

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ»**

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

ОТЧЕТ
по лабораторной работе №2
по дисциплине «Операционные системы»

Выполнила:
Студентка группы
НПИбд-01–21
Студенческий билет № 1032216538
Абдуллина Ляйсан Раисовна
« 22 » апреля 2022г.

Москва 2022

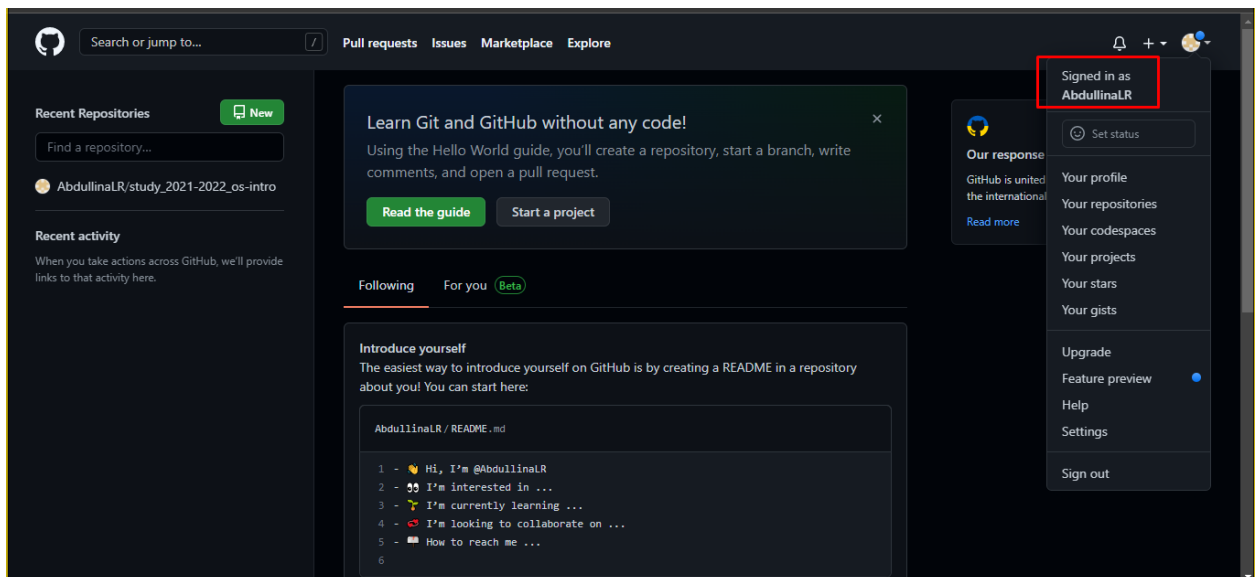
Цель работы:

Изучение идеологии и применения средств контроля версий, а также освоение умения по работе с git.

Ход работы:

[1] Настройка GitHub

Создадим учётную запись на <https://github.com> и заполним основные данные. (скриншот 1.1)



1.1 Успешная регистрация на GitHub'е с использованием своего имени, согласно Соглашению об именовании.

[2] Установка программного обеспечения

Поскольку программное обеспечение удалено из репозитория, мы должны установить его вручную, используя следующие команды(скриншоты 2.1–2.3):

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

```
[lrabdullina@fedora tmp]$ wget --no-check-certificate -q https://raw.githubusercontent.com/abdullinalr/gitflow/develop/contrib/gitflow-installer.sh
[lrabdullina@fedora tmp]$ chmod +x gitflow-installer.sh
chmod: невозможно получить доступ к 'gitflow-installer.sh': Нет такого файла или каталога
[lrabdullina@fedora tmp]$ wget --no-check-certificate -q https://raw.githubusercontent.com/petervanderdoes/gitflow/develop/contrib/gitflow-installer.sh
[lrabdullina@fedora tmp]$ chmod +x gitflow-installer.sh
[lrabdullina@fedora tmp]$ sudo ./gitflow-installer.sh install stable
```

Мы полагаем, что ваш системный администратор изложил вам основы безопасности. Как правило, всё сводится к трём следующим правилам:

- №1) Уважайте частную жизнь других.
- №2) Думайте, прежде что-то вводить.
- №3) С большой властью приходит большая ответственность.

```
[sudo] пароль для lrabdullina:
### 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 | 475.00 КиБ/с, готово.
Определение изменений: 100% (2533/2533), готово.
Уже обновлено.
Ветка «master» отслеживает внешнюю ветку «master» из «origin».
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'
'gitflow/git-flow-init' -> '/usr/local/bin/git-flow-init'
'gitflow/git-flow-feature' -> '/usr/local/bin/git-flow-feature'
```

2.1 Ввод выше представленных команд в консоль. Введение пароля для пользователя. Установка gitflow.

```
install: создание каталога '/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'
'gitflow/git-flow-hotfix' -> '/usr/local/bin/git-flow-hotfix'
'gitflow/git-flow-release' -> '/usr/local/bin/git-flow-release'
'gitflow/git-flow-support' -> '/usr/local/bin/git-flow-support'
'gitflow/git-flow-version' -> '/usr/local/bin/git-flow-version'
'gitflow/gitflow-common' -> '/usr/local/bin/gitflow-common'
'gitflow/gitflow-shFlags' -> '/usr/local/bin/gitflow-shFlags'
'gitflow/git-flow-config' -> '/usr/local/bin/git-flow-config'
'gitflow/hooks/filter-flow-hotfix-finish-tag-message' -> '/usr/local/share/doc/gitflow/hooks/filter-flow-hotfix-finish-tag-message'
'gitflow/hooks/filter-flow-hotfix-start-version' -> '/usr/local/share/doc/gitflow/hooks/filter-flow-hotfix-start-version'
'gitflow/hooks/filter-flow-release-branch-tag-message' -> '/usr/local/share/doc/gitflow/hooks/filter-flow-release-branch-tag-message'
'gitflow/hooks/filter-flow-release-finish-tag-message' -> '/usr/local/share/doc/gitflow/hooks/filter-flow-release-finish-tag-message'
'gitflow/hooks/filter-flow-release-start-version' -> '/usr/local/share/doc/gitflow/hooks/filter-flow-release-start-version'
'gitflow/hooks/post-flow-bugfix-delete' -> '/usr/local/share/doc/gitflow/hooks/post-flow-bugfix-delete'
'gitflow/hooks/post-flow-bugfix-finish' -> '/usr/local/share/doc/gitflow/hooks/post-flow-bugfix-finish'
'gitflow/hooks/post-flow-bugfix-publish' -> '/usr/local/share/doc/gitflow/hooks/post-flow-bugfix-publish'
'gitflow/hooks/post-flow-bugfix-pull' -> '/usr/local/share/doc/gitflow/hooks/post-flow-bugfix-pull'
'gitflow/hooks/post-flow-bugfix-start' -> '/usr/local/share/doc/gitflow/hooks/post-flow-bugfix-start'
'gitflow/hooks/post-flow-bugfix-track' -> '/usr/local/share/doc/gitflow/hooks/post-flow-bugfix-track'
'gitflow/hooks/post-flow-feature-delete' -> '/usr/local/share/doc/gitflow/hooks/post-flow-feature-delete'
'gitflow/hooks/post-flow-feature-finish' -> '/usr/local/share/doc/gitflow/hooks/post-flow-feature-finish'
'gitflow/hooks/post-flow-feature-publish' -> '/usr/local/share/doc/gitflow/hooks/post-flow-feature-publish'
'gitflow/hooks/post-flow-feature-pull' -> '/usr/local/share/doc/gitflow/hooks/post-flow-feature-pull'
'gitflow/hooks/post-flow-feature-start' -> '/usr/local/share/doc/gitflow/hooks/post-flow-feature-start'
'gitflow/hooks/post-flow-feature-track' -> '/usr/local/share/doc/gitflow/hooks/post-flow-feature-track'
'gitflow/hooks/post-flow-hotfix-delete' -> '/usr/local/share/doc/gitflow/hooks/post-flow-hotfix-delete'
'gitflow/hooks/post-flow-hotfix-finish' -> '/usr/local/share/doc/gitflow/hooks/post-flow-hotfix-finish'
'gitflow/hooks/post-flow-hotfix-publish' -> '/usr/local/share/doc/gitflow/hooks/post-flow-hotfix-publish'
```

2.2 Продолжение установки GitFlow и создание каталога /usr/local/share/doc/gitflow/hooks

```
'gitflow/hooks/post-flow-feature-track' -> '/usr/local/share/doc/gitflow/hooks/post-flow-feature-track'
'gitflow/hooks/post-flow-hotfix-delete' -> '/usr/local/share/doc/gitflow/hooks/post-flow-hotfix-delete'
'gitflow/hooks/post-flow-hotfix-finish' -> '/usr/local/share/doc/gitflow/hooks/post-flow-hotfix-finish'
'gitflow/hooks/post-flow-hotfix-publish' -> '/usr/local/share/doc/gitflow/hooks/post-flow-hotfix-publish'
'gitflow/hooks/post-flow-hotfix-start' -> '/usr/local/share/doc/gitflow/hooks/post-flow-hotfix-start'
'gitflow/hooks/post-flow-release-branch' -> '/usr/local/share/doc/gitflow/hooks/post-flow-release-branch'
'gitflow/hooks/post-flow-release-delete' -> '/usr/local/share/doc/gitflow/hooks/post-flow-release-delete'
'gitflow/hooks/post-flow-release-finish' -> '/usr/local/share/doc/gitflow/hooks/post-flow-release-finish'
'gitflow/hooks/post-flow-release-publish' -> '/usr/local/share/doc/gitflow/hooks/post-flow-release-publish'
'gitflow/hooks/post-flow-release-start' -> '/usr/local/share/doc/gitflow/hooks/post-flow-release-start'
'gitflow/hooks/post-flow-release-track' -> '/usr/local/share/doc/gitflow/hooks/post-flow-release-track'
'gitflow/hooks/pre-flow-feature-delete' -> '/usr/local/share/doc/gitflow/hooks/pre-flow-feature-delete'
'gitflow/hooks/pre-flow-feature-finish' -> '/usr/local/share/doc/gitflow/hooks/pre-flow-feature-finish'
'gitflow/hooks/pre-flow-feature-publish' -> '/usr/local/share/doc/gitflow/hooks/pre-flow-feature-publish'
'gitflow/hooks/pre-flow-feature-pull' -> '/usr/local/share/doc/gitflow/hooks/pre-flow-feature-pull'
'gitflow/hooks/pre-flow-feature-start' -> '/usr/local/share/doc/gitflow/hooks/pre-flow-feature-start'
'gitflow/hooks/pre-flow-feature-track' -> '/usr/local/share/doc/gitflow/hooks/pre-flow-feature-track'
'gitflow/hooks/pre-flow-hotfix-delete' -> '/usr/local/share/doc/gitflow/hooks/pre-flow-hotfix-delete'
'gitflow/hooks/pre-flow-hotfix-finish' -> '/usr/local/share/doc/gitflow/hooks/pre-flow-hotfix-finish'
'gitflow/hooks/pre-flow-hotfix-publish' -> '/usr/local/share/doc/gitflow/hooks/pre-flow-hotfix-publish'
'gitflow/hooks/pre-flow-hotfix-start' -> '/usr/local/share/doc/gitflow/hooks/pre-flow-hotfix-start'
'gitflow/hooks/pre-flow-release-branch' -> '/usr/local/share/doc/gitflow/hooks/pre-flow-release-branch'
'gitflow/hooks/pre-flow-release-delete' -> '/usr/local/share/doc/gitflow/hooks/pre-flow-release-delete'
'gitflow/hooks/pre-flow-release-finish' -> '/usr/local/share/doc/gitflow/hooks/pre-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/pre-flow-release-start'
'gitflow/hooks/pre-flow-release-track' -> '/usr/local/share/doc/gitflow/hooks/pre-flow-release-track'
```

2.3 Завершение установки gitflow

[3] Установка gh в Fedora Linux

С помощью команды, которая приведена ниже, устанавливаем gh в Fedora Linux. (скриншоты 3.1–3.2)

```
sudo dnf install gh
```

```
[lrabdullina@fedora tmp]$ sudo dnf install gh
Последняя проверка окончания срока действия метаданных: 1:12:07 назад, Чт 21 апр 2022 19:13:10.
Зависимости разрешены.
=====
Пакет          Архитектура      Версия            Репозиторий      Размер
```

3.1 Ввод команды для установки gh в Fedora Linux

```
Последняя проверка окончания срока действия метаданных: 1:12:07 назад, Чт 21 апр 2022 19:13:10.
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                                585 kB/s | 6.8 MB    00:11
=====
Общий размер                                             567 kB/s | 6.8 MB    00:12
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверка транзакции
Тест транзакции проведен успешно.
Выполнение транзакции
Подготовка : 1/1
Установка : gh-2.7.0-1.fc35.x86_64 1/1
Запуск скрипглета: gh-2.7.0-1.fc35.x86_64 1/1
Проверка : gh-2.7.0-1.fc35.x86_64 1/1
Установлен:
gh-2.7.0-1.fc35.x86_64
```

3.2 Установка gh в Fedora Linux, вывод информации

[4] Базовая настройка git

Используя следующие команды, зададим имя и почту владельца репозитория (заранее будем использовать ту почту, с которой мы регистрировались на GitHub), а также настроим utf-8 в выводе сообщений git. (скриншот 4.1)

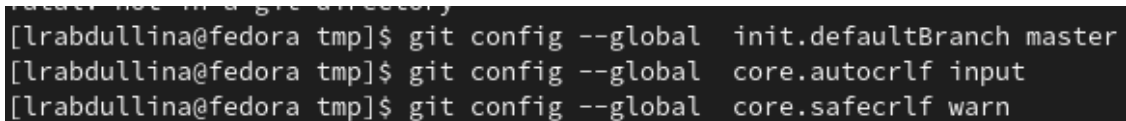
```
git config --global user.name AbdullinaLR
git config --global user.email owlwithcookie@gmail.com
git config --global core.quotepath false
```

```
[lrabdullina@fedora tmp]$ git config --global user.name AbdullinaLR
[lrabdullina@fedora tmp]$ git config --global user.email owlwithcookie@gmail.com
[lrabdullina@fedora tmp]$ git config --global core.quotepath false
```

4.1 Задаем имя и почту владельца репозитория, используя свои личные данные, а также настраиваем вывод сообщений

Настраиваем верификацию и подписание коммитов git, задаем имя начальной ветки (master), а также параметры autocrlf и autosafecrlf, используя следующий набор команд: (скриншот 4.2)

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

A screenshot of a terminal window showing three git configuration commands being executed. The prompt is [lrabdullina@fedora tmp]\$. The commands are: git config --global init.defaultBranch master, git config --global core.autocrlf input, and git config --global core.safecrlf warn.

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

4.2 Выполнение команд: настраивание верификации, подписание коммитов, имя начальной ветки – master.

[5] Создание ключа ssh

Ключ ssh можно создать двумя разными способами:

Первое: по алгоритму rsa с ключём размером 4096 бит:

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

Второе: по алгоритму ed25519:

```
ssh-keygen -t ed25519
```

Мы опробовали 2 способа, чтобы убедиться в их работоспособности и создали, тем самым, ключ ssh (скриншоты 5.1 – 5.2)

```
[lrabdullina@fedora tmp]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/lrabdullina/.ssh/id_rsa): home
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in home
Your public key has been saved in home.pub
The key fingerprint is:
SHA256:DY9NQaniiMFPwNWCKktg8GVirArGycFFV1aJ5UDRga0 lrabdullina@fedora
The key's randomart image is:
+---[RSA 4096]-----+
|+o==+oo*o==.      |
|.*=+o 00o+..      |
|=++ . . oo.        |
|=*o . .E.B         |
|*. = o .S +        |
|o . o .            |
|                   |
|                   |
|                   |
+-----[SHA256]-----+
[lrabdullina@fedora tmp]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
```

5.1 Ввод первого способа, создание ключа, вывод картинки, соответствующей ключу

```
Enter file in which to save the key (/home/lrabdullina/.ssh/id_ed25519): home
home already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in home
Your public key has been saved in home.pub
The key fingerprint is:
SHA256:aiUrqjTdyzxM9fA/jLnomcyUllbSwQcrk5HBDfoBYF0 lrabdullina@fedora
The key's randomart image is:
+--[ED25519 256]--+
|      oo.oE=.      |
|      . .oo+.o     |
|      . = + .      |
|      o. = o       |
|      o So o       |
|      . ..=.o+     |
| o .+.+o +=       |
| . ..o== *o +     |
|... +oB .. .      |
+-----[SHA256]-----+
```

5.2 Ввод второго способа, перезапись ключа, повторный вывод картинки (уже другой)

[6] Создаём ключи pgp

Генерируем ключ, используя следующую команду (скриншот 6.1):

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

Из предложенных опций выбираем (скриншоты 6.1 - 6.3):

- тип RSA and RSA;
- размер 4096;
- срок действия : 0

GPG запросит личную информацию, которая сохранится в ключе:

- Имя
- Адрес электронной почты (он соответствует адресу, используемому на GitHub)
- Комментарий

```
[lrabdullina@fedora 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: создан каталог '/home/lrabdullina/.gnupg'
gpg: создан щит с ключами '/home/lrabdullina/.gnupg/pubring.kbx'
Выберите тип ключа:
  (1) RSA and RSA
  (2) DSA and Elgamal
  (3) DSA (sign only)
```

6.1 Генерация ключа, создание каталога


```

(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: AbdullinaLR
Адрес электронной почты: owlwithcookie@gmail.com
Примечание: :)
Недопустимый символ в примечании
Примечание: hello
Вы выбрали следующий идентификатор пользователя:
    "AbdullinaLR (hello) <owlwithcookie@gmail.com>"

Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? 0
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.
Q
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.
jhk
jjj
gpg: /home/lrabdullina/.gnupg/trustdb.gpg: создана таблица доверия
gpg: ключ DE1663A4A93A7740 помечен как абсолютно доверенный
gpg: создан каталог '/home/lrabdullina/.gnupg/openpgp-revocs.d'
gpg: сертификат отзыва записан в '/home/lrabdullina/.gnupg/openpgp-revocs.d/FC671B506BE104B5C1BA7803DE1663A4A93A7740.rev'.
открытый и секретный ключи созданы и подписаны.

pub   rsa4096 2022-04-21 [SC]
      FC671B506BE104B5C1BA7803DE1663A4A93A7740
uid    AbdullinaLR (hello) <owlwithcookie@gmail.com>
sub    rsa4096 2022-04-21 [E]

```

6.2 Выбор нужной информации, заполнение личных данных

```

на клавиатуре, движения мыши, обращения к дискам); это даст генератору
случайных чисел больше возможностей получить достаточное количество энтропии.
Q
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.
jhk
jjj
gpg: /home/lrabdullina/.gnupg/trustdb.gpg: создана таблица доверия
gpg: ключ DE1663A4A93A7740 помечен как абсолютно доверенный
gpg: создан каталог '/home/lrabdullina/.gnupg/openpgp-revocs.d'
gpg: сертификат отзыва записан в '/home/lrabdullina/.gnupg/openpgp-revocs.d/FC671B506BE104B5C1BA7803DE1663A4A93A7740.rev'.
открытый и секретный ключи созданы и подписаны.

pub   rsa4096 2022-04-21 [SC]
      FC671B506BE104B5C1BA7803DE1663A4A93A7740
uid    AbdullinaLR (hello) <owlwithcookie@gmail.com>
sub    rsa4096 2022-04-21 [E]

```

6.3 Процесс генерации

[7] Добавление PGP ключа в GitHub

Выводим список ключей и копируем отпечаток приватного ключа(скриншоты 7.1–7.2):

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

```
[lrabdullina@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
```

7.1 Ввод команды для вывода списка ключей

```
/home/lrabdullina/.gnupg/pubring.kbx
-----
sec   rsa4096/DE1663A4A93A7740 2022-04-21 [SC]
      FC671B506BE104B5C1BA7803DE1663A4A93A7740
uid   [ абсолютно ] AbdullinaLR (hello) <owlwithcookie@gmail.com>
ssb   rsa4096/47B52FDE96BAC4FD 2022-04-21 [E]

[lrabdullina@fedora tmp]$ gpg --armor --export <PGP Fingerprint> | xclip -sel clip
bash: синтаксическая ошибка рядом с неожиданным маркером «|»
[lrabdullina@fedora tmp]$ gpg --armor --export <PGP Fingerprint> | xclip -sel clip
bash: синтаксическая ошибка рядом с неожиданным маркером «|»
[lrabdullina@fedora tmp]$ gpg --armor --export <PGP Fingerprint> xclip -sel clip
bash: PGP: Нет такого файла или каталога
[lrabdullina@fedora tmp]$ gpg --armor --export <PGP Fingerprint>
bash: синтаксическая ошибка рядом с неожиданным маркером «newline»
[lrabdullina@fedora tmp]$ gpg --armor --export <PGP Fingerprint>| xclip -sel clip
bash: PGP: Нет такого файла или каталога
[lrabdullina@fedora tmp]$ gpg --armor --export <PGP Fingerprint>|xclip -sel clip
bash: PGP: Нет такого файла или каталога
[lrabdullina@fedora tmp]$ gpg --armor --export DE1663A4A93A7740 | xclip -sel clip
bash: xclip: command not found...
Install package 'xclip' to provide command 'xclip'? [N/y] y

* Waiting in queue...
The following packages have to be installed:
xclip-0.13-15.git11cba61.fc35.x86_64  Command line clipboard grabber
Proceed with changes? [N/y] y

* Waiting in queue...
* Waiting for authentication...
```

7.2 Список ключей

С помощью формата строки находим отпечаток ключа (скриншот 7.3):

```
sec   rsa4096/DE1663A4A93A7740 2022-04-21 [SC]
      FC671B506BE104B5C1BA7803DE1663A4A93A7740
```

7.3 Копирование отпечатка ключа

Копируем сгенерированный PGP ключ в буфер обмена при помощи следующей команды: (скриншоты 7.4–7.5)

```
gpg --armor --export | xclip -sel clip
```

```
[lrabdullina@fedora tmp]$ gpg --armor --export DE1663A4A93A7740 | xclip -sel clip
bash: xclip: command not found...
Install package 'xclip' to provide command 'xclip'? [N/y] y

* Waiting in queue...
The following packages have to be installed:
xclip-0.13-15.git11cba61.fc35.x86_64    Command line clipboard grabber
Proceed with changes? [N/y] y

* Waiting in queue...
* Waiting for authentication...
```

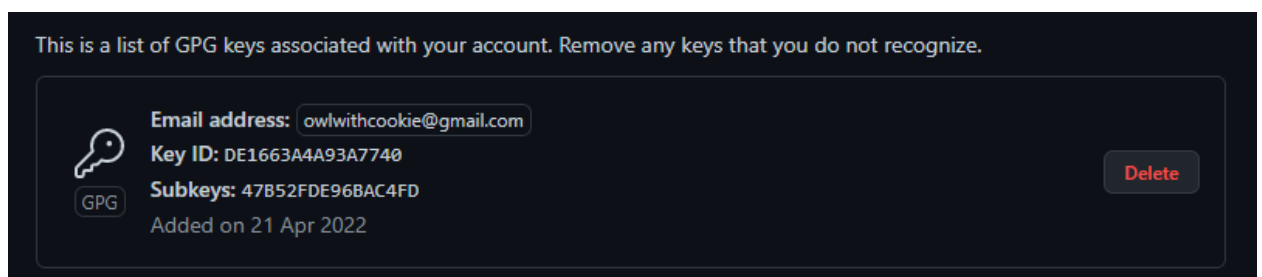
7.4 Ввод команды для копирования PGP ключа в буфер обмена и установка библиотеки xclip

```
* Waiting in queue...
* Downloading packages...
* Requesting data...
* Testing changes...
* Installing packages...

[lrabdullina@fedora tmp]$ gpg --armor --export DE1663A4A93A7740 | xclip -sel clip
[lrabdullina@fedora tmp]$ git config --global user.signingkey DE1663A4A93A7740
```

7.5 Копирование ключа в буфер обмена

Переходим в настройки GitHub (<https://github.com/settings/keys>), нажимаем на кнопку New GPG key и вставляем полученный ключ в поле ввода.



7.6 Добавление ключа PGP в GitHub

[8] Настройка автоматических подписей коммитов git

Используя введённый мною email, укажем Git применять его при подписи коммитов (скриншот 8.1):

```
git config --global user.signingkey  
git config --global commit.gpgsign true  
git config --global gpg.program $(which gpg2)
```

```
[lrabdullina@fedora tmp]$ git config --global user.signingkey DE1663A4A93A7740  
[lrabdullina@fedora tmp]$ git config --global commit.gpgsign true  
? What account do you want to log into? [Use arrow to move, type to filter]
```

8.1 Настройка автоматических подписей коммитов git

[9] Настройка gh

Авторизуемся и настроим все необходимое (скриншот 9.1):

```
[lrabdullina@fedora Операционные системы]$ gh auth login  
? What account do you want to log into? GitHub.com  
? You're already logged into github.com. Do you want to re-authenticate? Yes  
? What is your preferred protocol for Git operations? SSH  
? Generate a new SSH key to add to your GitHub account? Yes  
? Enter a passphrase for your new SSH key (Optional)  
? How would you like to authenticate GitHub CLI? Login with a web browser  
  
! First copy your one-time code: C902-833B  
Press Enter to open github.com in your browser...  
✓ Authentication complete.  
- gh config set -h github.com git_protocol ssh  
✓ Configured git protocol  
✓ Uploaded the SSH key to your GitHub account: /home/lrabdullina/.ssh/id_ed25519.pub  
✓ Logged in as AbdullinaLR
```

9.1 Авторизация через консоль

[10] Создание репозитория курса на основе шаблона

Необходимо создать шаблон рабочего пространства. Для 2021–2022 учебного года и предмета «Операционные системы» (код предмета os-intro) создание репозитория примет следующий вид: (скриншоты 10.1 – 10.2)

```
mkdir -p ~/work/study/2021-2022/"Операционные системы"
cd ~/work/study/2021-2022/"Операционные системы"
gh repo create study_2021-2022_os-intro --template=yamadharm/course-directory-student-template --public
git clone --recursive git@github.com:AbdukkinaLR:/study_2021-2022_os-intro.git os-intro
```

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

10.1 Создание нового каталога «Операционные системы» и переход в этот каталог

```
[lrabdullina@fedora Операционные системы]$ git clone --recursive git@github.com:AbdullinaLR/study_2021-2022_os-intro.git os-intro
Клонирование в «os-intro»...
remote: Enumerating objects: 20, done.
remote: Counting objects: 100% (20/20), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 20 (delta 2), reused 15 (delta 2), pack-reused 0
Получение объектов: 100% (20/20), 12.49 КиБ | 1.04 МиБ/с, готово.
Определение изменений: 100% (2/2), готово.
Подмодуль «template/presentation» (https://github.com/yamadharm/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharm/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/home/lrabdullina/work/study/2021-2022/Операционные системы/os-intro/template/presentation»...
remote: Enumerating objects: 42, done.
remote: Counting objects: 100% (42/42), done.
remote: Compressing objects: 100% (34/34), done.
remote: Total 42 (delta 9), reused 40 (delta 7), pack-reused 0
```

10.2 Клонирование информации

[11] Настройка каталога курса

Перейдём в каталог курса, удалим лишние файлы, создадим необходимые каталоги и отправим файлы на сервер, с помощью следующих команд: (скриншоты 11.1–11.2)

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

```
[lrabdullina@fedora Операционные системы]$ cd ~/work/study/2021-2022/"Операционные системы"/os-intro
[lrabdullina@fedora os-intro]$ rm package.json
[lrabdullina@fedora os-intro]$ make COURSE=os-intro
[lrabdullina@fedora os-intro]$ git add
Ничего не указано, ничего не добавлено.
подсказка: Maybe you wanted to say 'git add .' ?
подсказка: Turn this message off by running
подсказка: "git config advice.addEmptyPathsSpec false"
[lrabdullina@fedora os-intro]$ git add .
[lrabdullina@fedora os-intro]$ git commit -am 'feat(main): make course structure'
[master ee6e2a4] feat(main): make course structure
149 files changed, 16590 insertions(+), 14 deletions(-)
create mode 100644 labs/lab01/presentation/Makefile
create mode 100644 labs/lab01/presentation/presentation.md
create mode 100644 labs/lab01/report/Makefile
create mode 100644 labs/lab01/report/bib/cite.bib
create mode 100644 labs/lab01/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab01/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab01/report/report.md
create mode 100644 labs/lab02/presentation/Makefile
create mode 100644 labs/lab02/presentation/presentation.md
```

11.1 Переход в каталог курса, удаление лишних файлы, создание необходимых каталогов.

```
create mode 100644 project-personal/stage5/report/bib/cite.bib
create mode 100644 project-personal/stage5/report/image/placeimg_800_600_tech.jpg
create mode 100644 project-personal/stage5/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 project-personal/stage5/report/report.md
create mode 100644 project-personal/stage6/presentation/Makefile
create mode 100644 project-personal/stage6/presentation/presentation.md
create mode 100644 project-personal/stage6/report/Makefile
create mode 100644 project-personal/stage6/report/bib/cite.bib
create mode 100644 project-personal/stage6/report/image/placeimg_800_600_tech.jpg
create mode 100644 project-personal/stage6/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 project-personal/stage6/report/report.md
create mode 100644 structure
[lrabdullina@fedora os-intro]$ git push
Перечисление объектов: 20, готово.
Подсчет объектов: 100% (20/20), готово.
Сжатие объектов: 100% (16/16), готово.
Запись объектов: 100% (19/19), 266.52 КиБ | 770.00 КиБ/с, готово.
Всего 19 (изменений 2), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To github.com:AbdullinaLR/study_2021-2022_os-intro.git
 39a6ebe...ee6e2a4 master -> master
[lrabdullina@fedora os-intro]$
```

11.2 Отправка файлов на сервер

Контрольные вопросы:

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

Система контроля версий – это набор программного обеспечения, который предназначается для работы нескольких человек над одним проектом.

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

Хранилище – сервер, на котором хранится вся история изменений проекта.

Commit – фиксация «дельта-изменений», то есть изменений с последней версии с его последующей записью как версия в истории.

История – список всех изменений проекта (с возможностью «откатить» в любую точку изменений)

Рабочая копия – все файлы проекта, с которыми происходит основная работа.

3. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.

В централизованных VCS необходим центральный репозиторий для хранения файлов. Например, Subversion. В децентрализованных VCS центральный репозиторий не обязателен. Например, Git.

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

Инициализация системы управления версиями git через git init. Работа над проектом используя git-flow для отдельных частей проекта. Git commit – для фиксации изменений. При необходимости использование удаленного сервера для хранения с помощью remote и git push. Удаленный сервер также позволяет работать с нескольких устройств с использованием git pull.

5. Опишите порядок работы с общим хранилищем VCS.

При существующей версии проекта в хранилище, скопировать его оттуда через git pull. Использовать git-flow для работы над частями проекта. После окончания работы нужно зафиксировать изменения через git commit. Загрузить в хранилище через git push.

6. Каковы основные задачи, решаемые инструментальным средством git?

Ведение истории изменений, совмещение версий, веток и пр., фиксирование изменений, а также откат к прошлым версиям.

7. Назовите и дайте краткую характеристику командам git.

Git init – инициализация проекта с системой контроля версий

Git commit – фиксация изменений в отслеживаемом файле

Git add – добавление файла/директории в систему контроля версий как отслеживаемое

Git push – загрузка локальной версии на сервер

Git pull – выгрузка локальной версии на сервер

Git fetch – «часть» команды git pull, которая собирает актуальную версию (но не вносит в работу)

Git merge – слияние веток

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

При работе с локальным репозиторием все изменения хранятся локально и не выгружаются на удаленный сервер. Не нужно использование команд push, pull и т. п. При работе с удаленным репозиторием для отображения изменения, они должны быть выгружены на удаленный сервер (с помощью команды pull).

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

Ветви позволяют «разделять» части работы, работать отдельно над каждой имплементацией. Использование ветвей дает возможность комфортной проверки и обработки новой информации, которую ввели в основную ветвь.

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

Можно игнорировать с помощью .gitignore файла. В нем указываются пути, названия и другие идентификации нежелательных объектов (которые не

будут учитываться в commit). Это полезно для устранения «остаточных» файлов, которые более не являются полезными. А также это могут быть конфиденциальные данные, например пароли.

Вывод:

В ходе лабораторной работы мы изучили идеологию и применение средств контроля версий, а также освоение умения по работе с git.