

Отчёт по лабораторной работе 1

Супонина Анастасия Павловна

Содержание

Цель работы

Ознакомиться с системой контроля версий git. Изучить базовые конфигурации и научиться вносить изменения в репозиторий при помощи функций git.

Задание

- 1) Зарегистрироваться на Github.
- 2) Создать базовую конфигурацию для работы с git.
- 3) Создать ключ SSH.
- 4) Создать ключ PGP.
- 5) Настроить подписи git.
- 6) Создать локальный каталог для выполнения заданий по предмету.

Теоретическое введение

Базовая настройка git

`git config --global user.name "Name Surname"` - задает имя владельца репозитория

`git config --global user.email "work@mail"` - задает email владельца репозитория

`git config --global core.quotePath false` - настройка utf-8 в выводе сообщений git

`git config --global init.defaultBranch master` - задает имя начальной ветки

`git config --global core.autocrlf input` - настройка параметра autocrlf

`git config --global core.safecrlf warn` - настройка параметра safecrlf

Команды для создания ключей ssh

`ssh-keygen -t rsa -b 4096` - по алгоритму rsa с ключём размером 4096 бит

`ssh-keygen -t ed25519` - по алгоритму ed25519

Команды для работы с ключами gpg

`gpg --full-generate-key` - генерация gpg ключа с настройками

gpg --list-secret-keys --keyid-format LONG - вывод списка ключей

gpg --armor --export | xclip -sel clip - копирование ключа в буфер обмена

Команды для настройки автоматических подписей коммитов git

git config --global user.signingkey

git config --global commit.gpgsign true

git config --global gpg.program \$(which gpg2)

Для настройки каталога курса

cd - переход в нужную папку

make - создание новой папки/файла

rm - удаление ненужной папки/файла

Отправка файлов на сервер:

git add .

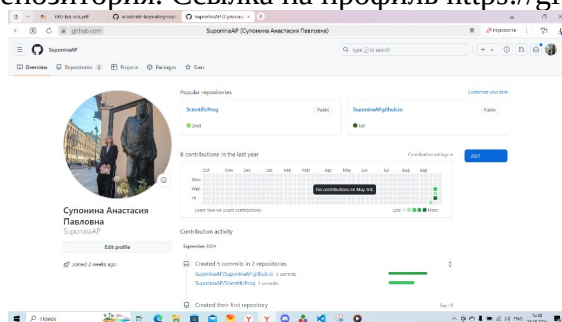
git commit -am 'feat(main): make course structure'

git push

Выполнение лабораторной работы

1) Зарегистрироваться на Github.

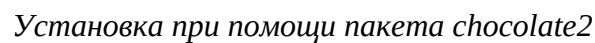
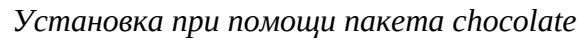
Создаю учетную запись на портале GitHub, для дальнейшей работы и загрузки репозитория. Ссылка на профиль <https://github.com/SuponinaAP/ScientificProg>



2) Создать базовую конфигурацию для работы с git.

Установка программного обеспечения

Устанавливаю программное обеспечение при помощи Chocolate, через команду в Windows PowerShell.



Провожу базовую настройку git при помощи команд указанных в теоретической части.

3) Создать ключ SSH.

При помощи команд создаю ключи SSH. - по алгоритму rsa

```

bermu@DESKTOP-RSEKJGC MSYS ~
$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/bermu/.ssh/id_rsa): SSHforGit
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in SSHforGit
Your public key has been saved in SSHforGit.pub
The key fingerprint is:
SHA256:bw/44LVDwN/rq5fhtVSLhkFuqjONvvu03IiqGRgZ3g bermu@DESKTOP-RSEKJGC
The key's randomart image is:
+---[RSA 4096]-----+
|
| ..
| . O...
| . O...
| . .++*O
| + E O,+* =
| . = O. .000|
| 000. .+O|
| O. O.+ O==|
| ...+B*Bo|
+---[SHA256]-----+

```

Ключ SSH

- по алгоритму ed25519

```

bermu@DESKTOP-RSEKJGC MSYS ~
$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/bermu/.ssh/id_ed25519): SSHforGit2
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in SSHforGit2
Your public key has been saved in SSHforGit2.pub
The key fingerprint is:
SHA256:HwJ51i1460d00PU3SrBCs54p8b7nb3aalC8nOQtTfu bermu@DESKTOP-RSEKJGC
The key's randomart image is:
+---[ED25519 256]--+
|
| . . . .
| O .O. . O
| O..000. OE
| .S00.O.+=
| +=+=. O=B|
| ..+=+.O.O|
| O + B.O |
| O+.***
+---[SHA256]-----+

```

Ключ SSH 2

4) Создать ключ PGP.

Генерирую gpg ключ, со следующими параметрами:

- тип RSA and RSA;
- размер 4096;

– выберите срок действия; значение по умолчанию— 0 (срок действия не истекает

```
bermu@DESKTOP-RSEKJGC MSYS ~
$ 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.

gpg: directory '/home/bermu/.gnupg' created
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 keysizes do you want? (3072) 4096
Requested keysizes 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
  <n>y = key expires in n years
Key is valid for? (0) 0
Key does not expire at all
Is this correct? (y/N) Y
```

никогда).

```
Real name: Anastasia
Email address: 1132249566@pfur.ru
Comment: Hello, world!
You selected this USER-ID:
  "Anastasia (Hello, world!) <1132249566@pfur.ru>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? 0
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: /home/bermu/.gnupg/trustdb.gpg: trustdb created
gpg: directory '/home/bermu/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/home/bermu/.gnupg/openpgp-revocs.d/AC9C0891B1237BF8272D7FFDAB8CC791C3AB1D9E.rev'
public and secret key created and signed.

pub  rsa4096 2024-09-19 [SC]
     AC9C0891B1237BF8272D7FFDAB8CC791C3AB1D9E
uid   rsa4096 2024-09-19 [E]
     Anastasia (Hello, world!) <1132249566@pfur.ru>
```

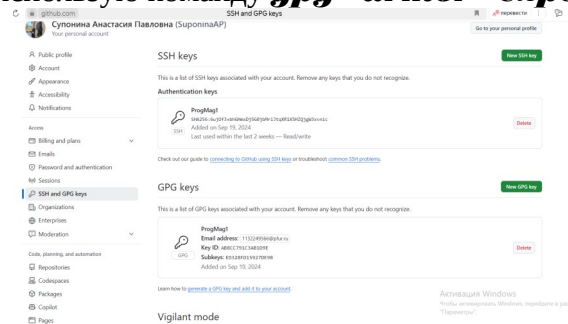
Далее захожу на гитхаб в настройки пользователя, вкладка ssh и gpg, и добавляю туда ранее сгенерированные ключи.

Ввожу команду для того чтобы получить отпечаток приватного ключа и с ним выполнить

```
bermu@DESKTOP-RSEKJGC MSYS ~
$ 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/AB8CC791C3AB1D9E 2024-09-19 [SC]
     AC9C0891B1237BF8272D7FFDAB8CC791C3AB1D9E
uid   [ultimate] Anastasia (Hello, world!) <1132249566@pfur.ru>
ssb  rsa4096/ED328FD15927DE98 2024-09-19 [E]
```

команду на его вывод.

Для копирования gpg ключа использую команду **gpg --armor --export** которая



выводит на экран сам ключ.

5) Настроить подписи git.

Используя программы из теоретической части, настраиваю автоматические подписи

```
bermu@DESKTOP-RSEKJGC MSYS ~
$ git config --global user.signingkey AB8CC791C3AB1D9E
bermu@DESKTOP-RSEKJGC MSYS ~
$ git config --global commit.gpgsign true
bermu@DESKTOP-RSEKJGC MSYS ~
$ git config --global gpg.program $(which gpg2)
which: no gpg2 in (/usr/local/bin:/usr/bin:/bin:/opt/bin:/c:/windows/System32:/c:/windows:/c:/windows/System32/wbem:/c:/windows/System32/windowsPowerShell/v1.0:/usr/bin/site_perl:/usr/bin/vendor_perl:/usr/bin/c
```

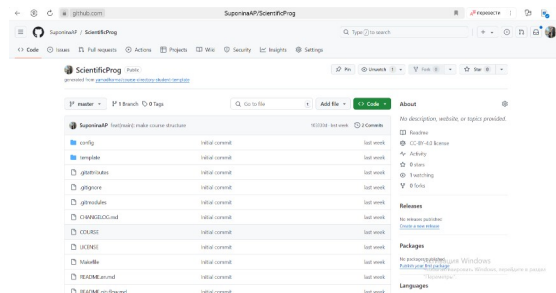
коммитов git.

6) Создать локальный каталог для выполнения заданий по предмету.

Так как работа производится через ОС Windows, через страницу гитхаб копирую себе репозиторий. При помощи msys2 сохраняю на компьютер репозиторий и удаляю файл package.gson, а также создаю необходимые каталоги.

Отправляю результаты обратно на сервер при помощи ранее упомянутых команд.

```
hermu@DESKTOP-RSEKJGC MSYS ~/work/study/2024-2025/ScientificProg/ScientificProg
$ git add .
hermu@DESKTOP-RSEKJGC MSYS ~/work/study/2024-2025/ScientificProg/ScientificProg
$ git commit -m 'feat(main): make course structure'
[master 1033304] feat(main): make course structure
1 file changed, 14 deletions(-)
delete mode 100644 package.json
```



Проверяю изменения зайдя на GitHub
Выводы

В процессе работы научилась пользоваться основными функциями git, ознакомилась с работой платформы GitHub, также изучила все базовые конфигурации и научилась вносить изменения пользуясь системой контроля версий git.

Список литературы

::: Пособие по лабораторной работе 1

{https://esystem.rudn.ru/pluginfile.php/2372898/mod_resource/content/5/002-lab_vcs.pdf} :::

Пособие по лабораторной работе 2

{https://esystem.rudn.ru/pluginfile.php/2372900/mod_resource/content/4/003-lab_markdown.pdf}