

Лабораторная работа №2

Операционные системы

Башиянц Александра Кареновна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Установка программного обеспечения	7
3.2	Базовая настройка git	7
3.3	Ключи ssh	7
3.4	Ключи pgr	8
3.5	Добавление PGP ключа в GitHub	9
3.6	Настройка автоматических подписей коммитов git	9
3.7	Настройка gh	10
3.8	Шаблон для рабочего пространства	10
4	Выводы	12

Список иллюстраций

3.1	Установка git	7
3.2	Настройка git	7
3.3	Ключ ssh 4096	8
3.4	Ключ ssh ed25519	8
3.5	Ключ pgp	9
3.6	Ключ pgp	9
3.7	Список ключей	9
3.8	Копирование ключа	9
3.9	Подпись коммитов	10
3.10	Авторизация	10
3.11	Авторизация	10
3.12	git clone	10
3.13	Работа с файлами	11
3.14	git push	11

Список таблиц

1 Цель работы

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

2 Задание

В этой лабораторной работе необходимо изучить работу виртуальной машины и ее настройки.

Необходимо научиться:

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

3 Выполнение лабораторной работы

3.1 Установка программного обеспечения

Установим git и gh (рис. 3.1).

```
root@akbashiya:~# 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. 3.1: Установка git

3.2 Базовая настройка git

Сделаем базовые настройки git (рис. 3.2).

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

Рис. 3.2: Настройка git

3.3 Ключи ssh

Создадим ключ по алгоритму rsa с размером 4096 бит (рис. 3.3).

```

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:
SHA256:6traCsreILNkcT3AAPn5vT40ErZMKAAy1mxg5v4CfzM root@akbashiyanc
The key's randomart image is:
+---[RSA 4096]-----+
|*B*                |
|O. *               |
|.o..o              |
|ooo. o             |
|+*. .. S           |
|o*oo. ..           |
|.+.E ..            |
|=oo+ Bo            |
|++..+*+            |
+---[SHA256]-----+

```

Рис. 3.3: Ключ ssh 4096

Создадим ключ по алгоритму ed25519 (рис. 3.4).

```

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 . =           |
|  . + . B o         |
|   + o S +          |
|  = = * + .         |
| .B o B B .         |
|+=+* o O .          |
|+=B0+ =..E          |
+---[SHA256]-----+

```

Рис. 3.4: Ключ ssh ed25519

3.4 Ключи pgr

Сгенерируем ключ (рис. 3.5-3.6).


```
Сбой при создании ключа! Время исчерпано
root@akbashiya:~# 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. 3.5: Ключ gpg

```
A42102E3EBB93272EFAE0ED2.FEV
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. 3.6: Ключ gpg

3.5 Добавление PGP ключа в GitHub

Выведем список ключей и скопируем отпечаток приватного ключа (рис. 3.7).

```
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, 0f, 1u
[keyboard]
-----
sec  rsa4096/A230C2E6EAE3D2C9 2025-02-27 [SC]
     3514295B615767AFE48EABCA230C2E6EAE3D2C9
uid           [ абсолютно ] Alexandra Bashiyants <1132246777@pfur.ru>
ssb  rsa4096/7C1694B2DDFD776E 2025-02-27 [E]
```

Рис. 3.7: Список ключей

Скопируем наш сгенерированный PGP ключ в буфер обмена (рис. 3.8).

```
gpg: Внимание! нечего экспортировать
root@akbashiya:~# gpg --armor --export sec EBB93272EFAE0ED2 | xclip -sel clip
root@akbashiya:~#
```

Рис. 3.8: Копирование ключа

3.6 Настройка автоматических подписей коммитов git

Используя введенный email, укажем Git применять его при подписи коммитов (рис. 3.9).

```

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

```

Рис. 3.9: Подпись коммитов

3.7 Настройка gh

Авторизируемся (рис. 3.10-3.11).

```

akbashiyanc@akbashiyanc:~/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/akbashiyanc/.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/akbashiyanc/.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/akbashiyanc/.ssh/id_ed25519.pub
✓ Logged in as AlexandraBashiyants

```

Рис. 3.10: 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@akbashiyanc:~#

```

Рис. 3.11: Авторизация

3.8 Шаблон для рабочего пространства

Перейдем в необходимый каталог и скопируем каталог (рис. 3.12).

```

akbashiyanc@akbashiyanc:~/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 Бита/с, готово.
Определение изменений: 100% (1/1), готово.

```

Рис. 3.12: git clone

Настроим каталог. Удалим лишние файлы, создадим необходимые файлы (рис. 3.13) и закоммитим на сервер (рис. 3.14)

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

Рис. 3.13: Работа с файлами

```
create mode 100644 project-personal/stages/report/report.md
akbashiya@akbashiya: ~/work/study/2024-2025/Операционные системы/os-intro$ git push
Перечисление объектов: 40, готово.
Подсчет объектов: 100% (40/40), готово.
Сжатие объектов: 100% (30/30), готово.
Запись объектов: 100% (38/38), 342.32 КиБ | 3.89 МБ/с, готово.
Total 38 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:AlexandraBashiyants/study_2024-2025_os-intro.git
 505774f..8bb05c4 master -> master
akbashiya@akbashiya: ~/work/study/2024-2025/Операционные системы/os-intro$
```

Рис. 3.14: git push

4 Выводы

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