utf-8 в выводе сообщений git, Настройте верификацию и подписание коммитов git. Зададим имя начальной ветки (будем называть её

master), Параметр autocrlf: Параметр safecrlf. (рис. [-@fig:005])

A terminal window titled 'edstarikova@fedora:~' with search, menu, and close buttons. It shows a series of git configuration commands being executed in a shell. The commands are: 'git config --global user.email "jenya.stari960687@gmail.com"', 'git config --global core.quotepath false', 'git config --global init.defaultBranch master', 'git config --global core.autocrlf input', and 'git config --global core.safecrlf warn'. The prompt '[edstarikova@fedora ~]\$' is visible at the start of each line.

```
[edstarikova@fedora ~]$ git config --global user.email "jenya.stari960687@gmail.com"
[edstarikova@fedora ~]$ git config --global core.quotepath false
[edstarikova@fedora ~]$ git config --global init.defaultBranch master
[edstarikova@fedora ~]$ git config --global core.autocrlf input
[edstarikova@fedora ~]$ git config --global core.safecrlf warn
[edstarikova@fedora ~]$
```

6. Создаём ключи ssh – по алгоритму rsa с ключём размером 4096 бит – по алгоритму ed25519 (рис. [-@fig:006]) (рис. [-@fig:006,1])

```
edstarikova@fedora:~  
[edstarikova@fedora ~]$ git config --global core.autocrlf input  
[edstarikova@fedora ~]$ git config --global core.safecrlf warn  
[edstarikova@fedora ~]$ ssh-keygen -t rsa -b 4096  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/edstarikova/.ssh/id_rsa): bit  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in bit  
Your public key has been saved in bit.pub  
The key fingerprint is:  
SHA256:Y7kNjNkgX350ZS0oqgk9KXY0q05tQe6lNrEB0ytQvJI edstarikova@fedora  
The key's randomart image is:  
+---[RSA 4096]---+  
|  oo          .o. |  
| .o+ .      . .o. |  
|.+.oo.....  . |  
|E =B=o=0.o . |  
|.o..+++S . |  
|o . +. * |  
|. . . . |  
|... |  
|.... |  
+---[SHA256]-----+  
[edstarikova@fedora ~]$
```

A terminal window titled 'edstarikova@fedora:~' showing the execution of the 'ssh-keygen -t ed25519' command. The user is prompted to enter a file path (defaulting to '/home/edstarikova/.ssh/id\_ed25519'), a passphrase (left empty), and to confirm the passphrase. The terminal displays the key's fingerprint and a randomart image. The window has a dark theme and standard Linux terminal window controls.

```
[....]
+----[SHA256]-----+
[edstarikova@fedora ~]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/edstarikova/.ssh/id_ed25519): ed25519
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ed25519
Your public key has been saved in ed25519.pub
The key fingerprint is:
SHA256:Pyd7fyuRIZtAGhn9MNyQPhj4biM1SYshDzsbXZKoYgo edstarikova@fedora
The key's randomart image is:
+--[ED25519 256]--+
|      . o . = . +      |
|    + = * B .          |
|   . * B X +           |
|Eo + + O + o .         |
|=  + o S o + o         |
|. . . + . o o          |
|      o . + . .        |
|      = . . .          |
|      .. .oo.          |
+----[SHA256]-----+
[edstarikova@fedora ~]$
```

#fig:006,1 width=70%}

7. Создаём ключи ргр. Генерируем ключ с параметрами: Из предложенных опций выбираем: – тип RSA and RSA; – размер 4096; – выберите срок действия; значение по умолчанию— 0 (срок действия не истекает никогда) (рис. [-@fig:007]) (рис. [-@fig:008]) (рис. [-@fig:009])



```
edstarikova@fedora:~ — gpg --full-generate-key
edstarikova@fedora:~ x edstarikova@fedora:~ — gpg --ful... x
| . o.=.+ |
| + = * B . |
| . * B X + |
| Eo ++ 0 + o . |
| = + o S o + o |
| . . . + . o o |
| o . + . . |
| =.. . |
| .. .oo. |
+-----[SHA256]-----+
[edstarikova@fedora ~]$ gpg --full-generate-key
gpg (GnuPG) 2.3.2; Copyright (C) 2021 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Выберите тип ключа:
(1) RSA and RSA
(2) DSA and Elgamal
(3) DSA (sign only)
(4) RSA (sign only)
(9) ECC (sign and encrypt) *default*
(10) ECC (только для подписи)
(14) Existing key from card
Ваш выбор? 1
```



```
edstarikova@fedora:~ — gpg --full-generate-key
edstarikova@fedora:~$ gpg --full-generate-key
gpg (GnuPG) 2.3.2; Copyright (C) 2021 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

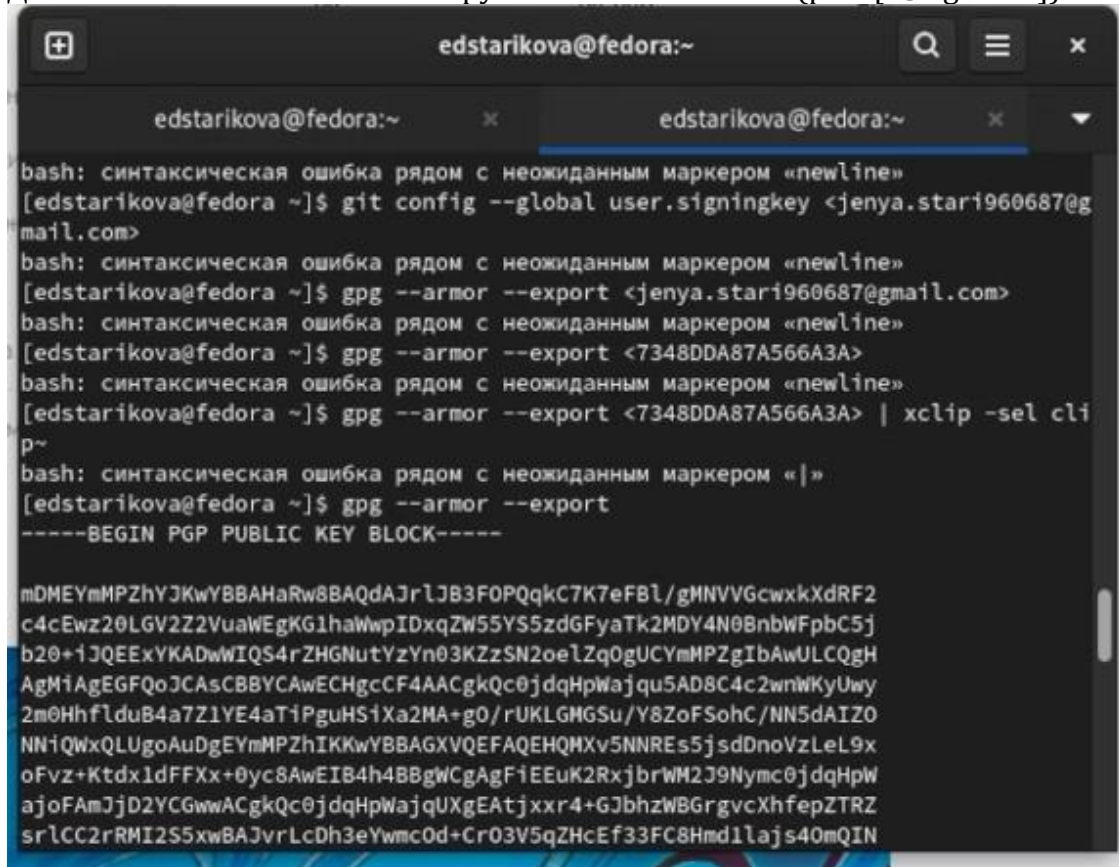
Выберите тип ключа:
  (1) RSA and RSA
  (2) DSA and Elgamal
  (3) DSA (sign only)
  (4) RSA (sign only)
  (9) ECC (sign and encrypt) *default*
 (10) ECC (только для подписи)
 (14) Existing key from card
Ваш выбор? 1
Длина ключей RSA может быть от 1024 до 4096.
Какой размер ключа Вам необходим? (3072) 4096
Упрощенный размер ключа - 4096 бит
Выберите срок действия ключа.
  0 = не ограничен
  <n> = срок действия ключа - n дней
  <n>w = срок действия ключа - n недель
  <n>m = срок действия ключа - n месяцев
  <n>y = срок действия ключа - n лет
Срок действия ключа? (0) 0
```

```
edstarikova@fedora:~ — gpg --full-generate-key
(3) DSA (sign only)
(4) RSA (sign only)
(9) ECC (sign and encrypt) *default*
(10) ECC (только для подписи)
(14) Existing key from card
Ваш выбор? 1
длина ключей RSA может быть от 1024 до 4096.
Какой размер ключа Вам необходим? (3072) 4096
Запрошенный размер ключа - 4096 бит
Выберите срок действия ключа.
    0 = не ограничен
    <n> = срок действия ключа - n дней
    <n>w = срок действия ключа - n недель
    <n>m = срок действия ключа - n месяцев
    <n>y = срок действия ключа - n лет
Срок действия ключа? (0) 0
Срок действия ключа не ограничен
Все верно? (y/N) y

GnuPG должен составить идентификатор пользователя для идентификации ключа.

Ваше полное имя: Evgenia
Адрес электронной почты: jenya.stari960687@gmail.com
Примечание: 
```

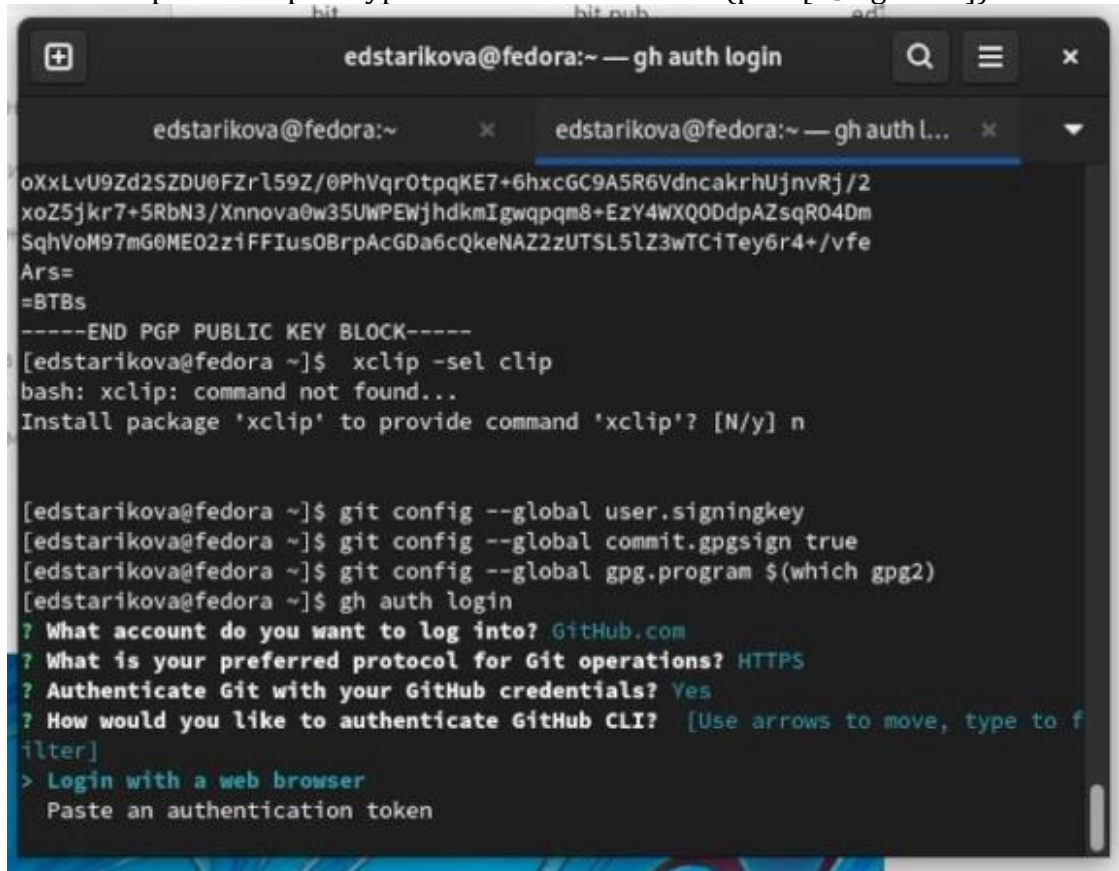
8. Добавляем ключ в Github. Копируем отпечаток ключа (рис. [-@fig:0010])



The image shows a terminal window titled 'edstarikova@fedora:~'. The user is attempting to generate a GPG key. The commands and their outputs are as follows:

```
edstarikova@fedora:~  
bash: синтаксическая ошибка рядом с неожиданным маркером «newline»  
[edstarikova@fedora ~]$ git config --global user.signingkey <jenya.stari960687@gmail.com>  
bash: синтаксическая ошибка рядом с неожиданным маркером «newline»  
[edstarikova@fedora ~]$ gpg --armor --export <jenya.stari960687@gmail.com>  
bash: синтаксическая ошибка рядом с неожиданным маркером «newline»  
[edstarikova@fedora ~]$ gpg --armor --export <7348DDA87A566A3A>  
bash: синтаксическая ошибка рядом с неожиданным маркером «newline»  
[edstarikova@fedora ~]$ gpg --armor --export <7348DDA87A566A3A> | xclip -sel cli  
p~  
bash: синтаксическая ошибка рядом с неожиданным маркером «|»  
[edstarikova@fedora ~]$ gpg --armor --export  
-----BEGIN PGP PUBLIC KEY BLOCK-----  
  
mDMEYmMPZhYJKwYBBAHAw8BAQdAJrLJB3FOPQqkC7K7eFBl/gMNVVGcwXkXDRF2  
c4cEwz20LGV2Z2VuaWEgKG1haWwIDxqZW55YS5zdGFyaTk2MDY4N08nbWFpbC5j  
b20+1JQEExYKADwIQS4rZHGNutYzYn03KZzSN2oeLZq0gUCYmMPZgIbAwULCQgH  
AgMiAgEGFQoJCAwCBBYCAwECHgcCF4AACgkQc0jddqHpWajqu5AD8C4c2wnWkyUwy  
2m0HhflduB4a7Z1YE4aTiPguHSiXa2MA+g0/rUKLGMGSu/Y8ZoFSohC/NN5dAIZO  
NNiQWxQLUgoAuDgEYmMPZhIKKwYBBAgXVQEFAQEhQMxv5NNREs5jsdDnoVzLeL9x  
oFvz+KtdxldFFXx+0yc8AwEIB4h4BBgWcAgFiEEuK2RxjbrWM2J9Nymc0jddqHpW  
ajoFAMJjD2YCGwwACgkQc0jddqHpWajquXgEAtjxxr4+GJbhzWBGrGvcXhfepZTRZ  
srLCC2rRMI2S5xwBAJvrLcDh3eYwmcOd+CrO3V5qZHcEf33FC8Hmd1lajs40mQIN
```

9. Сознание репозитория курса на основе шаблона. (рис. [-@fig:0011])



```
edstarikova@fedora:~ — gh auth login
oXxLvU9Zd2SZDU0FZr159Z/0PhVqr0tpqKE7+6hxcGC9A5R6VdncakrhUjnvRj/2
xoZ5jkr7+5RbN3/Xnnova0w35UWPEWjhdkmIgwqqm8+EzY4WXQ0DdpAZsqR04Dm
SqhVoM97mG0ME02ziFFIus0BrpAcGDacQkeNAZ2zUTSL5lZ3wTCiTey6r4+/vfe
Ars=
=BTBs
-----END PGP PUBLIC KEY BLOCK-----
[edstarikova@fedora ~]$ xclip -sel clip
bash: xclip: command not found...
Install package 'xclip' to provide command 'xclip'? [N/y] n

[edstarikova@fedora ~]$ git config --global user.signingkey
[edstarikova@fedora ~]$ git config --global commit.gpgsign true
[edstarikova@fedora ~]$ git config --global gpg.program $(which gpg2)
[edstarikova@fedora ~]$ 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? [Use arrows to move, type to filter]
> Login with a web browser
  Paste an authentication token
```

## Выводы

Я изучил идеологию и применение контроля версий.