

# Лабораторная работа №2

## Операционные системы

---

Верниковская Е. А., НПИбд-01-23

27 февраля 2024

Российский университет дружбы народов, Москва, Россия

# Вводная часть

---

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

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.

# Выполнение лабораторной работы

---

Устанавливаем git, введя *dnf install git* (рис. 1)

```
[eavernikovskaya@eavernikovskaya ~]$ sudo dnf install git
[sudo] пароль для eavernikovskaya:
Попробуйте ещё раз.
[sudo] пароль для eavernikovskaya:
Fedora 39 - x86_64 - Updates                                19 kB/s | 20 kB    00:01
Fedora 39 - x86_64 - Updates                                1.9 MB/s | 4.6 MB  00:02
Последняя проверка окончания срока действия метаданных: 0:00:19 назад, Чт 22 фев 2024 16:44:14.
Пакет git-2.43.2-1.fc39.x86_64 уже установлен.
Зависимости разрешены.
Нет действий для выполнения.
Выполнено!
[eavernikovskaya@eavernikovskaya ~]$
```

Рис. 1: Установка git

# Установка программного обеспечения

Далее устанавливаем `gh`, введя команду `dnf install gh` (рис. 2)

```
teavernikovskaya@teavernikovskaya ~]$ sudo dnf install gh
Последняя проверка окончания срока действия метаданных: 0:01:28 назад, Чт 22 фев 2024 16:44:14.
Зависимости разрешены.
=====
Пакет                Архитектура          Версия                Репозиторий          Размер
=====
Установка:
gh                   x86_64               2.43.1-1.fc39        updates              9.1 М
=====
Результат транзакции
=====
Установка 1 Пакет

Объем загрузки: 9.1 М
Объем изменений: 46 М
Продолжить? [д/н]: y
Загрузка пакетов:
gh-2.43.1-1.fc39.x86_64.rpm                                4.8 MB/s | 9.1 MB  00:01
-----
Общий размер                                              3.9 MB/s | 9.1 MB  00:02
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверка транзакции
Тест транзакции проведен успешно.
Выполнение транзакции
Подготовка      :                               1/1
Установка      : gh-2.43.1-1.fc39.x86_64 1/1
Запуск скрипта : gh-2.43.1-1.fc39.x86_64 1/1
Проверка       : gh-2.43.1-1.fc39.x86_64 1/1
Установлен:
gh-2.43.1-1.fc39.x86_64

Выполнено!
teavernikovskaya@teavernikovskaya ~]$
```

Рис. 2: Установка `gh`

Задаём имя и email владельца репозитория (рис. 3)

```
[eavernikovskaya@eavernikovskaya ~]$ git config --global user.name "Katerok27153"  
[eavernikovskaya@eavernikovskaya ~]$ git config --global user.email "1132236136@pfur.ru"  
[eavernikovskaya@eavernikovskaya ~]$
```

**Рис. 3:** Имя и email



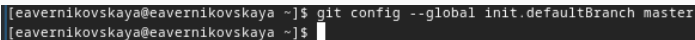
Настраиваем utf-8 в выводе сообщений git, с помощью команды *git config --global core.quotePath false* (рис. 4)



```
[eavernikovskaya@eavernikovskaya ~]$ git config --global core.quotePath false
[eavernikovskaya@eavernikovskaya ~]$
```

**Рис. 4:** Настройка utf-8

Далее задаём имя начальной ветки (будем называть её master), введя *git config --global init.defaultBranch master* (рис. 5)

A terminal window with a dark background. The prompt is [eavernikovskaya@eavernikovskaya ~]\$. The command entered is git config --global init.defaultBranch master. The prompt is shown again on the next line, indicating the command was executed successfully.

```
[eavernikovskaya@eavernikovskaya ~]$ git config --global init.defaultBranch master
[eavernikovskaya@eavernikovskaya ~]$
```

**Рис. 5:** Имя начальной ветки

Задаём параметры *autocrlf* и *safecrlf* (рис. 6)

```
[eavernikovskaya@eavernikovskaya ~]$ git config --global core.autocrlf input  
[eavernikovskaya@eavernikovskaya ~]$ git config --global core.safecrlf warn  
[eavernikovskaya@eavernikovskaya ~]$
```

**Рис. 6:** Параметры

Создаём ключ по алгоритму rsa размером 4096 бит (рис. 7)

```
[eavernikovskaya@eavernikovskaya ~]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/eavernikovskaya/.ssh/id_rsa):
Created directory '/home/eavernikovskaya/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/eavernikovskaya/.ssh/id_rsa
Your public key has been saved in /home/eavernikovskaya/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:8HPWdIWtGYCL+SIVCdPDqc6ThifjRVGqG8CuMr72d5s eavernikovskaya@eavernikovskaya
The key's randomart image is:
+---[RSA 4096]-----+
|      o=.o ... o. |
|      .oB .  o.. |
|      o  oo = .. .+ |
|      .. .oo+ .o .o |
|      . o= oS.o . |
|      . +oX .+. |
|+      ..* o . |
|oo      .. .. |
|.oo.. .E. |
+---[SHA256]-----+
[eavernikovskaya@eavernikovskaya ~]$
```

Рис. 7: Создание ключа SSH (1)

Создаём ключ по алгоритму ed25519 (рис. 8)

```
[eavernikovskaya@eavernikovskaya ~]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/eavernikovskaya/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/eavernikovskaya/.ssh/id_ed25519
Your public key has been saved in /home/eavernikovskaya/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:eJsIxAJVwvSreSaiQ5MNP+5Dk4PAcsa7IW9jmEPZpNU eavernikovskaya@eavernikovskaya
The key's randomart image is:
+--[ED25519 256]--+
|.++..|
|..+|
|...=|
|oo*E.|
|+o&.o.S|
|o&o=.o.o|
|*=Boo.o|
|*+*+|
|o*..|
+----[SHA256]-----+
[eavernikovskaya@eavernikovskaya ~]$
```

Рис. 8: Создание ключа SSH (2)

# Создание ZPGP ключа

- Генерируем ключ, с помощью команды `gpg --full-generate-key`
- Из предложенных опций выбираем тип RSA and RSA, размер 4096, срок действия - значение по умолчанию 0 (рис. 9)

```
[eavernikovskaya@eavernikovskaya ~]$ gpg --full-generate-key
gpg (GnuPG) 2.4.3; Copyright (C) 2023 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/eavernikovskaya/.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
Срок действия ключа? (0) 0
Срок действия ключа не ограничен
Все верно? (y/N) y
```

Рис. 9: Создание ключа PGP

# Настройка github

У меня уже есть аккаунт github, поэтому регистрироваться не надо (рис. 10)

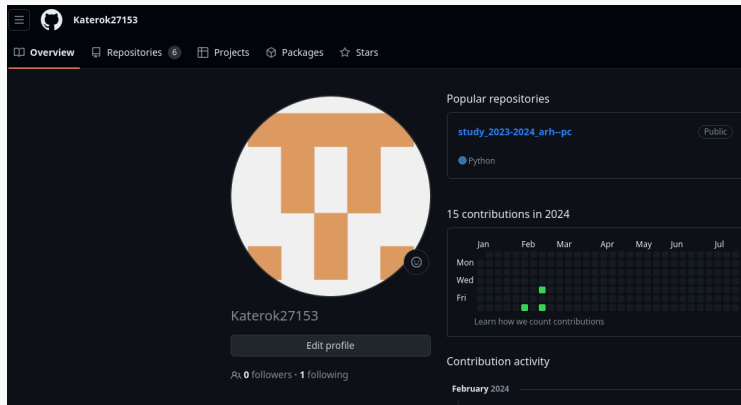


Рис. 10: Мой github

## Добавление PGP ключа в GitHub

Выводим список ключей и копируем отпечаток приватного ключа, введя *gpg --list-secret-keys --keyid-format LONG* (рис. 11)

```
[eavernikovskaya@eavernikovskaya ~]$ 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
[keyboxd]
-----
sec   rsa4096/47519BDC62A55FE7 2024-02-22 [SC]
      001AB0F571DA3173CB2CCCA647519BDC62A55FE7
uid           [ абсолютно ] VernikovskayaEkaterina <1132236136@pfur.ru>
ssb   rsa4096/69A1072281DCADFE 2024-02-22 [E]

[eavernikovskaya@eavernikovskaya ~]$
```

**Рис. 11:** Список ключей



## Добавление PGP ключа в GitHub

- Выводим наш сгенерированный ключ на экран, с помощью команды `gpg --armor --export`
- Копируем ключ (рис. 12)

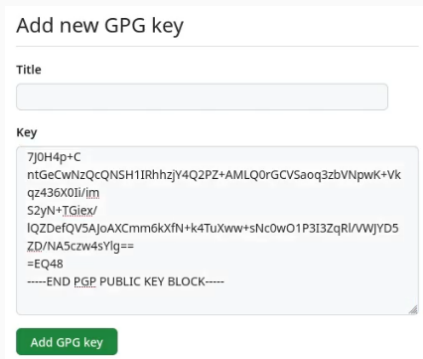
```
[eavernikovskaya@eavernikovskaya ~]$ gpg --armor --export 47519BDC62A55FE7
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGXXZ0sBEACo2jGJQVY4tEBTuw5IazhXVGdHsk0TjE4oQxE3WAAIUcgL3HbK
ZRKFccSjjgI06A+QIVcSJVfAsfqHjTJ59V9gX8pbvAvdKiKuUpDyHEMPB+LXxL4P
KcqM7mDH+7gpkLd3ivgUBjmPXFwMW3a0ZtnTm/qsbvcUvd3yfKMrEslp5XSKxo1M
20GP7e90JbY4xGjg/La67j/7bEvFnir2gTU+au/FtZjm+qnVmcuK5XQqr2bFkI1H
mQwgFANK1AStFqiQpdStUcuCE8cqy8y2LvcrJPaQ/hGECa++10txk0Za9CXws1e6
K2lwznK2y+OI f9IvG3CJkimbFpMZHesL5tEMERDo81ApqsiG/qic3cCQW14XnrtS
Fi4CkQJh0ncLM4g8Xx8WJFFRcTukar1pNisIn15Cu/Dw3H5qS2D4WAXTpQ0h2QnL
PI2ZL2VYAP9vrbnPZZ0BJrPr3a1Ev90rm5PEPsi1Is+7E/IgeXx06jRKq5ZGje7
NvOG94a48/1eJTJeWt0JBdYgdENaqXQhNFwwW9zqWL+dEBK20ft24XT/7gW1K7qq
cD4SYpSd5dtUN6xwyIaUdKFO0uPG7i1w518g032BSUX8p88UtinEKAf307Ve3xdI
7nELY6kGSHu2ID4GhxZtT0JsC4J3hopLwDE12JGqCN1H7B9Yn0h/AzTqHwARAQAB
tCtWZXJuaWtvdnNrYXlhRWthdGVyaW5hIDwxMTMyM2MTM2QHBmdXIucnU+IQJR
BBMBCAA7F1EEABqw9XHaMXPLLMymR1Gb3GK1X+cFAMXXZ0sCGwMFCwkIBwICIGIG
FQoJCA5CBBYCAwECHgcCF4AACgkQR1Gb3GK1X+eQLA/7BsBJAa+o5+He8AYZPg9j
```

Рис. 12: Ключ PGP (1)

# Добавление PGP ключа в GitHub

Переходим в настройки GitHub, нажимаем на кнопку New GPG key и вставляем полученный ключ в поле ввода (рис. 13)



Add new GPG key

Title

Key

```
7J0H4p+C
ntGeCwNzQcQNSH1IRhzhjY4Q2PZ+AMLQ0rGCVSaoq3zbVNPwK+Vk
qz436X0Ii/im
S2yN+TGieX/
IQZDefQV5AJoAXCmm6kXfN+k4TuXww+sNc0wO1P3I3ZqRI/VWJYD5
ZD/NA5czw4sYlg==
=EQ48
-----END PGP PUBLIC KEY BLOCK-----
```


Add GPG key

Рис. 13: Ключ PGP в поле ввода

### GPG keys

[New GPG key](#)

This is a list of GPG keys associated with your account. Remove any keys that you do not recognize.

GPG

**Email address:** 1132236136@pfur.ru

**Key ID:** 47519BDC62A55FE7

**Subkeys:** 69A1072281DCADFE

Added on Feb 22, 2024

[Delete](#)

Рис. 14: Ключ PGP в GitHub

# Настройка автоматических подписей коммитов git

Используя введённый email, указываем Git применять его при подписи коммитов (рис. 15)

```
[eavernikovskaya@eavernikovskaya ~]$ git config --global user.signingkey 47519BDC62A55FE7
[eavernikovskaya@eavernikovskaya ~]$ git config --global commit.gpgsign true
[eavernikovskaya@eavernikovskaya ~]$ git config --global gpg.program $(which gpg2)
[eavernikovskaya@eavernikovskaya ~]$
```

**Рис. 15:** Настройка автоматических подписей

- Для начала необходимо авторизоваться
- Отвечаем на несколько наводящих вопросов
- Авторизовываемся через браузер (рис. 16)

```
[eavernikovskaya@eavernikovskaya ~]$ gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations on this host? 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: 52B7-7458
Press Enter to open github.com in your browser...
█
```

Рис. 16: Авторизация

```
[eavernikovskaya@eavernikovskaya ~]$ gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations on this host? 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: 52B7-7458
Press Enter to open github.com in your browser...
✓ Authentication complete.
- gh config set -h github.com git_protocol https
✓ Configured git protocol
✓ Logged in as Katerok27153
```

Рис. 17: Завершение авторизации

- Создаём необходимые каталоги для рабочего пространства
- Используем команду `gh repo create study_2022-2023_os-intro --template=yamadharma/course-directory-student-template --public` (рис. 18)

```
[eavernikovskaya@eavernikovskaya ~]$ mkdir -p ~/work/study/2023-2024/"Операционные системы"
[eavernikovskaya@eavernikovskaya Операционные системы]$ gh repo create study_2023-2024_os-intro --template=yamadharma/course-directory-student-template --public
Created repository Katerok27153/study_2023-2024_os-intro on GitHub
https://github.com/Katerok27153/study_2023-2024_os-intro
[eavernikovskaya@eavernikovskaya Операционные системы]$
```

**Рис. 18:** Создание каталогов + использование gh

# Шаблон для рабочего пространства

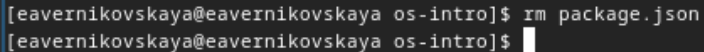
Далее клонируем репозиторий к себе в директорию, с помощью *git clone --recursive https://github.com//study\_2022-2023\_os-intro.git* (рис. 19)

```
[eavernikovskaya@eavernikovskaya Операционные системы]$ git clone --recursive https://github.com/Katerok27153/study_2023-2024_os-intro.git
Клонирование в «study_2023-2024_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 | 485.00 КиБ/с, готово.
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yamadharma/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharma/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/home/eavernikovskaya/work/study/2023-2024/Операционные системы/study_2023-2024_os-intro/template/presentation»...
remote: Enumerating objects: 95, done.
remote: Counting objects: 100% (95/95), done.
remote: Compressing objects: 100% (67/67), done.
remote: Total 95 (delta 34), reused 87 (delta 26), pack-reused 0
Получение объектов: 100% (95/95), 96.99 КиБ | 168.00 КиБ/с, готово.
Определение изменений: 100% (34/34), готово.
Клонирование в «/home/eavernikovskaya/work/study/2023-2024/Операционные системы/study_2023-2024_os-intro/template/report»...
remote: Enumerating objects: 126, done.
remote: Counting objects: 100% (126/126), done.
remote: Compressing objects: 100% (87/87), done.
remote: Total 126 (delta 52), reused 108 (delta 34), pack-reused 0
Получение объектов: 100% (126/126), 335.80 КиБ | 1.88 МБ/с, готово.
Определение изменений: 100% (52/52), готово.
Submodule path 'template/presentation': checked out '40a1761813e197d00e8443ff1ca72c60a304f24c'
Submodule path 'template/report': checked out '7c31ab8e5dfa8cdb2d67cae8a19ef8028ced88e'
```

Рис. 19: Клонирование репозитория



- Переходим в каталог курса `cd ~/work/study/2022-2023/“Операционные системы”/os-intro`
- Удаляем все лишние файлы, введя `rm package.json` (рис. 20)

A terminal window with a dark background. The prompt is [eavernikovskaya@eavernikovskaya os-intro]\$. The command rm package.json has been entered. The next line shows the prompt again with a cursor, indicating the command has been executed.

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

**Рис. 20:** Удаление лишних файлов

# Шаблон для рабочего пространства

Создаём необходимые каталоги введя команды:

1. *echo os-intro > COURSE* (рис. 21)
2. *make list, make prepare, make submodule*

```
[eavernikovskaya@eavernikovskaya os-intro]$ echo os-intro > COURSE
[eavernikovskaya@eavernikovskaya os-intro]$ make list
bash: make: команда не найдена
[eavernikovskaya@eavernikovskaya os-intro]$ make list
net-admin    Администрирование локальных сетей
net-os-admin Администрирование сетевых подсистем
arch-pc      Архитектура ЭВМ
sciprog-intro Введение в научное программирование
infosec      Информационная безопасность
computer-practice Компьютерный практикум по статистическому анализу данных
mathsec      Математические основы защиты информации и информационной безопасности
mathmod      Математическое моделирование
simulation-networks Моделирование сетей передачи данных
sciprog       Научное программирование
os-intro      Операционные системы
[eavernikovskaya@eavernikovskaya os-intro]$ make prepare
[eavernikovskaya@eavernikovskaya os-intro]$ make submodule
git submodule update --init --recursive
git submodule foreach 'git fetch origin; git checkout $(git rev-parse --abbrev-ref HEAD); git reset --hard origin/$(git rev-parse --abbrev-ref HEAD); git submodule update --recursive; git clean -dfx'
Entering 'template/presentation'
Указатель HEAD сейчас на коммите 40a1761 Merge branch 'release/1.0.3'
Entering 'template/report'
Указатель HEAD сейчас на коммите 7c3lab8 Merge branch 'release/1.0.4'
[eavernikovskaya@eavernikovskaya os-intro]$
```

Рис. 21: Создание необходимых каталогов

Отправляем файлы на сервер, с помощью *git add .*, *git commit -am* и *git push* (рис. 22)

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

**Рис. 22:** (git add .) + (git commit -am)

# Шаблон для рабочего пространства

```
[eavernikovskaya@eavernikovskaya os-intro]$ git push
Перечисление объектов: 39, готово.
Подсчет объектов: 100% (39/39), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (30/30), готово.
Запись объектов: 100% (38/38), 342.07 КиБ | 9.25 МиБ/с, готово.
Всего 38 (изменений 4), повторно использовано 1 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To https://github.com/Katerok27153/study_2023-2024_os-intro.git
    fd27d52..769626b  master -> master
[eavernikovskaya@eavernikovskaya os-intro]$
```

Рис. 23: git push

## **Выводы**

---

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