

презентация по лабораторной работе 1

Первоначальная настройка git

Хрусталеv В.Н.

Российский университет дружбы народов, Москва, Россия

Информация

- Хрусталев Влад Николаевич
- Студент ФМиЕН РУДН
- Группа НПИбд-02-22

Вводная часть

- Освоение git даёт широкий спектр возможностей для будущего его применения. Например, умение пользоваться GitHub-ом очень облегчает жизнь программисту.

- Изучить идеологию и применение средств контроля версий, в нашем случае git.

- Первичная настройка git
- Установка связи между аккаунтом GitHub и Linux
- Создание репозитория курса на основе шаблона

Первичная настройка git

Настройка параметров

После установки git, требуется настроить его. Для этого изменим(объявим) парочку глобальных переменных, как на скриншоте.

The screenshot shows a Kali Linux terminal window on the left and a web browser window on the right. The terminal window displays the output of the `dpkg-query -f='${Package} ${Version} ${Architecture}\n'` command, which lists installed packages. The output shows that `git` is installed. The terminal also shows the command `git config --global user.name "Name Surname"` and the output `git config --global user.email "work@mail"`. The terminal also shows the command `git config --global core.quotepath false` and the output `git config --global core.autocrlf input`. The terminal also shows the command `git config --global init.defaultBranch master` and the output `git config --global core.autocrlf input`. The terminal also shows the command `git config --global core.autocrlf input` and the output `git config --global core.autocrlf input`. The terminal also shows the command `git config --global core.autocrlf input` and the output `git config --global core.autocrlf input`.

The web browser window shows the URL `https://esystem.rudn.ru/mod/page/view.php?id=970680#orgc96c791`. The page title is "ОС_09.03.03: Лаборатория". The page content shows the following steps for creating an SSH key:

- Параметр `autocrlf`:
- Параметр `safecrif`:
- Параметр `safecrif`:

Создайте ключи `ssh`

- по алгоритму `rsa` с ключём размером 4096 бит:
- по алгоритму `ed25519`:

Создайте ключи `pgp`

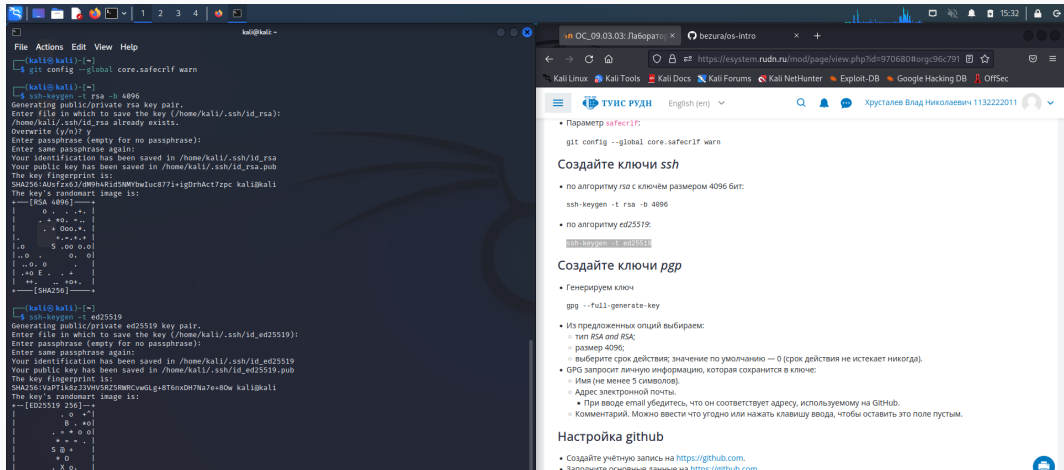
- Генерируем ключ
- Из предложенных опций выбираем:
- выберите срок действия; значение по умолчанию — 0 (срок действия не истекает никогда).
- GPG запросит личную информацию, которая сохранится в ключе:
- Имя (не менее 5 символов).
- Адрес электронной почты.
- При вводе email убедитесь, что он соответствует адресу, используемому на GitHub.
- Комментарий. Можно ввести что угодно или нажать клавишу ввода, чтобы оставить это поле пустым.

Установка связи между аккаунтом GitHub и Linux

Создание SSH ключей

Создадим два вида SSH ключей: rsa4096 и ed25519

Соответственно введя “ssh-keygen -t rsa -b 4096” и “ssh-keygen -t ed25519”



The image shows a Kali Linux terminal window on the left and a web browser window on the right. The terminal window displays the output of the following commands:

```
git config --global core.safecr1f warn
ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/kali/.ssh/id_rsa):
/home/kali/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/kali/.ssh/id_rsa
Your public key has been saved in /home/kali/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:AU5fzK0j/dM9HARid5NMWbIuc877i+ig0rhAct7zpc kali@kali
The key's randomart image is:
+--[RSA 4096]--+
|  . . . . .
|  . + + . . .
|  . + Ooo . .
|  . + . . . .
| 1.0 S .oo o.o
| ..0 . . 0.0
| ..0.0
| .+0 E . . +
| ++ .. +0+
+--[SHA256]--+
ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/kali/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/kali/.ssh/id_ed25519
Your public key has been saved in /home/kali/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:VaPtik8zJ3VHVSR25RwRCvwGLg+8T6nxDH7Na7e+80W kali@kali
The key's randomart image is:
+--[Ed25519 256]--+
|  . . . . .
|  . + + . . .
|  . + Ooo . .
|  . + . . . .
| 1.0 S .oo o.o
| ..0 . . 0.0
| ..0.0
| .+0 E . . +
| ++ .. +0+
+--[SHA256]--+
```

The web browser window shows the TUNIC RUDN website with the following content:

Создайте ключи ssh

- по алгоритму `rsa` с ключём размером 4096 бит:
`ssh-keygen -t rsa -b 4096`
- по алгоритму `ed25519`:
`ssh-keygen -t ed25519`

Создайте ключи `pgp`

- Генерируем ключ
`gpg --full-generate-key`
- Из предложенных опций выбираем:
 - тип `RSA` and `RSA`;
 - размер 4096;
 - выберите срок действия; значение по умолчанию — 0 (срок действия не истекает никогда).
- GPG запросит личную информацию, которая сохранится в ключе:
 - Имя (не менее 5 символов).
 - Адрес электронной почты.
 - При вводе email убедитесь, что он соответствует адресу, используемому на GitHub.
 - Комментарий. Можно ввести что угодно или нажать клавишу ввода, чтобы оставить это поле пустым.

Настройка github

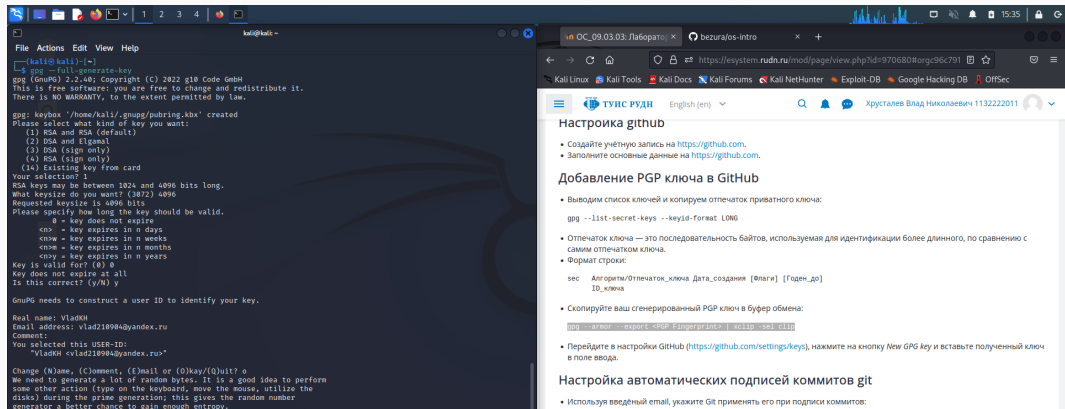
- Создайте учётную запись на <https://github.com>.
- Заполните основные данные на <https://github.com>.

Создание PGP ключей

Теперь создадим ключ PGP, введя команду “`gpg --full-generate-key`”.

Из предложенных опций выбираем RSA and RSA -> 4096 -> 0

Введём Имя, Адрес почты (как в ГитХабе) - чтобы завершить создание ключа



The screenshot shows a Kali Linux terminal window on the left and a web browser window on the right. The terminal window displays the output of the `gpg --full-generate-key` command, showing the selection of RSA and RSA (default) for the key type, 4096 bits for the key size, and 0 for the key expiration. The user's name is set to 'VladKH' and the email address is 'vlad210904@yandex.ru'. The browser window shows the GitHub 'Настройка github' (GitHub Settings) page, which includes instructions for adding a PGP key to GitHub. The page lists steps such as creating a GitHub account, adding a PGP key, and configuring automatic commit signatures.

```
gpg --full-generate-key
gpg (GnuPG) 2.2.40; Copyright (C) 2022 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: keybox '/home/kali/.gnupg/pubring.kbx' created
Please select what kind of key you want:
(1) RSA and RSA (default)
(2) DSA and Elgamal
(3) DSA (sign only)
(4) RSA (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
<W> = key expires in n weeks
<M> = key expires in n months
<Y> = key expires in n years
Key is valid for? (0) 0
Key does not expire at all
Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.

Real name: VladKH
Email address: vlad210904@yandex.ru
Comment:
You selected this USER-ID:
"VladKH <vlad210904@yandex.ru>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? o
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.
```

Настройка github

- Создайте учётную запись на <https://github.com>.
- Заполните основные данные на <https://github.com>.

Добавление PGP ключа в GitHub

- Выводим список ключей и копируем отпечаток приватного ключа:
`gpg --list-secret-keys --keyid-format LONG`
- Отпечаток ключа — это последовательность байтов, используемая для идентификации более длинного, по сравнению с самим отпечатком ключа.
- Формат строки:

sec	Алгоритм/Отпечаток_ключа	Дата_создания	{Флаги}	{Годен_до}
	ID_ключа			
- Скопируйте ваш сгенерированный PGP ключ в буфер обмена:
`gpg --armor --export <PGP_Fingerprint> | xclip -sel clip`
- Перейдите в настройки GitHub (<https://github.com/settings/keys>), нажмите на кнопку `New GPG key` и вставьте полученный ключ в поле ввода.

Настройка автоматических подписей коммитов git

- Используя введённый email, укажите Git применять его при подписи коммитов:

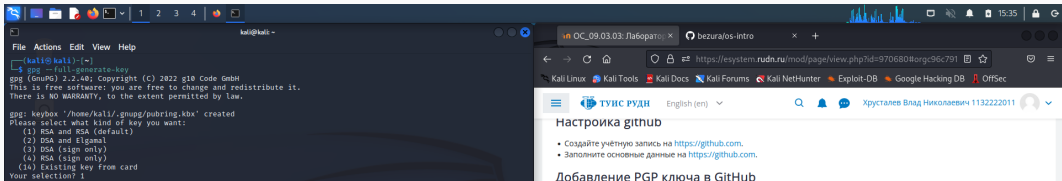
Добавление ключа в гитхаб

Введём “`gpg --list-secret-keys --keyid-format LONG`” - данная команда выведет отпечатки и др. информацию о GPG ключах в таком формате:

sec Алгоритм/Отпечаток_ключа Дата_создания [Флаги] [Годен_до] ID_ключа

Скопируем “Отпечаток_ключа” и вставим его в данной команде вместо -> “`gpg --armor --export`”

После этого отсаётся самое простое действие: Перейти в настройки GitHub (<https://github.com/settings/keys>), нажать на кнопку New GPG key и вставить полученный ключ в поле ввода.



Создание репозитория курса на
основе шаблона

Клонирование и скачивание репозитория

Введём данные команды создадим папку -> создадим клон репозитория -> скачаем его.

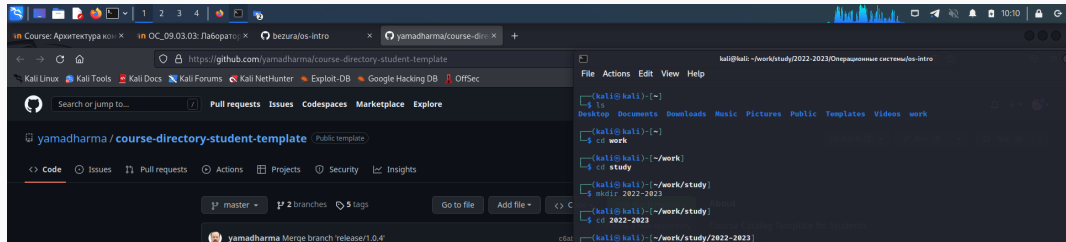
```
mkdir -p ~/work/study/2022-2023/“Операционные системы”
```

```
cd ~/work/study/2022-2023/“Операционные системы”
```

```
gh repo create study_2022-2023_os-intro
```

```
-template=yamadharm/course-directory-student-template -public
```

```
git clone -recursive git@github.com:/study_2022-2023_os-intro.git os-intro
```



Перейдем в каталог, удалим лишние файлы и создадим необходимые каталоги, введя команды: `“rm package.json” -> “echo os-intro > COURSE” -> “make”`.

И окончательный штрих, отправим изменённые файлы на сервер.

`“git add .” -> “git commit -am ‘feat(main): make course structure’” -> “git push”`

Итоги

- Таким образом мы сумели разобраться в основах Git и создать каталог для дальнейшей работы по Курсу