

# Лабораторная работа №2

## Операционные системы

---

Закиров Н.Д., НКАбд-03-23

26 февраля 2024

Российский университет дружбы народов, Москва, Россия

# Информация

---

- Закиров Нурислам Дамирович
- Студент 1 курса
- Группа: НКАбд-03-23
- Российский университет дружбы народов

# Лабораторная работа

---

Целью данной лабораторной работы является изучение идеологии и применение средств контроля версий и приобретение знаний по работе с git.

## Задание

---

- Зарегистрироваться на GitHub
- Создать базовую конфигурацию для работы с git
- Создать ключ SSH
- Создать ключ GPG
- Настроить подписи Git
- Создать локальный каталог для выполнения заданий по предмету.

## Установка программного обеспечения

Заранее регистрация в git уже была успешно выполнена, поэтому сразу переходим к следующим этапам. Устанавливаем git при помощи `dnf install git` и устанавливаем gh при помощи `dnf install gh` (рис. 1).

```
[ndzakirov@ndzakirov ~]$ sudo -i
[sudo] пароль для ndzakirov:
[root@ndzakirov ~]# dnf install git
Последняя проверка окончания срока действия метаданных:
 2024 18:21:13.
Пакет git-2.43.0-1.fc38.x86_64 уже установлен.
Зависимости разрешены.
Нет действий для выполнения.
Выполнено!
[root@ndzakirov ~]# dnf install gh
Последняя проверка окончания срока действия метаданных:
 2024 18:21:13.
Пакет gh-2.36.0-1.fc38.x86_64 уже установлен.
Зависимости разрешены.
Нет действий для выполнения.
Выполнено!
```

Рис. 1: Установка git и gh



Далее настраиваем верификацию и подписание коммитов git, задаваем имя начальной ветки master, параметр autocrlf и safecrlf(рис. 2).

```
[root@ndzakirov ~]# git config --global core.quotePath false
[root@ndzakirov ~]# git config --global init.defaultBranch master~
[root@ndzakirov ~]# git config --global core.autocrlf input
[root@ndzakirov ~]# git config --global core.safecrlf warn
```

**Рис. 2:** Базовая настройка git

## Создание ключа SSH

Создаем ключ ssh по алгоритму rsa с ключём размером 4096 бит и по алгоритму ed25519 (рис. 3).

```
[root@endzakirov ~]# ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_ed25519.
Your public key has been saved in /root/.ssh/id_ed25519.pub.
The key fingerprint is:
SHA256:Tx1z91A4J5hzS2oivTmg76UuNUtHQMksZcIt2FU08E
The key's randomart image is:
+--[ED25519 256]--+
|  + +=E    o  .. |
| . = *=..  + *.. |
| *   oo   o=o= . |
| .    o +.o+.o. |
|    .S+.=.   . |
| . +o=       |
|   + +o.    |
| . oo       |
| ++        |
```

# Создание ключа GPG

Следующим этапом мы генерируем ключ `pgp` выбирая следующие опции: тип RSA and RSA, размер 4096, срок действия неограниченный (рис. 4).

```
[root@ndzakirov ~]# gpg --full-generate-key
gpg (GnuPG) 2.4.0; 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 '/root/.gnupg'
gpg: создан щит с ключами '/root/.gnupg/pubring.kbx'
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
```

После чего, мы копируем данный нам ключ командой `gpg --armor --export | xclip -sel clip` в буфер обмена, где вместо “PGP Fingerprint” вставляем данные значения.(рис. 5).

```
[root@ndzakirov ~]# gpg --armor --export BFA077A162927416 | xclip -sel clip
[root@ndzakirov ~]#
```

**Рис. 5:** Копирование ключа в буфер обмена

## Создание ключа GPG

Теперь вставляем данный ключ при помощи комбинации Ctrl+V на официальном сайте GitHub в настройках GitHub, нажав на кнопку New GPG key(рис. 6).

### Add new GPG key

---

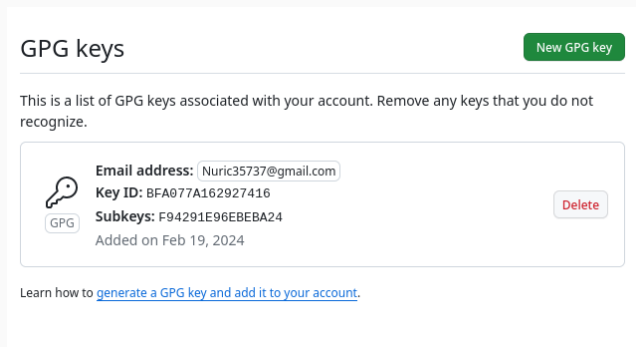
**Title**

**Key**  

```
-----BEGIN PGP PUBLIC KEY BLOCK-----  
  
mQINBGXTo3QBEADHOBWDanhIrtHo5MG/Wpe+QXazFT5Drftm+gX6Ltgf9NgHetd  
n5voLHQlkW072nvumM+Sj17bwuKYhBiqePWG2scRZTgvtYCSBESbMnErud9pFvm5  
VcKEcF3D7hB8YCZRzjzz2yD+5n3wOyeAjtWLMQ9lIm5wIIgD9Vp2wd23htosjL  
PkZx9vKNL9un6HQ/A4goPgHFuZiZKU3esRSulrrsRac6mfMEldgAdB065Ht/EqF  
2Q0YpNRvufxCVbyICITX640TV7DSXA7ZjqUzif6xsAsmZse2EUdyhtZ48rPsjLOf  
npCLQ8m/wypdogC0UMw6sVrmit9h8AapHhDWbdGYoa7v9tTSHppJxrA+L/qTEyXz  
EKD/YliSeoTJj7RxqpjvvOFFTVCEnfW148v/ImbET9aW8RWmEB8AniQnZ+GdQtMh
```

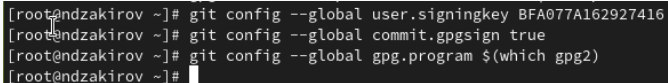
Add GPG key

Ключ успешно привязался(рис. 7).



**Рис. 7:** Окно GitHub

Следующим этапом мы используем введённый email и указываем Git применять его при подписи коммитов (рис. 8).

A terminal window with a dark background and light text. It shows four lines of commands being entered at a prompt. The first line sets the global signing key. The second line enables commit signing. The third line sets the gpg program to the one found by 'which'. The fourth line shows the prompt again with a cursor.

```
[root@ndzakirov ~]# git config --global user.signingkey BFA077A162927416
[root@ndzakirov ~]# git config --global commit.gpgsign true
[root@ndzakirov ~]# git config --global gpg.program $(which gpg2)
[root@ndzakirov ~]#
```

**Рис. 8:** Настройка подписей коммитов git

# Настройка gh

Далее мы авторизируемся в терминале через команду `gh auth login`.  
Отвечаем на все наводящие вопросы(рис. 9).

```
[root@ndzakirov ~]# gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations? HTTPS
X Sorry, your reply was invalid: "H" is not a valid answer, please try again
? 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: 27C9-5FE2
Press Enter to open github.com in your browser...
Running Firefox as root in a regular user's session is not supported. ($?)
ITY is /run/user/1001/.mutter-Xwaylandauth.00QAJ2 which is owned by ndzak
/usr/bin/xdg-open: строка 881: x-www-browser: команда не найдена
Running Firefox as root in a regular user's session is not supported. ($?)
ITY is /run/user/1001/.mutter-Xwaylandauth.00QAJ2 which is owned by ndzak
/usr/bin/xdg-open: строка 881: iceweasel: команда не найдена
/usr/bin/xdg-open: строка 881: seamonkey: команда не найдена
/usr/bin/xdg-open: строка 881: mozilla: команда не найдена
VMware: No 3D enabled (0, Выполнено).
libEGL warning: egl: failed to create dri2 screen

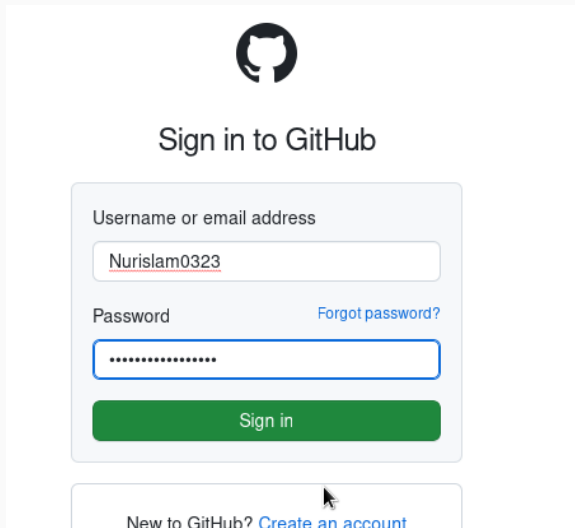
(epiphany:5252): Gtk-CRITICAL **: 22:05:54.360: Unable to register the app
on: GDBus.Error:org.freedesktop.DBus.Error.NameHasNoOwner: Could not activ
mote peer: unit failed.

** (process:2): WARNING **: 22:05:54.823: Failed to get atspi registered c
isteners: GDBus.Error:org.freedesktop.DBus.Error.NameHasNoOwner: Could not
ate remote peer: unit failed.

(epiphany:5252): Gtk-CRITICAL **: 22:05:59.768: Unable to register the app
on: GDBus.Error:org.freedesktop.DBus.Error.NameHasNoOwner: Could not activ
mote peer: unit failed.
```



После входим в свой аккаунт GitHub через браузер(рис. 10).



The screenshot shows the GitHub login interface. At the top is the GitHub logo (Octocat). Below it is the text "Sign in to GitHub". The login form is a light gray box containing two input fields: "Username or email address" with the value "Nurislam0323" and "Password" with masked characters. A blue link "Forgot password?" is next to the password field. A green "Sign in" button is at the bottom of the form. Below the form, a mouse cursor points to the text "New to GitHub? Create an account".

Sign in to GitHub

Username or email address

Nurislam0323

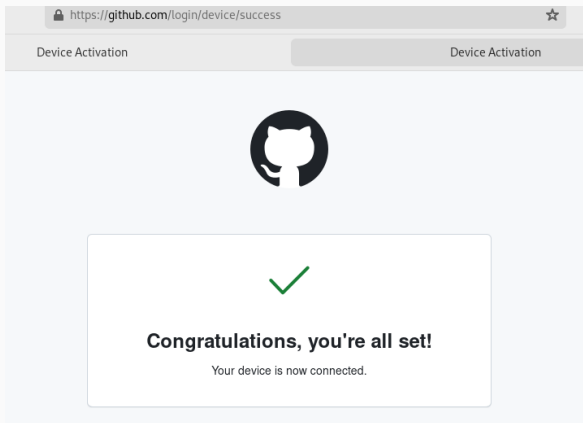
Password [Forgot password?](#)

.....

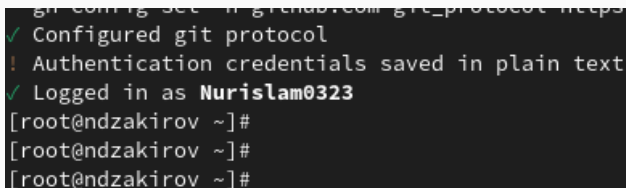
Sign in

New to GitHub? [Create an account](#)

После авторизации, Git просит ввести код, который мы заранее скопировали с терминала и вставили в браузер, после чего мы успешно авторизовались(рис. 11).



В терминале также подтверждается наша авторизация(рис. 12).



```
git config --set http.github.com git-protocol https
✓ Configured git protocol
! Authentication credentials saved in plain text
✓ Logged in as Nurislam0323
[root@ndzakirov ~]#
[root@ndzakirov ~]#
[root@ndzakirov ~]#
```

Рис. 12: Успешная авторизация

# Создание шаблона для рабочего пространства

Далее нам необходимо создать шаблон рабочего пространства. При помощи `mkdir -p ~/work/study/2023-2024/“Операционные системы”` мы создаем каталог, после чего переходим в нее. Создаем шаблон при помощи `gh repo create`(рис. 13).

```
[root@ndzakirov Операционные системы]# gh repo create study_2023-2024_os-intro --template=yamadharma/course-directory-student-template --public
✓ Created repository Nurislam0323/study_2023-2024_os-intro on GitHub
[root@ndzakirov Операционные системы]#
```

**Рис. 13:** Создание репозитории курса

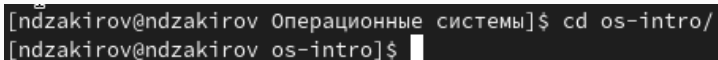
# Создание шаблона для рабочего пространства

После чего мы клонируем репозиторий курса на нашу систему при помощи git clone(рис. 14).

```
[ndzakirov@endzakirov Операционные системы]$ git clone --recursive https://github.com/Nurislam0323/study_2023-2024_os-intro.git os-intro
Клонирование в «os-intro»...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (31/31), done.
remote: Total 32 (delta 1), reused 18 (delta 0), pack-reused 0
Получение объектов: 100% (32/32), 18.60 КиБ | 1.09 МиБ/с, готово.
Оптимизация изменений: 100% (1/1), готово.
```

**Рис. 14:** Создание репозитории курса

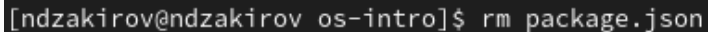
Переходим в созданную папку `os-intro` при помощи `cd`(рис. 15).

A terminal window with a dark background and light gray text. The first line shows the prompt '[ndzakirov@ndzakirov Операционные системы]\$' followed by the command 'cd os-intro/'. The second line shows the updated prompt '[ndzakirov@ndzakirov os-intro]\$' followed by a white cursor block.

```
[ndzakirov@ndzakirov Операционные системы]$ cd os-intro/  
[ndzakirov@ndzakirov os-intro]$
```

**Рис. 15:** Перемещение по каталогам

Удаляем лишние файлы при помощи rm (рис. 16).

A terminal window with a dark background. The prompt is [ndzakirov@ndzakirov os-intro]\$ and the command rm package.json is entered. A white cursor is visible at the end of the command.

```
[ndzakirov@ndzakirov os-intro]$ rm package.json
```

**Рис. 16:** Удаление файлов

Создаем необходимые каталоги при помощи echo и make(рис. 17).

```
[ndzakirov@ndzakirov os-intro]$ echo os-intro > COURSE
[ndzakirov@ndzakirov os-intro]$ make
Usage:
  make <target>

Targets:
  list           List of courses
  prepare       Generate directories structure
  submodule      Update submules

[ndzakirov@ndzakirov os-intro]$
```

**Рис. 17:** Создание каталогов



# Настройка каталога курса

Отправляем файлы на сервер GitHub при помощи git add, git commit и git push(рис. 18).

```
[ndzakirov@ndzakirov os-intro]$ git add .
[ndzakirov@ndzakirov os-intro]$ git commit -am 'feat(main): make course structure'
Author identity unknown

*** Пожалуйста, скажите мне кто вы есть.

Запустите

  git config --global user.email "you@example.com"
  git config --global user.name "Ваше Имя"

для указания идентификационных данных аккаунта по умолчанию.
Пропустите параметр --global для указания данных только для этого репозитория.

fatal: не удалось выполнить автоопределение адреса электронной почты (получено «ndzakirov@ndzakirov.(none)»)
[ndzakirov@ndzakirov os-intro]$ git config --global user.name "Nurislam Zakirov"
[ndzakirov@ndzakirov os-intro]$ git config --global user.email "nuric35737@gmail.com"
[ndzakirov@ndzakirov os-intro]$ git commit -am 'feat(main): make course structure'
[master fded8d6] feat(main): make course structure
 2 files changed, 1 insertion(+), 14 deletions(-)
 delete mode 100644 package.json
[ndzakirov@ndzakirov os-intro]$ git push
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
При сжатии изменений используется до 3 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 293 байта | 293.00 КиБ/с, готово.
Всего 3 (изменений 1), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
```

Рис. 18: Отправка файлов на сервер

В ходе данной лабораторной работы я изучил идеологии и применения средств контроля версий, а также приобрел знания по работе с git.

## Список литературы

---

1. Лабораторная работа № 2 [Электронный ресурс] URL:  
<https://esystem.rudn.ru/mod/page/view.php?id=970819>