

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ «РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ»**

Факультет физико-математических и естественных наук Кафедра информационных технологий

### **ОТЧЕТ**

по лабораторной работе 2

**ТЕМА** «Управление версиями» по дисциплине» по дисциплине «Операционные системы»

**Выполнил/ла:** \*\*Студент/ка группы\*\* НПИбд-02-21

**Студенческий билет №** 1032205421

**Студент/ка:** Стелина Петрити

## **Цель работы**

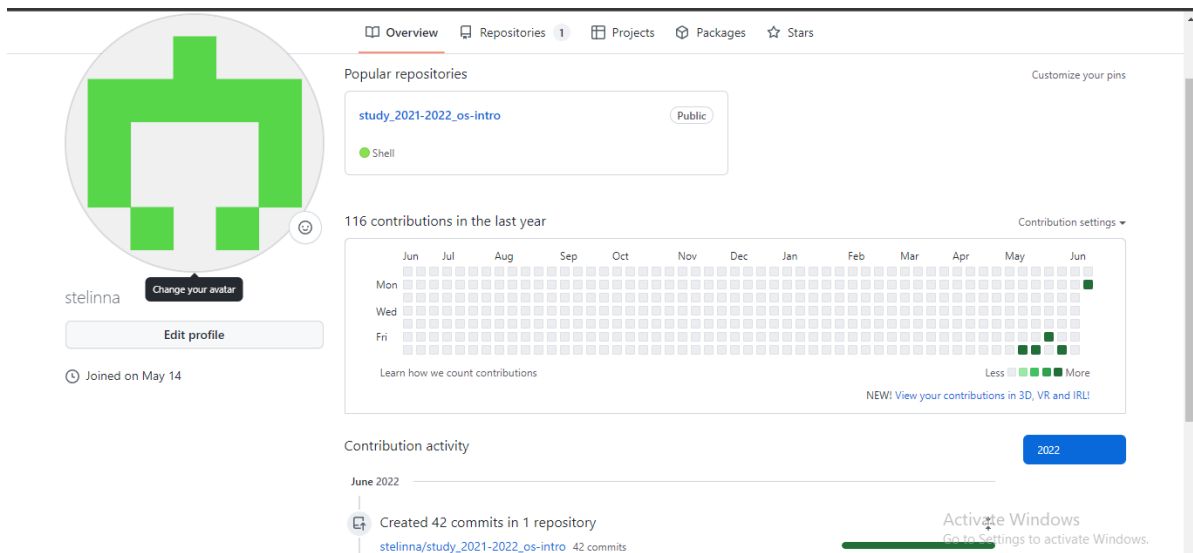
Изучить идеологию и применение средств контроля версий. Освоить умения по работе с git.

## **Последовательность выполнения работы**

### **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\*

1. Создадим учётную запись на <https://github.com>.



### 1.1. Создать учетную запись

## 2. Установите программное обеспечение git-flow через терминал

```
[inna@fedora ~]$ cd /tmp
[inna@fedora tmp]$ wget --no-check-certificate -q https://raw.githubusercontent.com/petervanderdoes/gitflow/develop/contrib/gitflow-installer.sh
[inna@fedora tmp]$ chmod +x gitflow-installer.sh
[inna@fedora tmp]$ sudo ./gitflow-installer.sh install stable
[sudo] password for inna:
### git-flow no-make installer ###
Installing git-flow to /usr/local/bin
Cloning repo from GitHub to gitflow
Cloning into 'gitflow'...
remote: Enumerating objects: 4270, done.
remote: Total 4270 (delta 0), reused 0 (delta 0), pack-reused 4270
Receiving objects: 100% (4270/4270), 1.74 MiB | 99.00 KiB/s, done.
Resolving deltas: 100% (2533/2533), done.
Already up to date.
Branch 'master' set up to track remote branch 'master' from 'origin'.
Switched to a new branch 'master'
install: creating directory '/usr/local/share/doc'
install: creating directory '/usr/local/share/doc/gitflow'
install: creating directory '/usr/local/share/doc/gitflow/hooks'
'gitflow/git-flow' -> '/usr/local/bin/git-flow'
'gitflow/git-flow-init' -> '/usr/local/bin/git-flow-init'
'gitflow/git-flow-feature' -> '/usr/local/bin/git-flow-feature'
'gitflow/git-flow-bugfix' -> '/usr/local/bin/git-flow-bugfix'
```

### 2.2. программное обеспечение git-flow через терминал

## 3. Установите программное обеспечение gh через терминал

```
[inna@fedora ~]$ sudo dnf install gh
[sudo] password for inna:
Fedora Modular 35 - x86_64 - Updates          1.3 kB/s | 19 kB      00:14
Last metadata expiration check: 0:00:01 ago on Sun 12 Jun 2022 08:27:13 PM CEST.
Package gh-2.11.3-2.fc35.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[inna@fedora ~]$
```

### 3.3. программное обеспечение gh через терминал

4. Давайте сделаем базовую настройку git

1. Укажите имя и адрес электронной почты владельца репозитория.
2. Настройте utf-8 при выводе сообщений git.
3. Настройте проверку и подписание коммитов git.
4. Давайте зададим имя начальной ветви (назовем ее master).
5. Установите параметры autocrlf и safecrlf.

```
[inna@fedora ~]$ git config --global user.name "Stelina Petriti"
[inna@fedora ~]$ git config --global user.email "stelina.petriti@gmail.com"
[inna@fedora ~]$
[inna@fedora ~]$ git config --global core.quotepath false
[inna@fedora ~]$ git config --global init.defaultBranch master
[inna@fedora ~]$ git config --global core.autocrlf input
[inna@fedora ~]$
[inna@fedora ~]$ git config --global core.safecrlf warn
[inna@fedora ~]$
```

fig(1,2,3,4,5)

### 5.5. Создайте SSH-ключ

1. в соответствии с алгоритмом rsa с размером ключа 4096 бит;

```

[inna@fedora ~]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/inna/.ssh/id_rsa):
Created directory '/home/inna/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/inna/.ssh/id_rsa
Your public key has been saved in /home/inna/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:1b+RdkiDfM10drdoeCzdUD93N+BSI7AG1xXu8y5S
The key's randomart image is:
+---[RSA 4096]-----+
|      . oo..Bo.*|
|      o .+0.*=*|
|      o.++0+=B|
|      ..  *+  +*|
|      S   o* .|
|      +o+ |
|      + .. |
|      o .. |
|      .E. . . |
+-----[SHA256]-----+
[inna@fedora ~]$

```

1.1.rsa с размером ключа 4096 бит;

2. в соответствии с алгоритмом ed25519.

```

[inna@fedora ~]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/inna/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/inna/.ssh/id_ed25519
Your public key has been saved in /home/inna/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:g3X6Tu4sXSj0MILHp5neiRYtej9swcFYWM6N
The key's randomart image is:
+--[ED25519 256]--+
|      ==      |
|      +o.=     |
|      + == o   |
|      . *oOo   |
|      ..&S= .   |
|      E *oo .   |
|      + * =o.   |
|      . = B=.   |
|      o o.+    |
+-----[SHA256]-----+
[inna@fedora ~]$

```

2.1. с алгоритмом ed25519

6. Создайте ключ GPG

– Выберите параметры, описанные в лаборатории:

– введите RSA и RSA; размер 4096; — выберите дату истечения срока действия; значение по умолчанию – 0 - GPG запросит личную информацию, которая будет сохранена в ключе. – Имя. – Адрес электронной почты.

```
[inna@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 it and to redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: directory '/home/inna/.gnupg' created
gpg: keybox '/home/inna/.gnupg/pubring.kbx' created
Please select what kind of key you want:
  (1) RSA and RSA
  (2) DSA and Elgamal
  (3) DSA (sign only)
  (4) RSA (sign only)
  (9) ECC (sign and encrypt) *default*
 (10) ECC (sign only)
 (14) Existing key from card
Your selection? 1
RSA keys may be between 1024 and 4096 bits long;
What keysize do you want? (3072) 4096
Requested keysize is 4096 bits
Please specify how long the key should be valid
    0 = key does not expire
    <n> = key expires in n days
    <n>w = key expires in n weeks
    <n>m = key expires in n months
    <n>y = key expires in n years
Key is valid for? (0) 0
Key does not expire at all
Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify
```

*6.1.RSA и RSA/ размер 4096/ дата истечения срока действия*

```

Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify
your key.

Real name: Stelina
Email address: stelinaPETRITI@gmail.com
Comment:
You selected this USER-ID:
    "Stelina <stelinaPETRITI@gmail.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit.
We need to generate a lot of random bytes. It is
recommended that you perform some other action (type on the keyboard, move the
mouse, use the disks) during the prime generation; this gives
the random number generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is
recommended that you perform some other action (type on the keyboard, move the
mouse, use the disks) during the prime generation; this gives
the random number generator a better chance to gain enough entropy.
gpg: /home/inna/.gnupg/trustdb.gpg: trustdb created
gpg: key 3AD1E7E531B42C3F marked as ultimately trusted
gpg: directory '/home/inna/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/home/inna/.gnupg/openpgp-revocs.d/3AD1E7E531B42C3F.1'
public and secret key created and signed.

pub   rsa4096 2022-06-12 [SC]
       0A56EDCEFC360FF36F251EED3AD1E7E531B42C3F
uid           Stelina <stelinaPETRITI@gmail.com>
sub   rsa4096 2022-06-12 [E]

```

## 6.2.GPG

7. Выведите список ключей и скопируйте отпечаток закрытого ключа

```

[inna@fedora ~]$ gpg --list-secret-keys --keyid-format LONG
gpg: checking the trustdb
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u
/home/inna/.gnupg/pubring.kbx
-----
sec   rsa4096/3AD1E7E531B42C3F 2022-06-12 [SC]
       0A56EDCEFC360FF36F251EED3AD1E7E531B42C3F
uid           [ultimate] Stelina <stelinaPETRITI@gmail.com>
ssb   rsa4096/80A9BA7DF5453356 2022-06-12 [E]

```

7.1. список ключей/копия отпечаток закрытого ключа

8.Настройте gh. Сначала вам нужно войти в систему, вы можете воспользоваться браузером

```
[inna@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? Login with a web browser

! First copy your one-time code: 5EB4-6FD0
Press Enter to open github.com in your browser...
```

8.1.gh

9.Создание репозитория и необходимых файлов, которые были необходимы в этой лаборатории

master ▾ 1 branch 0 tags			Go to file	Add file ▾	Code ▾
stelinna Delete presentation 932Fc3a 9 days ago 96 commits					
📁 Labs	Delete presentation	9 days ago			
📁 config	Initial commit	last month			
📁 template	Initial commit	last month			
📄 .gitattributes	Initial commit	last month			
📄 .gitignore	Initial commit	last month			
📄 .gitmodules	Initial commit	last month			
📄 LICENSE	Initial commit	last month			
📄 Makefile	Initial commit	last month			
📄 README.en.md	Initial commit	last month			
📄 README.git-flow.md	Initial commit	last month			
📄 README.md	Initial commit	last month			

## Выводы

В этой лаборатории мы научились изучать идеологию и применение инструментов контроля версий. Также мы освоили навыки работы с git

## Контрольные вопросы

**1.Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются?**

это программные инструменты, помогающие командам разработчиков управлять изменениями в исходном коде с течением времени

**2.Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.**

система контроля версий (VCS) — это система, регистрирующая изменения в одном или нескольких файлах

Хранилище-система,обеспечивает хранение всех существовавших вариантов файлов

Commit-фиксация изменений. Команде commit можно передать сообщение, описывающее изменения в ревизии.

История-список предыдущих ревизий.

Рабочая копия - копия другой ветки.

### **3. Что представляют собой и чем отличаются централизованные и децентрализованные**

Существуют СКВ централизованные- имеется один репозиторий, в который собираются изменения со всех рабочих копий разработчиков, и децентрализованные, когда репозитория много, и они могут обмениваться изменениями между собой. централизованные СКВ - Репозиторий один.

### **4.Опишите действия с VCS при единоличной работе с хранилищем.**

Создание хранилища (git init).

Push файлов (git push).

Настройка хранилища.

Коммиты нужных файлов (git add , git commit).

### **5.Опишите порядок работы с общим хранилищем VCS.**

Push файлов (git push).

Создание хранилища (git init).

Добавление в коммит различных файлов (git add , git commit), коммит ресурсов.

### **6. Каковы основные задачи, решаемые инструментальным средством git?**

предоставление возможности отката в случае неработоспособности кода

хранение всех версий продукта.

### **7.Назовите и дайте краткую характеристику командам git**

git clean- удаления мусора из рабочего каталога. Это результаты сборки проекта или файлы конфликтов слияний.

git rm - используется в Git для удаления файлов из индекса и рабочей копии.

git mv — это всего лишь удобный способ переместить файл.

git reset - используется в основном для отмены изменений.

git commit - берёт все данные, добавленные в индекс с помощью git add, и сохраняет их слепок во внутренней базе данных

git add - добавляет содержимое рабочего каталога в индекс (staging area) для последующего коммита.

### **8.Приведите примеры использования при работе с локальным и удалённым репозиториями.**

Локальные репозитории - разработчик один. Удаленные - когда работает целая команда.

### **9.Что такое и зачем могут быть нужны ветви (branches)?**



ветви - чтобы разделять код. Если мы делаем новый функционал, то мы создаем новую ветку, а после окончания мы сделали в основную ветку.

#### **10. Как и зачем можно игнорировать некоторые файлы при commit?**

Игнорируемый файл — файл, явным образом помеченный для Git как файл, который необходимо игнорировать. Игнорируемые файлы отслеживаются в специальном файле `.gitignore`, который регистрируется в корневом каталоге репозитория.