

# Отчет по лабораторной работе №2

Операционные системы

---

Ицков А.С.

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Ицков Андрей Станиславович
- НКАбд-02-2024 № Студенческого билета: 1132246737
- Российский университет дружбы народов
- [https://github.com/a-its/study\\_2024-2025\\_os-intro](https://github.com/a-its/study_2024-2025_os-intro)

## Цель работы

---

Целью данной работы является изучение идеологии и применения средств контроля версий и освоение умения по работе с git.

## Задание

---

1. Создать базовую конфигурацию для работы с git.
2. Создать ключ SSH.
3. Создать ключ PGP.
4. Настроить подписи git.
5. Зарегистрироваться на Github.
6. Создать локальный каталог для выполнения заданий по предмету.

## Теоретическое введение

---



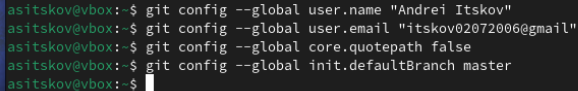
Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещать изменения, произведённые разными участниками проекта, производить откат к любой более ранней версии проекта, если это требуется.

## Выполнение лабораторной работы

---

Установим гит командой `dnf install git`, установим gh командой `dnf install gh`

Открываем терминал. При помощи команд `git config --global user.name` и `git config --global user.email` зададим имя пользователя и адрес электронной почты. При помощи команды `git config --global core.quotepath false` настроим utf-8 в выводе сообщений git. При помощи команды `git config --global init.defaultBranch master` зададим начальной ветке имя master.



```
asitskov@vbox:~$ git config --global user.name "Andrei Itskov"
asitskov@vbox:~$ git config --global user.email "itskov02072006@gmail"
asitskov@vbox:~$ git config --global core.quotepath false
asitskov@vbox:~$ git config --global init.defaultBranch master
asitskov@vbox:~$
```

Рис. 1: Ввод команд в терминал

## Создание ssh ключа.

Для создания ключа используем команду `ssh-keygen -t` в терминале. Зададим ключу размер 4096 бит. Сменим пароль при помощи команды `ssh-keygen -p`.

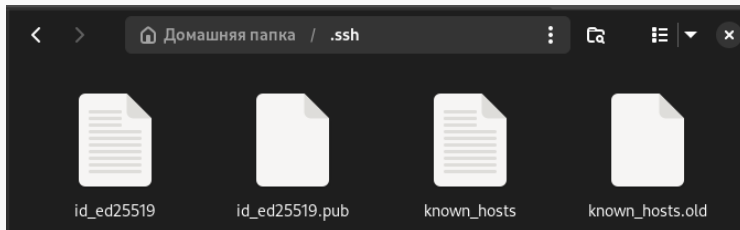


Рис. 2: Создание ключа

## Добавление ssh-ключа в учетную запись ГитХаб.

Копируем созданный ключ и переносим его на сайт гитхаб в раздел ssh и gpg keys.

Создаем новый ключ, задаем ему название и переносим ключ в поле key, добавляем ключ на сайт.

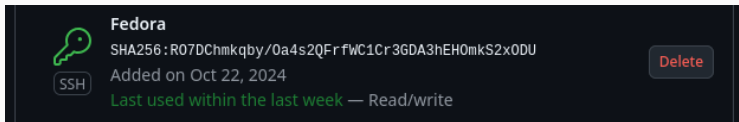
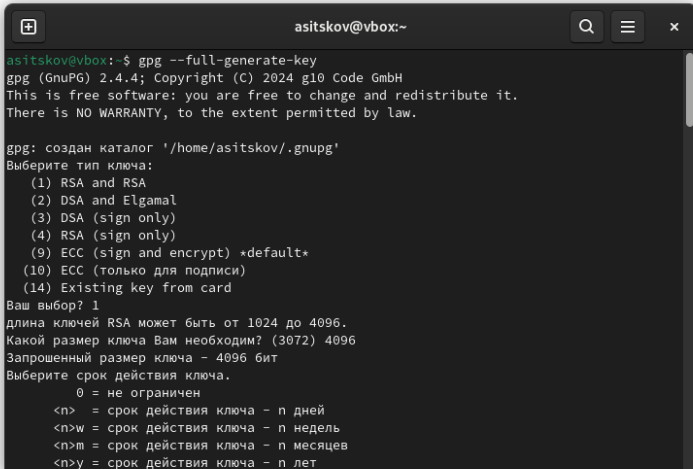


Рис. 3: Новый ключ ssh

## Создание PGP ключа.

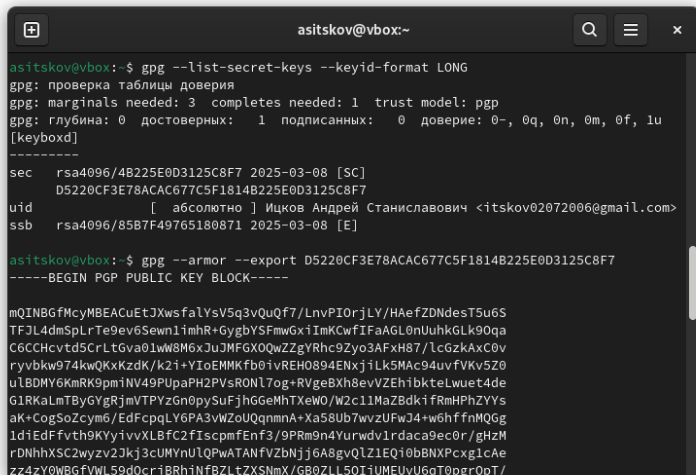
Генерируем ключ командой `gpg --full-generate-key`, настраиваем его по заданным требованиям.



```
asitskov@vbox:~  
asitskov@vbox:~$ gpg --full-generate-key  
gpg (GnuPG) 2.4.4; Copyright (C) 2024 g10 Code GmbH  
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/asitskov/.gnupg'  
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  
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
```



Выводим ключ в терминал командой `gpg --list-secret-keys --keyid-format LONG`. После этого экспортируем его командой `gpg --armor --export`.

A screenshot of a terminal window with a dark background. The title bar shows the username 'asitskov@vbox:~'. The terminal displays the output of two GPG commands. The first command, 'gpg --list-secret-keys --keyid-format LONG', shows details about a secret key, including its ID, creation date, and owner information. The second command, 'gpg --armor --export D5220CF3E78ACAC677C5F1814B225E0D3125C8F7', exports the key in ASCII armor format, starting with '-----BEGIN PGP PUBLIC KEY BLOCK-----' and ending with a long base64-encoded string.

```
asitskov@vbox:~$ 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
[keyboard]
-----
sec   rsa4096/4B225E0D3125C8F7 2025-03-08 [SC]
      D5220CF3E78ACAC677C5F1814B225E0D3125C8F7
uid           [ абсолютно ] Ицков Андрей Станиславович <itskov02072006@gmail.com>
ssb   rsa4096/85B7F49765180871 2025-03-08 [E]

asitskov@vbox:~$ gpg --armor --export D5220CF3E78ACAC677C5F1814B225E0D3125C8F7
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGfMcyMBEACuEtJXwsfalYsV5q3vQuQf7/LnvPIOrjLY/HAefZDNdesT5u6S
TFJL4dmSpLrTe9ev6Sewn1imhR+GygbYSFmwGxiImKCwfIFaAGL0nUuhkGLk90qa
C6CCHcvtD5CrLtGva01wW8M6xJuJMF6X0QwZZgYRhc9Zyo3AFxH87/lcGzkAxC0v
ryvbkw974kwQKxKzdK/k2i+YIoEMMKfb0ivREH0894ENxjilK5MAc94uvfVKv5Z0
ulBDMY6KmRK9pmiNV49PUpaPH2PVsRONl7og+RVgeBXh8evVZEhibkteLwuet4de
G1RkaLmTByGYgRjmVTPYzGn0pySuFjhGGeMhTXew0/W2c1lMaZBdki fRmHPHZZYs
aK+CogSoZcym6/EdFcpqLY6PA3vWZoUQqnmnA+Xa58Ub7wvzUFwJ4+w6hffnMQGg
ldiEdFfvth9KYyivvXLBfC2fIscpmfEnf3/9PRm9n4Yurwdv1rdaca9ec0r/gHzM
rDNhhXSC2wyzv2Jkj3cUMYnUlQPwATANfVZbNjj6A8gvQLZ1EQi0bBNXPcxglcAe
zz4zY0WBGfVWL59d0criBRhiNfBZLtZXSnmX/GB0ZLL50iIUMEUvU6aT0perOpT/
```

Скопировав ключ, переносим его на ГитХаб, создаем на сайте новый ключ и вставляем скопированный ключ в необходимое поле.

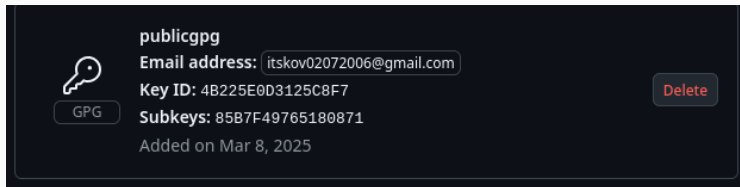
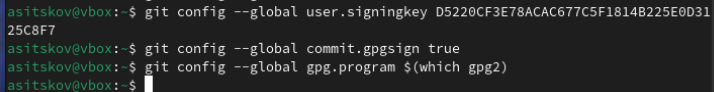


Рис. 6: Новый ключ PGP

## Настройка автоматических подписей коммитов git

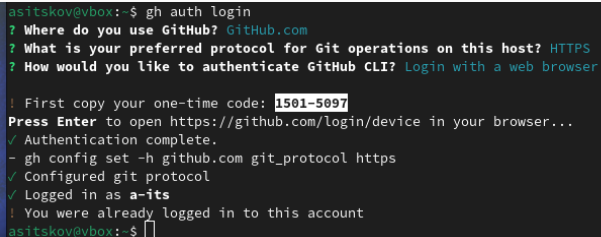
При помощи команд `git config --global user.signingkey`, `git config --global commit.gpgsign true` и `git config --global gpg.program $(which gpg2)` самостоятельно выбираем подписи коммитов в git.

A terminal window with a dark background and light green text. It shows four lines of commands being executed in a shell. The first line sets the global signing key. The second line enables commit signing. The third line sets the gpg program to gpg2. The fourth line shows the prompt after the command.

```
asitskov@vbox:~$ git config --global user.signingkey D5220CF3E78ACAC677C5F1814B225E0D31
25C8F7
asitskov@vbox:~$ git config --global commit.gpgsign true
asitskov@vbox:~$ git config --global gpg.program $(which gpg2)
asitskov@vbox:~$
```

Рис. 7: Настройка необходимых подписей коммитов

Введя в терминал команду `gh auth login`, ответим на необходимые в терминале вопросы, после чего авторизуемся через браузер.



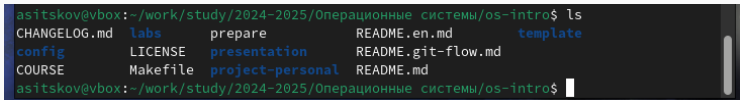
```
asitskov@vbox:~$ gh auth login
? Where do you use GitHub? GitHub.com
? What is your preferred protocol for Git operations on this host? HTTPS
? How would you like to authenticate GitHub CLI? Login with a web browser

! First copy your one-time code: 1501-5097
Press Enter to open https://github.com/login/device in your browser...
✓ Authentication complete.
- gh config set -h github.com git_protocol https
✓ Configured git protocol
✓ Logged in as a-its
! You were already logged in to this account
asitskov@vbox:~$
```

Рис. 8: Настройка gh и авторизация в браузере

## Создание и настройка репозитория курса.

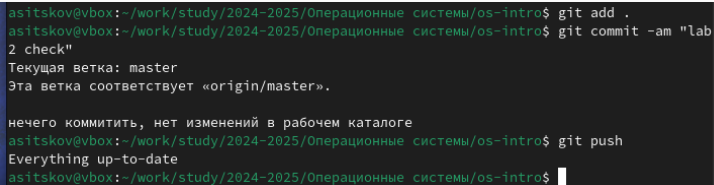
Используя команды `mkdir`, `gh repo`, `create study` и `git clone` создаем репозиторий курса.

A terminal window with a dark background and light green text. The prompt is 'asitskov@vbox:~/work/study/2024-2025/Операционные системы/os-intro\$'. The command 'ls' has been executed, displaying a list of files and directories in a multi-column format. The files are: CHANGELOG.md, config, COURSE, labs, LICENSE, Makefile, prepare, presentation, project-personal, README.en.md, README.git-flow.md, README.md, and template.

```
asitskov@vbox:~/work/study/2024-2025/Операционные системы/os-intro$ ls
CHANGELOG.md  labs      prepare    README.en.md  template
config        LICENSE   presentation  README.git-flow.md
COURSE        Makefile  project-personal  README.md
asitskov@vbox:~/work/study/2024-2025/Операционные системы/os-intro$
```

Рис. 9: Созданный репозиторий, папка первой лабораторной работы

Отправляем файлы первой лабораторной работы на сервер.

A terminal window with a dark background and light green text. The prompt is 'asitskov@vbox:~/work/study/2024-2025/Операционные системы/os-intro\$'. The user enters 'git add .' followed by 'git commit -am "lab 2 check"'. The terminal shows the current branch is 'master' and it matches 'origin/master'. It then says 'nothing to commit, working tree clean'. Finally, the user enters 'git push' and the terminal shows 'Everything up-to-date'.

```
asitskov@vbox:~/work/study/2024-2025/Операционные системы/os-intro$ git add .
asitskov@vbox:~/work/study/2024-2025/Операционные системы/os-intro$ git commit -am "lab
2 check"
Текущая ветка: master
Эта ветка соответствует «origin/master».

ничего коммитить, нет изменений в рабочем каталоге
asitskov@vbox:~/work/study/2024-2025/Операционные системы/os-intro$ git push
Everything up-to-date
asitskov@vbox:~/work/study/2024-2025/Операционные системы/os-intro$
```

Рис. 10: Отправка файлов на сервер

## Вывод

---

В результате выполнения данной лабораторной работы я приобрел необходимые навыки работы с гит, научился созданию репозитория, gpg и ssh ключей, настроил каталог курса и авторизовался в gh.