

Содержание

| | |
|----------------------|---|
| лаборатория 2 | 1 |
| цель работы..... | 1 |
| Описание задачи..... | 1 |

лаборатория 2

David Michael Francis

цель работы

Изучить идеологию и применение средств контроля версий. Освоить умения по работе с git.

Описание задачи

1.во-первых, я установил git и gh, используя команды `dnf install git` и `dnf install gh`

```
and without modifying the system state.
midavid@vbox:~$ sudo -i
[sudo] password for midavid:
root@vbox:~# dnf install git
Updating and loading repositories:
  google-chrome           100% | 4.4 KiB/s | 1.3 KiB | 00m00s
  Copr repo for PyCharm owned by phracek 100% | 2.8 KiB/s | 2.1 KiB | 00m01s
  RPM Fusion for Fedora 41 - Nonfree - S 100% | 13.0 KiB/s | 8.7 KiB | 00m01s
  RPM Fusion for Fedora 41 - Nonfree - N 100% | 34.3 KiB/s | 9.1 KiB | 00m00s
  Fedora 41 - x86_64 - Updates             100% | 55.7 KiB/s | 25.1 KiB | 00m00s
  google-chrome           100% | 8.4 KiB/s | 3.2 KiB | 00m00s
  RPM Fusion for Fedora 41 - Nonfree - N 100% | 31.4 KiB/s | 18.2 KiB | 00m01s
  Fedora 41 - x86_64 - Updates             100% | 2.6 MiB/s | 7.2 MiB | 00m03s
Repositories loaded.
Package "git-2.48.1-1.fc41.x86_64" is already installed.

Nothing to do.
root@vbox:~# git config --global Ushie47 "Michael david"
error: key does not contain a section: Ushie47
root@vbox:~# dnf install gh
Updating and loading repositories:
Repositories loaded.
Package                               Arch   Version                               Repository   Size
Installing:
```

во-первых

2.Я ввел свое имя и адрес электронной почты. Затем я настроил utf-8 в выходных данных git-сообщений. Я настроил проверку и подписание git-коммита.Я также создал ssh-ключ

```

Complete!
root@vbox:~# git config --global user.name "Michael David"
root@vbox:~# git config --global user.email "ukanadaushie@gmail.com"
root@vbox:~# git config --global core.quotepath false
root@vbox:~# git config --global init.defaultBranch master
root@vbox:~# git config --global core.autocrlf input
root@vbox:~# git config --global core.safecrlf warn
root@vbox:~# 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 for "/root/.ssh/id_rsa" (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:PqD2qYvtkCIW5BI0DaVHiAJ4pbZmr8RB/YVlouyTDIE root@vbox
The key's randomart image is:
+---[RSA 4096]-----+
|==*+. . o          |
|E.*. . =          |
|o+*.+ . .          |
|o+.= o .          |
|.o= = o S          |
|.+.+ o o          |
|o.= +  o          |

```

Screenshot 2

3.Я сгенерировал ключ *pgp*, используя команду *gpg --full-generate-key*.

```

root@vbox:~# 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.

Please select what kind of key you want:
  (1) RSA and RSA
  (2) DSA and Elgamal
  (3) DSA (sign only)
  (4) RSA (sign only)
  (9) ECC (sign and encrypt) *default*
 (10) ECC (sign only)
 (14) Existing key from card
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072) 4096
Requested keysize is 4096 bits
Please specify how long the key should be valid.
    0 = key does not expire
    <n> = key expires in n days
    <n>w = key expires in n weeks
    <n>m = key expires in n months

```

Screenshot 3

4.Я добавил ключ *pgp* на *github*

```

We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: /root/.gnupg/trustdb.gpg: trustdb created
gpg: directory '/root/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/root/.gnupg/openpgp-revocs.d/0E9F96A906BAF43
16C1D8D28E27DFA4588FD67F8.rev'
public and secret key created and signed.

pub   rsa4096 2025-03-02 [SC]
       0E9F96A906BAF4316C1D8D28E27DFA4588FD67F8
uid           Michael <omotole47@gmail.com>
sub   rsa4096 2025-03-02 [E]

root@vbox:~# gpg --list-secret-keys --keyid-format LONG
gpg: checking the trustdb
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u
[keyboard]
-----
sec   rsa4096/E27DFA4588FD67F8 2025-03-02 [SC]

```

Screenshot 4

```

pub   rsa4096 2025-03-02 [SC]
       0E9F96A906BAF4316C1D8D28E27DFA4588FD67F8
uid           Michael <omotole47@gmail.com>
sub   rsa4096 2025-03-02 [E]

root@vbox:~# gpg --list-secret-keys --keyid-format LONG
gpg: checking the trustdb
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u
[keyboard]
-----
sec   rsa4096/E27DFA4588FD67F8 2025-03-02 [SC]
       0E9F96A906BAF4316C1D8D28E27DFA4588FD67F8
uid           [ultimate] Michael <omotole47@gmail.com>
ssb   rsa4096/29A1F451E7B997DF 2025-03-02 [E]

root@vbox:~# gpg --armor --export <PGP Fingerprint> | xclip -sel clip
-bash: syntax error near unexpected token `|'
root@vbox:~# sudo apt install xclip
sudo: apt: command not found
root@vbox:~# gpg --armor --export E27DFA4588FD67F8
bash: gpg: command not found...
root@vbox:~# gpg --armor --export E27DFA4588FD67F8
-----BEGIN PGP PUBLIC KEY BLOCK-----
mQINBGfE0ggBEADfTc6zw0vEx3DdEXV30EHWst176nKuMbSTZu5MdeoEgnTpRIAN


```

Screenshot 5

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 key
Email address: omotole47@gmail.com Unverified
Key ID: E27DFA4588FD67F8
Subkeys: 29A1F451E7B997DF
Added on Mar 2, 2025

[Delete](#)

Learn how to [generate a GPG key and add it to your account](#).

Screenshot 6

5. *Настройка подписей автоматической фиксации* `git config --global user.signingkey`
`git config --global commit.gpgsign true` `git config --global gpg.program $(which gpg2)`

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
root@vbox:~# git config --global user.signingkey <E27DFA4588FD67F8>
-bash: syntax error near unexpected token `newline'
root@vbox:~# git config --global user.signingkey <E27DFA4588FD67F8>
-bash: syntax error near unexpected token `newline'
root@vbox:~# git config --global user.signingkey E27DFA4588FD67F8
root@vbox:~# git config --global commit.gpgsign true
root@vbox:~# git config --global gpg.program $(which gpg2)
root@vbox:~# 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? /root/.ssh/id_ed25519.pub
? Title for your SSH key: Ssh key
? How would you like to authenticate GitHub CLI? Login with a web browser

! First copy your one-time code: AE5C-EFF6
Press Enter to open https://github.com/login/device in your browser...
Running Firefox as root in a regular user's session is not supported. ($XAUTHORITY
is /run/user/1000/.mutter-Xwaylandauth.ADVQ22 which is owned by midavid.)
✓ Authentication complete.
- gh config set -h github.com git_protocol ssh
✓ Configured git protocol
! Authentication credentials saved in plain text
✓ Uploaded the SSH key to your GitHub account: /root/.ssh/id_ed25519.pub
✓ Logged in as Ushie47
root@vbox:~# mkdir -p ~/study-2024-2025/"operations systems"
```

Screenshot 7

6. *Настройка каталога курсов*

```

root@vbox: ~/study-2024-2025_os-intro/operations_systems
root@vbox:~/study-2024-2025_os-intro/operations_systems# mkdir os-intro
root@vbox:~/study-2024-2025_os-intro/operations_systems# echo os-intro > COURSE
root@vbox:~/study-2024-2025_os-intro/operations_systems# git add .
root@vbox:~/study-2024-2025_os-intro/operations_systems# cd ~
root@vbox:~# git add .
fatal: not a git repository (or any of the parent directories): .git
root@vbox:~# cd study-2024-2025_os-intro
root@vbox:~/study-2024-2025_os-intro# git add .
root@vbox:~/study-2024-2025_os-intro# git commit -am 'feat(main): make course structure'
midavid@vbox:~$

```

Screenshot 8

Выход: В этой лабораторной работе мы проверили, как использовать git. Создаем репозитории и клонируем их. Как создать ssh- и pgp-ключи. А также как извлекать и отправлять в github

Ответы на контрольные вопросы

1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначены? *Системы контроля версий (VCS) отслеживают изменения в файлах, позволяя нескольким людям сотрудничать, вести историю и возвращаться к предыдущим версиям. Они решают такие проблемы, как случайная потеря данных, конфликты кода и отслеживание изменений.*

2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.

Хранилище: Центральное место, где хранятся все версии файлов. Фиксация: моментальный снимок изменений, сохраненных в репозитории. История: Запись коммитов, показывающая прошлые изменения. Рабочая копия: локальная версия файлов, изменяемых пользователем.

3. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.

Централизованная система контроля версий (CVCS): Все изменения хранятся в едином центральном репозитории. Пример: Subversion (SVN), Perforce.

Децентрализованная система контроля версий (DVCS): У каждого пользователя есть полная копия репозитория, что позволяет работать в автономном режиме. Пример: Git, Mercurial.

4. Действия VCS при работе с хранилищем в одиночку

Инициализируйте репозиторий. Вносите изменения и фиксируйте их. Просмотрите историю и при необходимости вернитесь к ней.

5. Работа с VCS Shared Storage

Clone/pull the latest version. Make changes and commit them locally. Push changes to the shared repository. Resolve conflicts if necessary.

6. Основные задачи, решаемые Git

Отслеживание изменений Совместная работа и слияние Ветвление для параллельной разработки Откат версии

7. Команды Git и их роли

Developers: Write and commit code. Maintainers: Review and merge changes. Release Managers: Manage versions and releases.

8. Использование локальных и удаленных репозиторий - Локальный репозиторий:

```
git init
git add .
git commit -m "Initial commit"
```

- Удаленный репозиторий:

```
git remote add origin <repo_url>
git push origin main
git pull origin main
```

9. Ветви и их назначение: Ветки позволяют вести параллельную разработку, не затрагивая основную кодовую базу.

```
git branch feature-branch
git checkout feature-branch
```

10. Игнорирование файлов в Git

```
node_modules/
*.log
```

This prevents unnecessary files from being committed.