

Отчёт по лабораторной работе №3

Операционные системы

Сячинова Ксения Ивановна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	13

Список иллюстраций

3.1	Создание учётной записи	7
3.2	Делаем конфигурацию	7
3.3	Делаем конфигурацию	8
3.4	Настройка git	8
3.5	Создание ssh ключа	9
3.6	Создание gpg ключа	10
3.7	Добавление ключей	10
3.8	Полученные ключи	11
3.9	Настройка коммитов	11
3.10	Содание репозитория	12
3.11	Итог	12

Список таблиц

1 Цель работы

Цель работы: научиться оформлять отчёты с помощью легковесного языка разметки Markdown, а также познакомиться с основными возможностями разметки Markdown.

2 Задание

Сделайте отчёт по предыдущей лабораторной работе в формате Markdown. В качестве отчёта просьба предоставить отчёты в 3 форматах: pdf, docx и md (в архиве, поскольку он должен содержать скриншоты, Makefile и т.д.)

3 Выполнение лабораторной работы

1) Создаём учётную запись на github.(рис. 3.1)

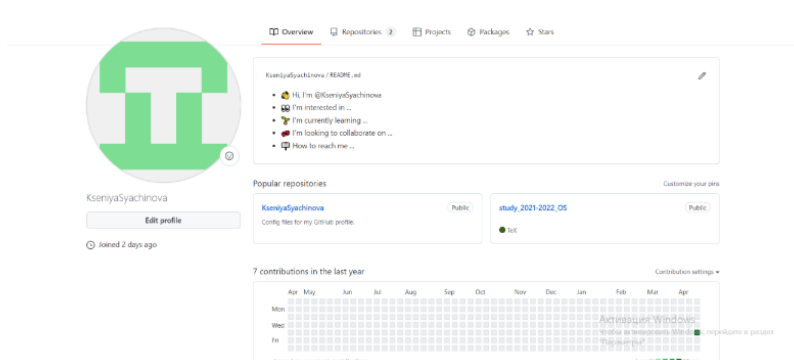


Рис. 3.1: Создание учётной записи

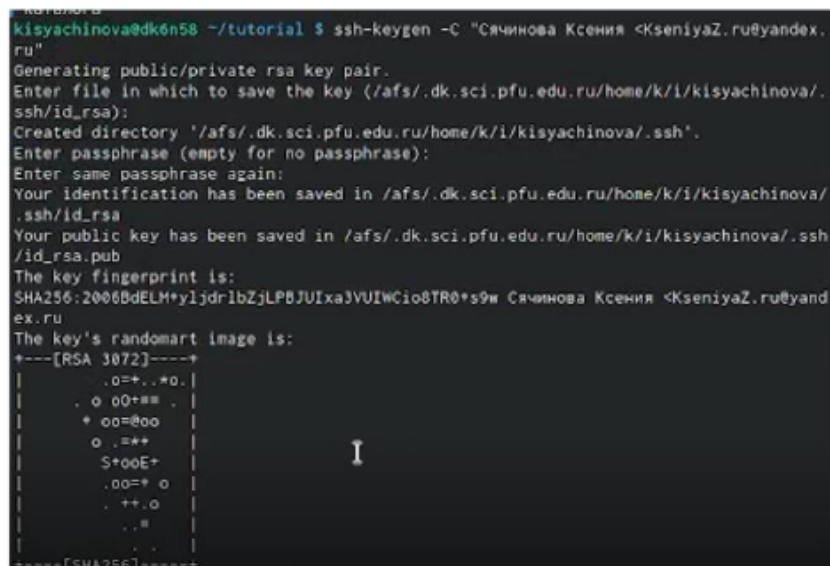
2) Сделаем предварительную конфигурацию, указав имя и email владельца репозитория с помощью `git config --global user.name "Имя Фамилия"`, `git config --global user.email "work@mail"`. (рис. 3.2)

```
kisyachinova@dk6n58 ~ $ git config --global user.name "Ksedniys Sya  
chinova"  
kisyachinova@dk6n58 ~ $  
kisyachinova@dk6n58 ~ $ git config -- global user.email "KseniyaZ.  
ru@yandex.ru"  
fatal: not in a git directory  
kisyachinova@dk6n58 ~ $ git config --global user.email "KseniyaZ.ru  
@yandex.ru"
```

Рис. 3.2: Делаем конфигурацию

3) После этого создаём новый ключ на github (команда `ssh-keygen -C "KseniyaSyachinova KseniyaZ.ru@yandex.ru"` и привязываем его к компьютеру)

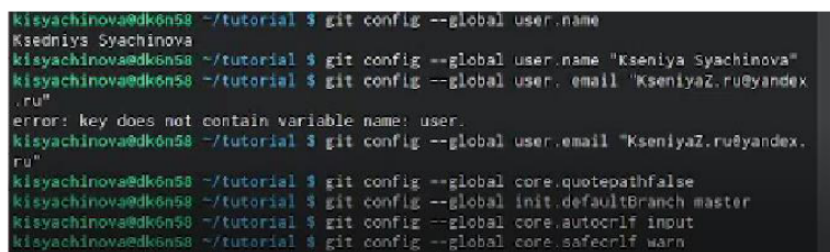
ру через консоль. После этого, скопировав из локальной консоли ключ в буфер обмена, вставляем ключ в появившееся на сайте поле. (рис. 3.3)



```
kisyachinova@dk6n58 ~/tutorial $ ssh-keygen -C "Сячинова Ксения <KseniyaZ.ru@yandex.ru>"
Generating public/private rsa key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/k/i/kisyachinova/.ssh/id_rsa):
Created directory '/afs/.dk.sci.pfu.edu.ru/home/k/i/kisyachinova/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /afs/.dk.sci.pfu.edu.ru/home/k/i/kisyachinova/.ssh/id_rsa
Your public key has been saved in /afs/.dk.sci.pfu.edu.ru/home/k/i/kisyachinova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:2006BdELM+yldrlbZjLPBJUia3VUIWcio8TR0*s9w Сячинова Ксения <KseniyaZ.ru@yandex.ru>
The key's randomart image is:
+---[RSA 3072]---+
|
|.o=+.o.
|o o0+=.
|+o0=@oo
|o .==+
|S+ooE+
|.oo=+ o
|. ++.o
|..#
|..
+---[SHA256]-----+
```

Рис. 3.3: Делаем конфигурацию

- 4) Приступаем к базовой настройке git. Зададим имя и email владельца репозитория: `git config --global user.name "Name Surname"`, `git config --global user.email "work@mail"`. Настроим utf-8 в выводе сообщений git: `git config --global core.quotepath false`. Настроим верификацию и подписание коммитов git. Зададим имя начальной ветки: `git config --global init.defaultBranch master`. Параметр `autocrlf`: `git config --global core.autocrlf input`. Параметр `safecrlf`: `git config --global core.safecrlf warn`. (рис. 3.4)



```
kisyachinova@dk6n58 ~/tutorial $ git config --global user.name Kseniya Syachinova
kisyachinova@dk6n58 ~/tutorial $ git config --global user.name "Kseniya Syachinova"
kisyachinova@dk6n58 ~/tutorial $ git config --global user.email "KseniyaZ.ru@yandex.ru"
error: key does not contain variable name: user.
kisyachinova@dk6n58 ~/tutorial $ git config --global user.email "KseniyaZ.ru@yandex.ru"
kisyachinova@dk6n58 ~/tutorial $ git config --global core.quotepath false
kisyachinova@dk6n58 ~/tutorial $ git config --global init.defaultBranch master
kisyachinova@dk6n58 ~/tutorial $ git config --global core.autocrlf input
kisyachinova@dk6n58 ~/tutorial $ git config --global core.safecrlf warn
```

Рис. 3.4: Настройка git

- 5) Затем создаём свой ключ по алгоритму: `ssh-keygen -t rsa -b 4096` и `ssh-keygen -t ed25519`. (рис. 3.5)

```
kisyachinova@dk6n58 ~/tutorial $ ssh-keygen -t rsa -b4096
Generating public/private rsa key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/k/i/kisyachinova/.ssh/id_rsa):
/afs/.dk.sci.pfu.edu.ru/home/k/i/kisyachinova/.ssh/id_rsa already exists.
Overwrite (y/n)?
kisyachinova@dk6n58 ~/tutorial $ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/afs/.dk.sci.pfu.edu.ru/home/k/i/kisyachinova/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /afs/.dk.sci.pfu.edu.ru/home/k/i/kisyachinova/.ssh/id_ed25519
Your public key has been saved in /afs/.dk.sci.pfu.edu.ru/home/k/i/kisyachinova/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:5cm1e3j5b1d3fy71uiAZIyqJYbzdLNFtrkbF0Xe0W10 kisyachinova@dk6n58
The key's randomart image is:
+--[ED25519 256]--+
|
| . . . o .
| o . = . + =E.
| . =.oo +S.O.+ .
| =.* ...o..B . B|
|...+ .o o .+*+|
| ..0 .. . . . *|
| .. o*o|
+-----[SHA256]-----+
kisyachinova@dk6n58 ~/tutorial $
```

Рис. 3.5: Создание ssh ключа

- 6) Создаём ключ `gpg`: `gpg --full-generate-key`. Затем настраиваем: Тип RSA and RSA Размер 4096 Срок действия, значение по умолчанию — 0 (срок действия не истекает никогда). Имя Адрес электронной почты (рис. 3.6)

```
kisyachinova@dk6n58 ~/tutorial $ gpg --full-generate-key
gpg (GnuPG) 2.2.27; Copyright (C) 2021 Free Software Foundation, Inc.
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 и RSA (по умолчанию)
  (2) DSA и Elgamal
  (3) DSA (только для подписи)
  (4) RSA (только для подписи)
  (14) Имеющийся на карте ключ
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: Kseniya Syachinova
Адрес электронной почты: KseniyaZ.ru@yandex.ru
Примечание:
Вы выбрали следующий идентификатор пользователя:
  "Kseniya Syachinova <KseniyaZ.ru@yandex.ru>"

Сменить (N)Имя, (C)Примечание, (E)Адрес: ((O)Организация)
Необходимо получить много случайных чисел
в процессе генерации выполняли какие-то д
на клавиатуре, движения мыши, обращения к
случайных чисел больше возможностей получ
```

Рис. 3.6: Создание gpg ключа

- 7) Добавляем PGP ключ в GitHub. Используем `gpg --list-secret-keys --keyid-format LONG`. По образцу видим отпечаток моего ключа, вставляем его в следующую конструкцию: `gpg--armor--export<pgpFingerprint> | xclip-selclip`. Затем перешли в настройки github и вставили полученный ключ. (рис. 3.7)

```
kisyachinova@dk6n58 ~/tutorial $ 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: 1 подписанных: 0 доверие: 0-, 0q, 0n, 0m, 0f, 1u
/afs/.dk.sci.pfu.edu.ru/home/k/i/kisyachinova/.gnupg/pubring.kbx
-----
sec  rsa4096/9D4621DBCE7B46BA 2022-04-21 [SC]
      7EA44CEf09A5CCBEEAD49B4F9D4621DBCE7B46BA
uid          [ абсолютно ] Kseniya
ssb  rsa4096/B8E84A634B9942CB 2022-04-21

kisyachinova@dk6n58 ~/tutorial $ gpg --armor --export 9D4621DBCE7B46BA | xclip -sel clip
```

Рис. 3.7: Добавление ключей

- 8) Таким образом, у нас получились следующие ssh и gpg ключи: (рис. 3.8)

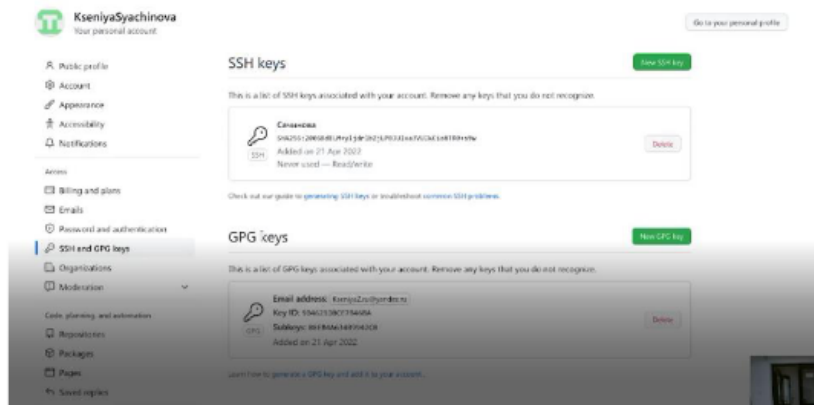


Рис. 3.8: Полученные ключи

9) Затем настраиваем автоматические подписи коммитов git: `git config --global user.signingkey`, `git config --global commit.gpgsign true`, `git config --global gpg.program $(which gpg2)` (рис. 3.9)

```
kisyachinova@dk6n58 ~/tutorial $ git config --global user.signingkey 904621DBCE7B46B
A
kisyachinova@dk6n58 ~/tutorial $ git config --global commit.gpgsigntrue
kisyachinova@dk6n58 ~/tutorial $ git config --global gpg.program$(which gpg2)
error: недействительный ключ: gpg.program/usr/bin/gpg2
kisyachinova@dk6n58 ~/tutorial $ git config --global gpg.program$(which gpg)
error: недействительный ключ: gpg.program/usr/bin/gpg
kisyachinova@dk6n58 ~/tutorial $ git config --global gpg.program $(which gpg2)
```

Рис. 3.9: Настройка коммитов

10) После этого создаём репозиторий курса на основе шаблона. (рис. 3.10)

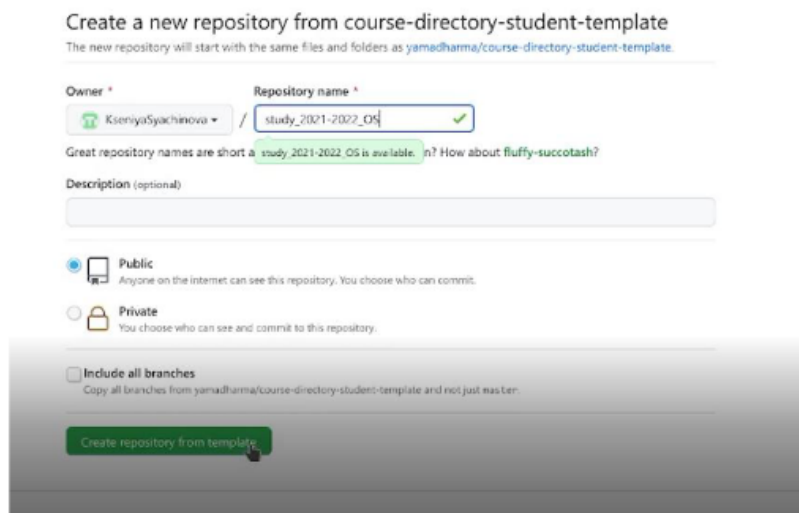


Рис. 3.10: Создание репозитория

- 11) Копируем ссылку и с помощью `git clone -recursive` добавляем наши лабораторные работы на github. Впоследствии удаляем лишние файлы: `rm package.json`, создаём необходимые каталоги: `make COURSE=os-intro`, и отправляем файлы на сервер: `git add .`, `git commit -am 'feat(main): make course structure'`, `git push`. (рис. 3.11)

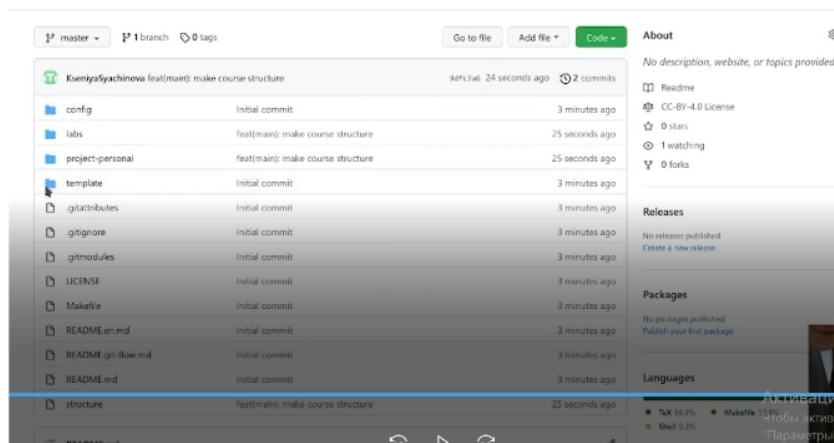


Рис. 3.11: Итог

4 Выводы

Я научилась оформлять отчёты в Markdown, познакомилась с основными его возможностями.