

Лабораторная работа №2

Первоначальная настройка git

Комягин А.Н.

1 марта 2024

Российский университет дружбы народов, Москва, Россия

Изучить концепцию и применение средств контроля версий. Приобрести навыки работы с git.

- Создать базовую конфигурацию для работы с git
- Создать ключ SSH
- Создать ключ PGP
- Настроить подписи git
- Зарегистрироваться на Github
- Создать локальный каталог для выполнения заданий по предмету

Выполнение лабораторной работы

- УСТАНОВИМ *git*:

```
dnf install git
```

УСТАНОВКА *gh*

- Fedora:

```
dnf install gh
```

Настроим git

```
[ankomyagin@ankomyagin ~]$ git config --global user.name  
"andrey komyagin"  
[ankomyagin@ankomyagin ~]$ ^C  
[ankomyagin@ankomyagin ~]$ git config --global user.email  
"Komyagin12345@mail.ru"  
[ankomyagin@ankomyagin ~]$ git config --global core.quote  
path false  
[ankomyagin@ankomyagin ~]$ git config --global init.defau  
ltBranch master  
[ankomyagin@ankomyagin ~]$ git config --global core.autoc  
rlf input  
[ankomyagin@ankomyagin ~]$ git config --global core.safec  
rlf warn  
[ankomyagin@ankomyagin ~]$ █
```



Создадим ключи ssh

```
[ankomyagin@ankomyagin ~]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ankomyagin/.ssh/id_rsa):
Created directory '/home/ankomyagin/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ankomyagin/.ssh/id_rsa
Your public key has been saved in /home/ankomyagin/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:9m313eI51W95JTozMHeNeh7bchczV/J77a/E9HN6ZAU ankomyagin@ankomyagin
The key's randomart image is:
+---[RSA 4096]-----+
|
|             E |
|             . |
|             . o|
|      S      o+=|
|   . . o . . +=B@|
|   . +o +=*^ |
|   . *o +X% |
|   *B00 |
+-----[SHA256]-----+
[ankomyagin@ankomyagin ~]$
```

```
[ankomyagin@ankomyagin ~]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/ankomyagin/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ankomyagin/.ssh/id_ed25519
Your public key has been saved in /home/ankomyagin/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:uz3LZEWKf5tBKVd8Fk0UmyY2q+noeRwg70b6gwdgi3w ankomyagin@ankomyagin
The key's randomart image is:
+--[ED25519 256]--+
|
|      ==|
|      . =|
|      .+o+o|
|    o . o o.o=o |
| . o o S + =. |
| o E . = *o |
| . B =o+ |
| o @=+ + |
| +=*+o |
+-----[SHA256]-----+
[ankomyagin@ankomyagin ~]$
```

Создадим ключ gpg

```
[ankomyagin@ankomyagin ~]$ 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/ankomyagin/.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
```

```
GnuPG должен составить идентификатор пользователя для идентификации ключа.
```


Добавим ключ на Github

```
[ankomyagin@ankomyagin ~]$ git config --global  
user.signingkey Komyagin12345@mail.ru  
[ankomyagin@ankomyagin ~]$ git config --global  
commit.gpgsign true  
[ankomyagin@ankomyagin ~]$ git config --global  
commit.gpgProgram $(which gpg2)
```

Device Activation



Signed in as ANKomyagin

Enter the code displayed on your device

-

Continue

GitHub staff will never ask you to enter your code on this page.

```
? How would you like to authenticate GitHub CLI?
login with a web browser

! First copy your one-time code: 5892.0637
Press Enter to open github.com in your browser...
^C
[ankomyagin@ankomyagin ~]$
[ankomyagin@ankomyagin ~]$ gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations on this host? SSH
? Upload your SSH public key to your GitHub account?
t? /home/ankomyagin/.ssh/id_rsa.pub
? Title for your SSH key: sway
? How would you like to authenticate GitHub CLI?
login with a web browser

! First copy your one-time code: D389.D985
Press Enter to open github.com in your browser...
|
```

Создание репозитория

```
[ankomyagin@ankomyagin ~]$ mkdir -p ~/work/study/2023-2024/"Операционные системы"
[ankomyagin@ankomyagin ~]$ cd ~/work/study/2023-2024/Операционные\ системы/
[ankomyagin@ankomyagin Операционные системы]$ ^C
[ankomyagin@ankomyagin Операционные системы]$ gh repo create study_2023-2024_os-intro --template=yamadharma/course-directory-student-template --public
✓ Created repository ANKomyagin/study_2023-2024_os-intro on GitHub
https://github.com/ANKomyagin/study_2023-2024_os-intro
[ankomyagin@ankomyagin Операционные системы]$ git clone --recursive git@github.com:<owner>/study_2022-2023_os-intro.git os-intro
bash: owner: Нет такого файла или каталога
[ankomyagin@ankomyagin Операционные системы]$ git clone --recursive git@github.com:<owner>/study_2023-2024_os-intro.git os-intro
bash: owner: Нет такого файла или каталога
[ankomyagin@ankomyagin Операционные системы]$ git clone --recursive git@github.com:ANKomyagin/study_2023-2024_os-intro.git os-intro
Клонирование в «os-intro»...
The authenticity of host 'github.com (140.82.121.4)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvC0qU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? █
```

Настройка каталога и отправка на git

```
[ankomyagin@ankomyagin Операционные системы]$ cd ~/work/study/2023-2024/
Операционные\ системы/os-intro/
[ankomyagin@ankomyagin os-intro]$ rm package.json
[ankomyagin@ankomyagin os-intro]$ echo os-intro > COURSE
[ankomyagin@ankomyagin os-intro]$ make
Usage:
  make <target>

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

[ankomyagin@ankomyagin os-intro]$ make prepare
[ankomyagin@ankomyagin os-intro]$ git add .
[ankomyagin@ankomyagin os-intro]$ git commit -am 'feat(main): make cours
e structure'
```

Контрольные вопросы

1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначены?

Системы контроля версий (VCS) - это инструменты, которые помогают отслеживать изменения в исходном коде и управлять ими. Они предназначены для решения задач хранения истории изменений, совместной работы над проектами, отката к предыдущим версиям и т. д.

2. Объясните понятия VCS

- Хранилище (repository) - это место, где хранится история изменений проекта.
- Commit - это операция сохранения изменений в репозитории.
- История (history) - это список всех коммитов, которые были сделаны в проекте.
- Рабочая копия (working copy) - это каталог на компьютере разработчика, в котором он работает над проектом.

3. Что представляют собой и чем отличаются централизованные и децентрализованные VCS?

Централизованные VCS имеют единственный центральный сервер, к которому подключаются все разработчики (например, SVN).

Децентрализованные VCS позволяют каждому разработчику иметь полную копию репозитория, с которой он может работать независимо (например, Git).

4. Опишите действия с VCS при единоличной работе с хранилищем.

При единоличной работе с хранилищем в VCS разработчик делает изменения в своей рабочей копии проекта, коммитит их в локальный репозиторий и при необходимости откатывается к предыдущим версиям.

5. Опишите порядок работы с общим хранилищем VCS.

Порядок работы с общим хранилищем VCS включает получение изменений из удаленного репозитория, коммит изменений в локальный репозиторий и отправку изменений обратно в удаленный репозиторий.

6. Каковы основные задачи, решаемые инструментальным средством git?

Git предназначен для управления версиями файлов, совместной работы над проектами, отслеживания изменений и управления различными ветками разработки.

7. Назовите и дайте краткую характеристику командам git.

- `git add` - добавление файла в индекс (обычно точка).
- `git commit -m "Сообщение"` - создание коммита с описанием изменений.
- `git push` - отправка изменений в удаленный репозиторий.
- `git pull` - получение изменений из удаленного репозитория.

8. Приведите примеры использования при работе с локальным и удалённым репозиториями.

При работе с локальным репозиторием можно использовать команды `git add`, `git commit`.

При работе с удаленным репозиторием - `git clone`, `git push`, `git pull`.

9. Что такое и зачем могут быть нужны ветви (branches)?

Ветви (branches) в Git позволяют разрабатывать различные функциональности независимо друг от друга, а затем объединять изменения.

10. Как и можно игнорировать некоторые файлы при commit?

Для игнорирования некоторых файлов при коммите можно использовать файл `.gitignore`, в который записываются шаблоны файлов или папок, которые не должны попадать в репозиторий.

В ходе выполнения лабораторной работы я повторил правила работы с git, узнал о системе подписей и pgr ключах.