

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук Кафедра фундаментальной информатики и информационных технологий

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1 дисциплина: Операционные системы

Студент: Мальков Роман Сергеевич

Группа: НФИбд-02-21

МОСКВА 2022 г.

Цель работы

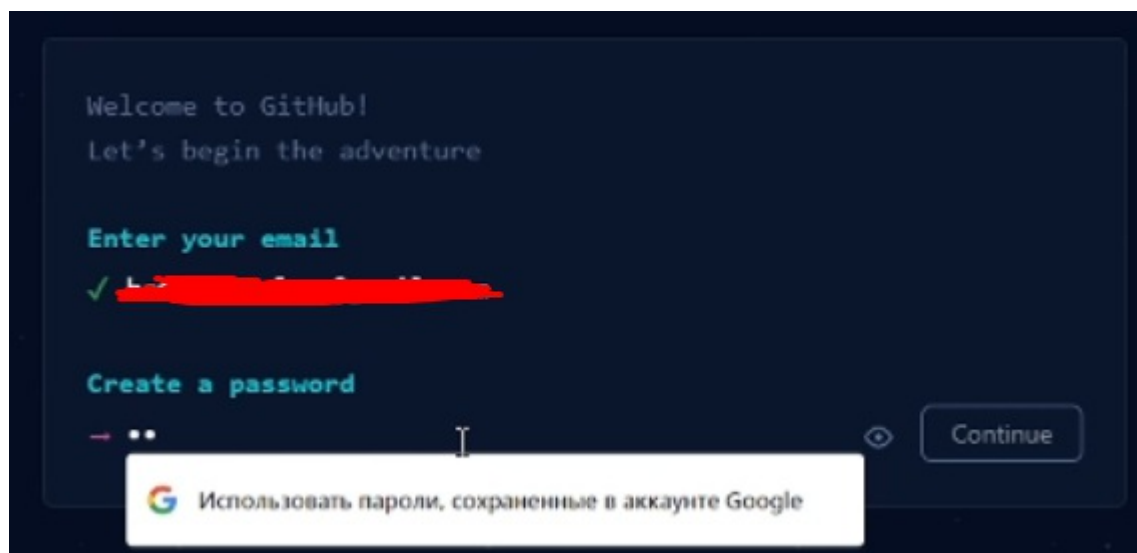
- Изучить идеологию и применение средств контроля версий.
- Освоить умения по работе с git.

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.

Выполнение лабораторной работы

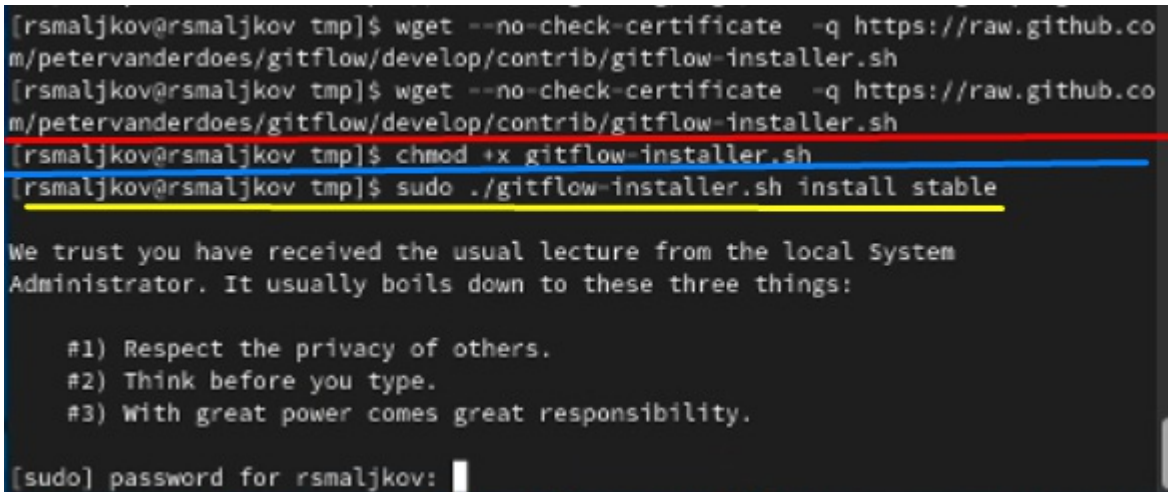
Создаем учётную запись на <https://github.com> и заполняем нужные данные(Скриншот 1).



(Скриншот 1)

Устанавливаем git-flow и gh на Fedora Linux(Скриншот 2-3), используя следующие команды.

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



```
[rsmaljkov@rsmaljkov tmp]$ wget --no-check-certificate -q https://raw.githubusercontent.com/petervanderdoes/gitflow/develop/contrib/gitflow-installer.sh
[rsmaljkov@rsmaljkov tmp]$ wget --no-check-certificate -q https://raw.githubusercontent.com/petervanderdoes/gitflow/develop/contrib/gitflow-installer.sh
[rsmaljkov@rsmaljkov tmp]$ chmod +x gitflow-installer.sh
[rsmaljkov@rsmaljkov tmp]$ sudo ./gitflow-installer.sh install stable

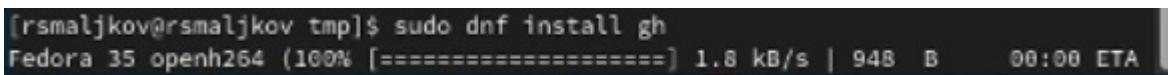
We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.

[sudo] password for rsmaljkov: 
```

(Скриншот

2)



```
[rsmaljkov@rsmaljkov tmp]$ sudo dnf install gh
Fedora 35 openh264 (100% [=====]) 1.8 kB/s | 948 B 00:00 ETA
```

(Скриншот

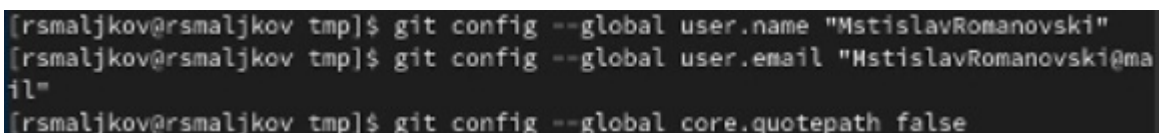
3)

Производим базовую настройку git, задаем имя и электронную почту владельца репозитория(Скриншот 4):

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

Настраиваем utf-8 в выводе сообщений git(Скриншот 4):

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



```
[rsmaljkov@rsmaljkov tmp]$ git config --global user.name "MstislavRomanovski"
[rsmaljkov@rsmaljkov tmp]$ git config --global user.email "MstislavRomanovski@mail"
[rsmaljkov@rsmaljkov tmp]$ git config --global core.quotePath false
```

(Скриншот

4) Настраиваем верификацию и подписание git, задаем имя начальной ветки, задаем параметр autocrlf и safecrlf(Скриншот 5):

```
git config --global init.defaultBranch master
git config --global core.autocrlf input
git config --global core.safecrlf warn
```

```
[rsmaljkov@rsmaljkov tmp]$ git config --global init.defaultBranch master
[rsmaljkov@rsmaljkov tmp]$ git config --global core.autocrlf input
[rsmaljkov@rsmaljkov tmp]$ git config --global core.safecrlf warn
```

(Скриншот

5)

Создаем ключи ssh по алгоритму rsa размером 4096, и по алгоритму ed25519(Скриншот 6-7):

```
ssh-keygen -t rsa -b 4096
ssh-keygen -t ed25519
```

```
[rsmaljkov@rsmaljkov tmp]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/rsmaljkov/.ssh/id_rsa):
Created directory '/home/rsmaljkov/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/rsmaljkov/.ssh/id_rsa
Your public key has been saved in /home/rsmaljkov/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:0KNE/5QvG1bgKGpm5W+7k5p3FmRfgI3FW9G6vYDEvY rsmaljkov@rsmaljkov
The key's randomart image is:
+----[RSA 4096]-----+
|      .  . * . . . . |
|      . 0 000+ . + |
|      = = +0... + |
|      = + ++00..0 |
|      = 0 50=0.E0 |
|      + . ..+0... |
|      0... .. |
|      0+.0 . |
|      000= |
+----[SHA256]-----+
[rsmaljkov@rsmaljkov tmp]$
```

(Скриншот

```
[rsmaljkov@rsmaljkov tmp]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/rsmaljkov/.ssh/id_ed25519):
```

6)

(Скриншот 7)

Создаем ключ pgp(Скриншот 8):

```
gpg --full-generate-key
```

- Из предложенных опций выбираем:
- тип RSA and RSA;
- размер 4096;
- выберите срок действия; значение по умолчанию— 0 (срок действия не истекает никогда).
- GPG запросит личную информацию, которая сохранится в ключе:
- Имя (не менее 5 символов).
- Адрес электронной почты.
- При вводе email убедитесь, что он соответствует адресу, используемому на GitHub.
- Комментарий. Можно ввести что угодно или нажать клавишу ввода, чтобы оставить это поле пустым.

```
[rsmaljkov@rsmaljkov tmp]$ 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.

gpg: directory '/home/rsmaljkov/.gnupg' created
gpg: keybox '/home/rsmaljkov/.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?
```

(Скриншот

8)

Далее копируем отпечаток приватного ключа, он выводился на экран консоли, при генерации(Скриншот 9).

```
sec  rsa4096/CA665AC57D7CE375 2022-04-23 [SC]
      C83458C75ACF1F15EE4696F0CA665AC57D7CE375
```

(Скриншот 9)

После этого, используем команду указанную ниже, чтобы получить ключ, и ввести его на <https://github.com/settings/keys> (Скриншот 10):

```
[rsmaljkov@rsmaljkov tmp]$ gpg --armor --export CA665AC57D7CE375
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGJj7ykBEADQJ5Q5LSdxnLUpH8K9aqxMo9ltPopA58dNLAS28jiku9lRBrik
REKAU+SBG9F4EQx9oPLnlj0zlkKtNc3HctTc5daja15gdG6/r9UAQD1xCj95s0HY
6wzW8DA1Ds7uww47pPDhG93Xpgwh0rEgSjvbE2NJJbgahyd7vmyNHX3JaEfVxUK+
dqe/IsBk3DpsJYdub6RYiaRLU3tEvFVDWlLv+bYABxwEsWAXfnQK+/MWVfCHKxp
Y0AhkeSaY57lJurVxAfWvYR1ez2NoXHyXCi5gGV0dAqDW+GKW3p7f8h8UmKYbXg
RNz/5VsUTiDxUz043ItCLbWW0z92LA4p6jwuxafaI6rjSVAuvsF/8g9Nj6hlp2/I
0JVLh/AuEt6ZCqqgSrNvBcTiNQkeFgvZqIGN0SsUpnyE0wi5kR6Dx77INJq5LFNP
/dmC9lKCs0JMIpjbvBC8Wetg48HGHI1dljsjvnFhkvZbGYoiZNzWDMvdjM/CdoyA2
xPb2gz2gSe15IPAHhQ0q5QpS5/I7j70Qzz5hhL80AV0yK7GJs6qdDJ+ISbQulSWp
```

(Скриншот

10)

```
gpg --armor --export <Ваш отпечаток приватного ключа>
```

Настраиваем автоматические подписи git(Скриншот 11):

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

```
[rsmaljkov@rsmaljkov tmp]$ git config --global user.signingkey CA665AC57D7CE375
[rsmaljkov@rsmaljkov tmp]$ git config --global commit.gpgsign true
[rsmaljkov@rsmaljkov tmp]$ git config --global gpg.program $(which gpg2)
```

(Скриншот

11)

Настройка gh(Скриншот 12):

```
gh auth login
```

```
[rsmaljkov@rsmaljkov tmp]$ gh auth login
? What account do you want to log into? [Use arrows to move, type to filter]
> GitHub.com
  GitHub Enterprise Server
```

(Скриншот 12)

Создаем репозиторий курса на основе шаблона(Скриншот 13-15):

```
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=yamadharma/course-directory-student-template --public
4 git clone --recursive
  ↪ https://github.com/MstislavRomanovski/study_2021-2022_os-intro.git
```

```
[rsmaljkov@rsmaljkov tmp]$ mkdir -p ~/work/study/2021-2022/"Operating Systems"
[rsmaljkov@rsmaljkov tmp]$ cd ~/work/study/2021-2022/"Operating Systems"
```

(Скриншот 13)

```
[rsmaljkov@rsmaljkov Operating Systems]$ gh repo create study_2021-2022_os-intro --templ
ate=yamadharma/course-directory-student-template --public
✓ Created repository MstislavRomanovski/study_2021-2022_os-intro on GitHub
```

(Скриншот 14)

```
[root@rsmaljkov Operating Systems]# git clone --recursive https://github.com/MstislavRom
anovski/study_2021-2022_os-intro.git
```

(Скриншот 15)

Настраиваем каталог курса, а имеено, переходим в каталог курса, удаляем лишние файлы(у меня не удалились), создаем необходимые каталоги(пришлось получить рут права для этого), и отправляем файлы на сервер(Скриншоты 16-20):

```
cd ~/work/study/2021-2022/"Operating Systems"/study_2021-2022_os-intro
rm package.json
make COURSE=os-intro
git add .
git commit -am 'feat(main): make course structure'
git push
```

```
[rsmaljkov@rsmaljkov work]$ cd ~/work/study/2021-2022/"Operating Systems"/study_
2021-2022_os-intro
[rsmaljkov@rsmaljkov study_2021-2022_os-intro]$ ls
config  Makefile      README.en.md    README.md
LICENSE package.json  README.git-flow.md  template
[rsmaljkov@rsmaljkov study_2021-2022_os-intro]$ rm package.json
```

```
[root@rsmaljkov study_2021-2022_os-intro]# rm package.json
```

```
[root@rsmaljkov study_2021-2022_os-intro]# make COURSE=os-intro
```

```
[root@rsmaljkov study_2021-2022_os-intro]# git add .  
[root@rsmaljkov study_2021-2022_os-intro]# git commit -am 'feat(main):make course structure'  
[root@rsmaljkov study_2021-2022_os-intro]# git push
```

(Скриншоты 16 - 20)

Выводы

Мы изучили идеологию по контролю версий, и научились работать с git.