

Лабораторная работа №2

Операционные системы

Калашникова Ольга Сергеевна НПИбд-01-23

02 марта 2024

Российский университет дружбы народов, Москва, Россия

Цель работы

Целью данной работы является изучение идеологии и применение средств контроля версий, а так же освоение умения по работе с git.

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

Установка программного обеспечения

Установим git при помощи `dnf install git`, а так же установим gh при помощи `dnf install gh` (в моём случае программное обеспечение уже установлено, так как вышло не с первого раза) (рис. 1).

```
root
[oskalashnikova@oskalashnikova ~]$ sudo -i
[sudo] пароль для oskalashnikova:
[root@oskalashnikova ~]# dnf install git
Fedora 39 - x86_64 - Updates                20 kB/s | 20 kB    00:01
Fedora 39 - x86_64 - Updates                2.1 MB/s | 3.6 MB   00:01
Последняя проверка окончания срока действия метаданных: 0:00:09 назад, Ч
т 29 фев 2024 07:11:36.
Пакет git-2.43.2-1.fc39.x86_64 уже установлен.
Зависимости разрешены.
Нет действий для выполнения.
Выполнено!
[root@oskalashnikova ~]# dnf install gh
Последняя проверка окончания срока действия метаданных: 0:00:32 назад, Ч
т 29 фев 2024 07:11:36.
Пакет gh-2.43.1-1.fc39.x86_64 уже установлен.
Зависимости разрешены.
Нет действий для выполнения.
Выполнено!
```

Рис. 1: Установка git и gh

Зададим имя владельца репозитория при помощи `git config --global user.name "lacrimell"` и email при помощи `git config --global user.email "lacrimell@yandex.by"` (рис. 2).

```
[root@oskalashnikova ~]# git config --global user.name "lacrimell"
[root@oskalashnikova ~]# git config --global user.email "lacrimell@yandex.by"
[root@oskalashnikova ~]#
```

Рис. 2: Задаём имя и почту

Настроим utf-8 в выводе сообщений git при помощи `git config --global core.quotepath false` (рис.3).

```
[root@oskalashnikova ~]# git config --global core.quotepath false
[root@oskalashnikova ~]#
```

Рис. 3: utf-8

Зададим имя начальной ветки (будем называть её master) при помощи `git config --global init.defaultBranch master`, далее параметр `autocrlf` с помощью `git config --global core.autocrlf input` и параметр `safecrlf` при помощи `git config --global core.safecrlf warn` (рис.4).

```
[root@oskalashnikova ~]# git config --global init.defaultBranch master
[root@oskalashnikova ~]# git config --global core.autocrlf input
[root@oskalashnikova ~]# git config --global core.safecrlf warn
[root@oskalashnikova ~]#
```

Рис. 4: Настройка верификации

Создайте ключи ssh

Go алгоритму rsa с ключём размером 4096 бит создаём ключ ssh при помощи ssh-keygen -t rsa -b 4096, а по алгоритму ed25519 с ssh-keygen -t ed25519 (рис.5).

```
[root@oskalashnikova ~]# 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:A0qcztsy/EVhExHr5jRMtdup4fY12L3aav5AZDchPc root@oskalashnikova
The key's randomart image is:
+---[RSA 4096]-----+
|  o.o+  ..          |
|  .o= +.+          |
|  B = *...         |
|  + 0 = . E        |
|  o B o S o        |
|  . = + = o        |
|  o o o * = .      |
|  . = o.o          |
|  o+=.             |
+----[SHA256]-----+
[root@oskalashnikova ~]# 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:pPu8jqLb5paQVkwWnnf5t1WpnDjES/X3oJiFwUuxKMIQ root@oskalashnikova
The key's randomart image is:
+---[ED25519 256]---+
|  +o ...           |
```


Создайте ключи pgp

Генерируем ключ при помощи gpg --full-generate-key и выбираем тип RSA and RSA, размер 4096, срок действия не истекает никогда (рис.6).

```
[root@oskalashnikova ~]# 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.

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.

Ваше полное имя: lacrimell
Адрес электронной почты: lacrimell@yandex.by
Примечание: 123cake!
Вы выбрали следующий идентификатор пользователя:
  "lacrimell (123cake!) <lacrimell@yandex.by>"

Сменить (N)Имя (C)Примечание (E)Адрес: (O)Получить/(O)Выход? N
```

Создайте ключи pgp

Вводим имя, адрес почты используемый на GitHub.Я так же ввела комментарий, чтобы не забыть пароль (рис. 7).

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

Ваше полное имя: lacrimell
Адрес электронной почты: lacrimell@yandex.by
Примечание: 123cake!
Вы выбрали следующий идентификатор пользователя:
"lacrimell (123cake!) <lacrimell@yandex.by>"

Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? o
Необходимо получить много случайных чисел. Желательно, чтобы Вы
в процессе генерации выполняли какие-то другие действия (печать
на клавиатуре, движения мыши, обращения к дискам); это даст генератору
случайных чисел больше возможностей получить достаточное количество энтропии.
Необходимо получить много случайных чисел. Желательно, чтобы Вы
в процессе генерации выполняли какие-то другие действия (печать
на клавиатуре, движения мыши, обращения к дискам); это даст генератору
случайных чисел больше возможностей получить достаточное количество энтропии.
gpg: сертификат отзыва записан в '/root/.gnupg/openpgp-revocs.d/46AD87DC
366EA410C2A48EFBC9BF719AA2931FD4.rev'.
открытый и секретный ключи созданы и подписаны.

pub   rsa4096 2024-02-29 [SC]
       46AD87DC366EA410C2A48EFBC9BF719AA2931FD4
uid           lacrimell (123cake!) <lacrimell@yandex.by>
sub   rsa4096 2024-02-29 [E]

[root@oskalashnikova ~]#
```

Рис. 7: Генерация ключа pgp 2

Добавление PGP ключа в GitHub

У меня уже была создана учетная запись и заполнены основные данные на <https://github.com>, так что я перешла сразу к выводу список ключей при помощи `gpg --list-secret-keys --keyid-format LONG` (рис. 8).

```
[root@oskalashnikova ~]# 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: 2  подписанных: 0  доверие: 0-, 0q, 0n
, 0m, 0f, 2u
[keyboard]
-----
sec   rsa4096/C9BF719AA2931FD4 2024-02-29 [SC]
      46AD87DC366EA410C2A48EFBC9BF719AA2931FD4
uid   [ абсолютно ] lacrimell (123cake!) <lacrimell@yandex.
by>
ssb   rsa4096/A04E6263CAA5EA 2024-02-29 [E]

sec   rsa4096/E92D0B168D9F3929 2024-02-27 [SC]
      E021F68B5D6169D503FC39F8E92D0B168D9F3929
uid   [ абсолютно ] lacrimell (cake) <lacrimell@yandex.by>
ssb   rsa4096/F058C18BC87FD728 2024-02-27 [E]
```

Рис. 8: Список ключей

Добавление PGP ключа в GitHub

Далее мы должны скопировать сгенерированный PGP ключ в буфер обмена при помощи `gpg --armor --export | xclip -sel clip`, но данная команда у меня не сработала и я копировала вручную (рис.9).

```
[root@oskalashnikova ~]# gpg --armor --export 46AD87DC366EA410C2A48EFBC9
BF719AA2931FD4
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGXgB68BEAC/Psa0y9yotE8Hh/QgJaxnZJ9FRf0rEeaSo+sr1MAgwB6ZbYNK
oUw4A4QGs8CXTMbUsE38XoE07xnQzkwsyQwC2NTDvZ+ETQ0E0y/K++8Fb00bNie7
7hAlpjwWkJEh+/tovj51ZWY2cxe9T07j+r5wNb8t1ujjkhdYZvegYXagRdf0zrR
xDHYZ0fEmVXMYBYCdQ/9+HxtVqG5pnk2UbZcasqD3XrV7igtVziwsESdr1n/6yMk
pnESuXKhbYkMPbDzhK4wcgeKm2GaelT/7AAhCb6S6PrPhAHgv1HDp+e7e1wNjkuD
1++rVHZG0oFmbKXYk6gX1/ZhRjxi3UPEWGCkj6b0vLZcgyrR0xFvpWobHF69B69K
8fwyxfuwZa1IGiT2dN9nU1+Dv9SLDo3UnQfsE2Y1Ts9fwue1cvh+vRwW/BYsyNUj
c7AHT8B3chHVra1MvaB0U0B1chuyvqE077n8197NcdFERfVhKwLXtUkt1utpNth
```

Рис. 9: Копирование ключа

Добавление PGP ключа в GitHub

Переходим в настройки GitHub (<https://github.com/settings/keys>), нажимаем на кнопку New GPG key и вставляем полученный ключ в поле ввода (рис.10).

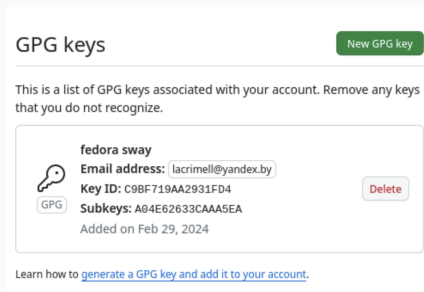


Рис. 10: Готовый ключ

##Настройка автоматических подписей коммитов git

Используя свой email, указываем Git применять его при подписи

Авторизируемся при помощи `gh auth login`. Утилита задаёт несколько наводящих вопросов, после авторизируемся через браузер (рис.12).

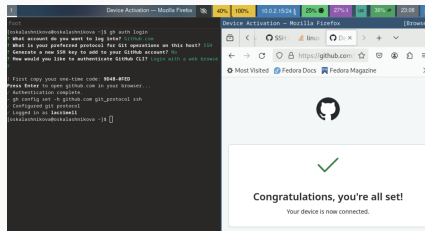


Рис. 12: Авторизация

Создаём папку при помощи `mkdir -p ~/work/study/2023-2024/“Операционные системы”` и переходим в неё `cd ~/work/study/2023-2024/“Операционные системы”` (рис.13).

A terminal window with a dark background and light blue text. It shows two commands being executed. The first command creates a directory, and the second command changes the current directory to the newly created one. The prompt is [oskalashnikova@oskalashnikova ~].

```
[oskalashnikova@oskalashnikova ~]$ mkdir -p ~/work/study/2023-2024/"Операционные системы"  
[oskalashnikova@oskalashnikova ~]$ cd ~/work/study/2023-2024/"Операционные системы"
```

Рис. 13: Создание папки

Создаём репозиторий `gh repo create study_2023-2024_os-intro --template=yamadharm/course-directory-student-template --public` (рис.14).

```
[oskalashnikova@oskalashnikova Операционные системы]$ gh repo create study_2023-2024_os-intro --template=yamadharm/course-directory-student-template --public
✓ Created repository lacrimell/study_2023-2024_os-intro on GitHub
https://github.com/lacrimell/study_2023-2024_os-intro
```

Рис. 14: Создание репозитория

Клонируем его на виртуальную машину `git clone --recursive git@github.com:oskalashnikova/study_2023-2024_os-intro.git os-intro` (рис.15).

```
[oskalashnikova@oskalashnikova Операционные системы]$ git clone --recursive git@github.com:lacrimell/study_2023-2024_os-intro.git
Клонирование в «study_2023-2024_os-intro»...
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:+D1Y3wvvV6TuJJhbpZ1sF/zLDA0zPMSvHdkr4U
```

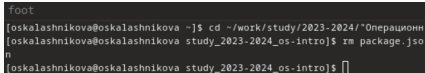
Рис. 15: Клонирование

Создание репозитория курса на основе шаблона

Проверяем с помощью `ls` (рис.16).

Левая панель		Файл	Команда	
<-- ...dy_2023-2024_os-intro --.[^]>				
.и	Имя	Размер	Время	правки
/..		-BBERX-	мар	1 08:20
/.	.git	152	мар	1 08:20
/	config	24	мар	1 08:20
/	template	36	мар	1 08:20
.	.gitat~butes	1765	мар	1 08:20
.	.gitignore	4637	мар	1 08:20
.	.gitmodules	278	мар	1 08:20
.	CHANGELOG.md	4507	мар	1 08:20
.	COURSE	0	мар	1 08:20
.	LICENSE	18657	мар	1 08:20
.	Makefile	980	мар	1 08:20
.	README.en.md	152	мар	1 08:20
.	README~ow.md	5653	мар	1 08:20
.	README.md	4304	мар	1 08:20
.	package.json	401	мар	1 08:20

Переходим в каталог курса при помощи `cd ~/work/study/2022-2023/“Операционные системы”/os-intro` и удаляем лишние файлы с помощью `rm package.json` (рис.17).

A terminal window with a dark background. The prompt is 'foot'. The first command is '[oskalashnikova@oskalashnikova ~]\$ cd ~/work/study/2023-2024/“Операционн'. The second command is '[oskalashnikova@oskalashnikova study_2023-2024_os-intro]\$ rm package.json'. The output is 'n'. The final prompt is '[oskalashnikova@oskalashnikova study_2023-2024_os-intro]\$ ' with a cursor.

```
foot
[oskalashnikova@oskalashnikova ~]$ cd ~/work/study/2023-2024/“Операционн
[oskalashnikova@oskalashnikova study_2023-2024_os-intro]$ rm package.json
n
[oskalashnikova@oskalashnikova study_2023-2024_os-intro]$ 
```

Рис. 17: Удаление файла

Настройка каталога курса

Проверяем, удалился ли файл (рис.18).

foot				
Левая панель		Файл	Команда	
<- ...dy_2023-2024_os-intro			. [^]>	
.и	Имя	Размер	Время	правки
./..		-BBEPX-	map	1 08:20
/.git		152	map	1 08:20
/config		24	map	1 08:20
/template		36	map	1 08:20
.gitat~butes		1765	map	1 08:20
.gitignore		4637	map	1 08:20
.gitmodules		278	map	1 08:20
CHANGELOG.md		4507	map	1 08:20
COURSE		0	map	1 08:20
LICENSE		18657	map	1 08:20
Makefile		980	map	1 08:20
README.en.md		152	map	1 08:20
README~ow.md		5653	map	1 08:20
README.md		4304	map	1 08:20

Создаём необходимые каталоги при помощи `echo os-intro > COURSE` (рис.19).

```
[oskalashnikova@oskalashnikova study_2023-2024_os-intro]$ echo os-intro  
> COURSE  
[oskalashnikova@oskalashnikova study_2023-2024_os-intro]$
```

Рис. 19: Создание

Проверяем созданся ли (рис.20).

.gitignore	1037	map	1 08:20
.gitmodules	278	map	1 08:20
CHANGELOG.md	4507	map	1 08:20
COURSE	9	map	1 08:33
LICENSE	18657	map	1 08:20
Makefile	980	map	1 08:20

Рис. 20: Проверка

Настройка каталога курса

Используем make (рис.21).

```
[oskalashnikova@oskalashnikova study_2023-2024_os-intro]$ make
Usage:
  make <target>

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

[oskalashnikova@oskalashnikova study_2023-2024_os-intro]$ make list
  net-admin    Администрирование локальных сетей
  net-os-admin Администрирование сетевых подсистем
  arch-pc      Архитектура ЭВМ
  sciprogram   Введение в научное программирование
  infosec      Информационная безопасность
  computer-practice Компьютерный практикум по статистическому анализу д
аннх
  mathsec      Математические основы защиты информации и информаци
онной безопасности
  mathmod      Математическое моделирование
  simulation-networks Моделирование сетей передачи данных
  sciprogram   Научное программирование
  os-intro     Операционные системы

[oskalashnikova@oskalashnikova study_2023-2024_os-intro]$ make prepare
[oskalashnikova@oskalashnikova study_2023-2024_os-intro]$ make submodule
git submodule update --init --recursive
git submodule foreach 'git fetch origin; git checkout $(git rev-parse --
abbrev-ref HEAD); git reset --hard origin/$(git rev-parse --abbrev-ref H
EAD); git submodule update --recursive; git clean -dfx'
Entering 'template/presentation'
Указатель HEAD сейчас на коммите 40a1761 Merge branch 'release/1.0.3'
Entering 'template/report'
Указатель HEAD сейчас на коммите 7c31ab8 Merge branch 'release/1.0.4'
[oskalashnikova@oskalashnikova study_2023-2024_os-intro]$
```

Рис. 21: make

Отправляем файлы на сервер 1 (рис.22).

```
[oskalashnikova@oskalashnikova study_2023-2024_os-intro]$ git add .  
[oskalashnikova@oskalashnikova study_2023-2024_os-intro]$ git commit -am '  
feat(main): make course structure'  
Текущая ветка: master  
Ваша ветка опережает «origin/master» на 1 коммит.  
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
```

Рис. 22: git add . , git commit

Отправляем файлы на сервер 2 (рис.23).

```
[oskalashnikova@oskalashnikova study_2023-2024_os-intro]$ git push
Username for 'https://github.com': lacrimell
Password for 'https://lacrimell@github.com':
Перечисление объектов: 40, готово.
Подсчет объектов: 100% (40/40), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (30/30), готово.
Запись объектов: 100% (38/38), 341.45 КиБ | 10.67 Миб/с, готово.
Всего 38 (изменений 4), повторно использовано 0 (изменений 0), повторно
использовано пакетов 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To https://github.com/lacrimell/study_2023-2024_os-intro.git
  4fc3003..9459418  master -> master
[oskalashnikova@oskalashnikova study_2023-2024_os-intro]$
```

Рис. 23: git push

Я изучила идеологию и применение средств контроля версий, атак же освоила умения по работе с git

Список литературы

<https://git-scm.com/book/ru/v2/Основы-Git-Работа-с-удалёнными-репозиториями>

<https://devpractice.ru/git-for-beginners-part-1-what-is-vcs/>

<https://blog.skillfactory.ru/glossary/git/>

туис