

Лабораторная работа №2

Операционные системы

Башиянц А. К.

05 марта 2025

Российский университет дружбы народов, Москва, Россия

Вводная часть

Цель данной работы — изучение идеологий и применение средств контроля версий, освоение умений по работе с git.

- Делать базовую настройку git;
- Работать с ключами;
- Настраивать и работать с github.

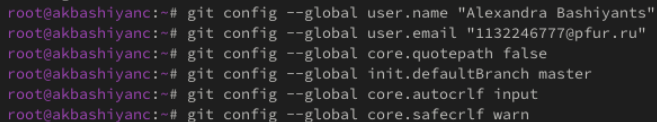
Выполнение лабораторной работы

```
root@akbashiync:~# dnf install git

Updating and loading repositories:
  Fedora 41 - x86_64 - Updates      100% | 21.1 KiB/s | 22.7 KiB | 00m01s
  Fedora 41 - x86_64 - Updates      100% | 1.5 MiB/s | 3.2 MiB | 00m02s
Repositories loaded.
Пакет "git-2.47.0-1.fc41.x86_64" уже установлен.
```

Рис. 1: Установка git

Сделаем базовые настройки git.

A terminal window with a dark background and light-colored text. It shows a series of five commands being executed in a shell, each preceded by a prompt 'root@akbashiyanc:~#'. The commands are for setting global git configuration: user.name, user.email, core.quotePath, init.defaultBranch, core.autocrlf, and core.safecrlf.

```
root@akbashiyanc:~# git config --global user.name "Alexandra Bashiyants"
root@akbashiyanc:~# git config --global user.email "1132246777@pfur.ru"
root@akbashiyanc:~# git config --global core.quotePath false
root@akbashiyanc:~# git config --global init.defaultBranch master
root@akbashiyanc:~# git config --global core.autocrlf input
root@akbashiyanc:~# git config --global core.safecrlf warn
```

Рис. 2: Настройка git

Создадим ключ по алгоритму rsa с размером 4096 бит.

```
root@akbashiyanc:~# ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:6traCsreILNKcT3AAPn5vT40ErZMKAAYlmg5v4CfzM root@akbashiyanc
The key's randomart image is:
+---[RSA 4096]-----+
|*B*|
|O. *|
|.o..o|
|ooo. o|
|+.*. .. S|
|o*oo. ..|
|.+.E ..|
|=oo+ Bo|
|+...+*+|
+----[SHA256]-----+
```

Рис. 3: Ключ ssh 4096

Создадим ключ по алгоритму ed25519.

```
root@akbashiyanc:~# 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:IwlnfbsoqK87DzPaKx7KfShWZrE9J9cxlo26ltawLP8 root@akbashiyanc
The key's randomart image is:
+--[ED25519 256]--+
|      .      |
|      o o . = |
|      . + . B o |
|      + o S + |
|      = = * + . |
|      .B o B B . |
|+==+ o O . |
|+=+BO+ =..E |
+----[SHA256]-----+
```

Рис. 4: Ключ ssh ed25519

```
Сбой при создании ключа: время истекло
root@akbashiyan:~# gpg --full-generate-key
gpg (GnuPG) 2.4.5; 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.

Выберите тип ключа:
(1) RSA and RSA
```

Рис. 5: Ключ pgp

```
42102E3EBB93272EFAE0ED2 :
открытый и секретный ключи созданы и подписаны.

pub   rsa4096 2025-02-27 [SC]
      90044EEAAB6D5F0A42102E3EBB93272EFAE0ED2
uid           Alexandra Bashiyants <1132246777@pfur.ru>
sub   rsa4096 2025-02-27 [E]
```

Рис. 6: Ключ pgp

Выведем список ключей и скопируем отпечаток приватного ключа.

```
akbashiya@akbashiya:~/work/study/2024-2025/Операционные системы$ 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/A230C2E6EAE3D2C9 2025-02-27 [SC]
      3514295B615767AFE48EABACA230C2E6EAE3D2C9
uid           [ абсолютно ] Alexandra Bashiyants <1132246777@pfur.ru>
ssb   rsa4096/7C1694B2DDFD776E 2025-02-27 [E]
```

Рис. 7: Список ключей

Используя введённый email, укажем Git применять его при подписи коммитов.

```
akbashiyc@akbashiyc:~/work/study/2024-2025/Операционные системы$ gpg --armor --export A230C2E6EAE3D2C9 | xclip -  
sel clip  
akbashiyc@akbashiyc:~/work/study/2024-2025/Операционные системы$ git config --global user.signingkey A230C2E6EAE  
3D2C9  
akbashiyc@akbashiyc:~/work/study/2024-2025/Операционные системы$ git config --global commit.gpgsign true  
akbashiyc@akbashiyc:~/work/study/2024-2025/Операционные системы$ git config --global gpg.program $(which gpg2)
```

Рис. 8: Подпись коммитов

```
akbashiync@akbashiync:~/work/study/2024-2025/Операционные системы$ gh auth login
? Where do you use GitHub? GitHub.com
? What is your preferred protocol for Git operations on this host? SSH
? Upload your SSH public key to your GitHub account? /home/akbashiync/.ssh/id_ed25519.pub
? Title for your SSH key: ssh3
? How would you like to authenticate GitHub CLI? Login with a web browser

! First copy your one-time code: 6623-7809
Press Enter to open https://github.com/login/device in your browser...
restorecon: SELinux: Could not get canonical path for /home/akbashiync/.mozilla/firefox/* restorecon: No such file or directory.
✓ Authentication complete.
- gh config set -h github.com git_protocol ssh
✓ Configured git protocol
✓ SSH key already existed on your GitHub account: /home/akbashiync/.ssh/id_ed25519.pub
✓ Logged in as AlexandraBashiyants
```

Рис. 9: Авторизация

```
! First copy your one-time code: AA7F-48F1
Open this URL to continue in your web browser: https://github.com/login/device
✓ Authentication complete.
! Authentication credentials saved in plain text
✓ Logged in as AlexandraBashiyants

root@akbashiync:~#
```

Рис. 10: Авторизация

Перейдем в необходимый каталог и сколируем каталог.

```
akbashiyan@akbashiyan:~/work/study/2024-2025/Операционные системы$ git clone --recursive git@github.com:AlexandraB  
ashiyants/study_2024-2025_os-intro.git os-intro  
  
Клонирование в «os-intro»...  
remote: Enumerating objects: 36, done.  
remote: Counting objects: 100% (36/36), done.  
remote: Compressing objects: 100% (35/35), done.  
remote: Total 36 (delta 1), reused 21 (delta 0), pack-reused 0 (from 0)  
Получение объектов: 100% (36/36), 19.37 Киб | 1024 байта/с, готово.  
Определение изменений: 100% (1/1), готово.
```

Рис. 11: git clone

Настроим каталог. Удалим лишние файлы, создадим необходимые файлы и закоммитим на сервер.d

```
akbashiyanc@akbashiyanc:~/work/study/2024-2025/Операционные системы/os-intro$ rm package.json  
akbashiyanc@akbashiyanc:~/work/study/2024-2025/Операционные системы/os-intro$ echo os-intro > COURSE  
akbashiyanc@akbashiyanc:~/work/study/2024-2025/Операционные системы/os-intro$ make prepare
```

Рис. 12: Работа с файлами

Выводы

- В этой лабораторной работе мы изучили работу системы контроля версий и git.