

Лабораторная работа №2

Отчёт

Александр Денисович Мосолов

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	12
	Список литературы	13

Список иллюстраций

3.1	Установка git и gh	7
3.2	Задаём имя и email	7
3.3	Настраиваем кодировку, задаём имя ветки и указываем параметры	8
3.4	Создаём ключ по алгоритму rsa	8
3.5	Создаём ключ по алгоритму ed25519	8
3.6	Генерируем ключ	9
3.7	Выводим и копируем ключ	9
3.8	Настраиваем конфиг	10
3.9	Материалы курса на моём github	10
3.10	Удаляем лишние файлы	11
3.11	Создаём необходимые каталоги	11
3.12	Фиксируем изменения на github	11

Список таблиц

1 Цель работы

Научиться оформлять отчёты с помощью легковесного языка разметки Markdown. [1]

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

Установим git и gh (рис. 3.1):

```
[almos05@almos05 ~]$ sudo dnf install git
[sudo] пароль для almos05:
Последняя проверка окончания срока действия метаданных: 1:34:39 назад, в
т 27 фев 2024 22:39:20.
Пакет git-2.43.2-1.fc39.x86_64 уже установлен.
Зависимости разрешены.
Нет действий для выполнения.
Выполнено!
[almos05@almos05 ~]$ dnf install gh
Ошибка: Эту команду нужно запускать с привилегиями суперпользователя (на
большинстве систем - под именем пользователя root).
[almos05@almos05 ~]$ sudo dnf install gh
Последняя проверка окончания срока действия метаданных: 1:35:25 назад, в
т 27 фев 2024 22:39:20.
Зависимости разрешены.
```

Рис. 3.1: Установка git и gh

Зададим имя и email владельца репозитория (рис. 3.2):

git config --global user.name "Aleksandr Mosolov"

git config --global user.email "tenderboylove2@gmail.com":

```
[almos05@almos05 ~]$ git config --global user.name "Aleksandr Mosolov"
[almos05@almos05 ~]$ git config --global user.email "work@mail"
[almos05@almos05 ~]$ git config --global user.email "tenderboylove2@gmail
1.com"
```

Рис. 3.2: Задаём имя и email

Настроим utf-8 в выводе сообщений git. Зададим имя начальной ветки (будем называть её master). Параметр autocrlf. Параметр safecrlf (рис. 3.3):

```
[almos05@almos05 ~]$ git config --global core.quotepath false
[almos05@almos05 ~]$ git config --global init.defaultBranch master
[almos05@almos05 ~]$ git config --global core.autocrlf input
[almos05@almos05 ~]$ git config --global core.safecrlf warn
```

Рис. 3.3: Настраиваем кодировку, задаём имя ветки и указываем параметры

По алгоритму rsa с ключём размером 4096 бит (рис. 3.4):

ssh-keygen -t rsa -b 4096

```
[almos05@almos05 ~]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/almos05/.ssh/id_rsa):
Created directory '/home/almos05/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/almos05/.ssh/id_rsa
Your public key has been saved in /home/almos05/.ssh/id_rsa.pub
```

Рис. 3.4: Создаём ключ по алгоритму rsa

По алгоритму ed25519 (рис. 3.5):

ssh-keygen -t ed25519

```
[almos05@almos05 ~]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/almos05/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/almos05/.ssh/id_ed25519
Your public key has been saved in /home/almos05/.ssh/id_ed25519.pub
```

Рис. 3.5: Создаём ключ по алгоритму ed25519

Генерируем ключ (рис. 3.6):

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).


```
[almos05@almos05 ~]$ 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/almos05/.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
```

Рис. 3.6: Генерируем ключ

Выводим список ключей и копируем отпечаток приватного ключа:

```
gpg --list-secret-keys --keyid-format LONG
```

Выводим и копируем ключ (рис. 3.7):

```
gpg --armor --export tenderboylive2@gmail.com | cat
```

```
[almos05@almos05 ~]$ gpg --armor --export tenderboylive2@gmail.com | cat
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINOGXeUnsULADKncqkRIINacNIi6GIn6eIIYxoFuLnt3PgW/DSwhc/yqJ5dm0uJH
98gfsy8oSHXpywv8Xhzoy0DF3X0w7VpARACr0KQBjReg3iWLy3MuNQhyXvho99/I
qLyn53BXaGPrIcs08q5D11Ww810wEdWGLE71jTg9nWaoHzylzJjggSt553cRx5C
GRJHaa919W6ac5D51K8uA614XWzCCT86769zN13ARZT1PS101f-Chd5wfbPbgwQ2
KiKdHrCx/yHbjJCwgZS/nmeX21nruRGIj7GYZZDK9BjJT2p3sU3SUPHJ792RwYou
gpnHzIF0f6Gk57f2qXvPIn5drzqBkp385SqA5nMtVAtnGKHGVlwkTqxcPhn7q+2v
2qzC+pei2sNeL6Up0RP4NJoVtsqg/cN0DtYgfeI2znKPD1ZdHlAQc3zrNn5bn1R1m
9Dgd5Ezb1U0aydjwd0g0k1f0xdhGzRL9MN+Yam4e0NAuLRfGt4H+V1zJFxArJcSU
wzhX8dqZrwQ0qE+FTuQFpnUDRapDNiKJd/6kVZcFon2R1DXVeUHRnEkg97aQ81N
eWFebTHnGSL0F7vJtz1AGAULgXpwFcqsg1ZdXycKL0Srz3wgs8ALznISbPG+GvP3s
nX9+jcFQME381md9-HzSjgLYN0dmHfLuVs1G2rN5kk2/ECuEUq08xcljQARAQA8
tCx8hGvVrc2FuZHIqTM9zbxvd1A8dGVu7GVyYn95bG12ZTJA721haMwuY29tPokC
UQQIAQgA0xYhUD2+/egUA5sFAGFH1HgTR08v0S928QJ131J7AhS08QsJCAcCA1IC
BhUKCQgLAqQMAgMBAh4HAheAAkoJEHgTR08v0S92EbgP/j3sXHLX0v1cXIzeKfFH
6BRn0Njxbk9ZhvQG181/PfP/5q5qh5ZfAjV6GJUURnuI9TYnXvUhcGV1WjTGDshq6
KuLDe7csx1Va2KKH41x1gemJpICvNUX4wan4ZxKUSqStdJm1LF4tE/Y4-Kt9A1Y4
IMpQkfjppgyDDgcqoa1H8XZJRgV9k0pdzzqvDSMNy4H6y1YE7QJOYt0QkyjETNRY
WREFY5KdAn7TPkn026UoMzSB55HCWXXZVwMfjnnG+J26SjFwedUdJna7N8aU+wjT
zKbK2-mbFw70zPvmo4P6moCskoA6QNVVRVhtxgn1C-Nob2KfyCECzZC0GzeHbayf
-----END PGP PUBLIC KEY BLOCK-----
```

Рис. 3.7: Выводим и копируем ключ

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

Настраиваем конфиг (рис. 3.8):

```
git config --global user.signingkey tenderboylive2@gmail.com
```

```
git config --global commit.gpgsign true
git config --global gpg.program $(which gpg2)
```

```
[almos05@almos05 ~]$ git config --global user.signingkey tenderboylive2@gmail.com
[almos05@almos05 ~]$ git config --global commit.gpgsign true
[almos05@almos05 ~]$ git config --global gpg.program $(which gpg2)
[almos05@almos05 ~]$ 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? /home/almos05/.ssh/id_rsa.pub
? Title for your SSH key: 3way
? How would you like to authenticate GitHub CLI? [Use arrows to move, type to filter]
> Login with a web browser
  Paste an authentication token
```

Рис. 3.8: Настраиваем конфиг

Необходимо создать шаблон рабочего пространства (рис. 3.9):

```
mkdir -p ~/work/study/2023-2024/“Операционные системы”
cd ~/work/study/2023-2024/“Операционные системы”
gh repo create study_2023-2024_os-intro --template=yamadharm/course-directory-student-template --public
git clone --recursive git@github.com:/study_2023-2024_os-intro.git os-intro
```

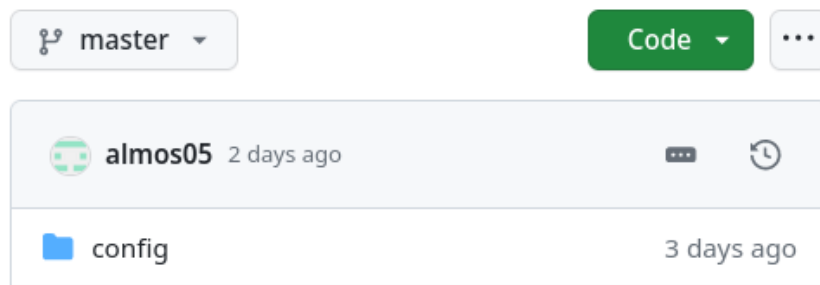


Рис. 3.9: Материалы курса на моём github

Переходим в каталог курса и удаляем лишние файлы (рис. 3.10):

```
cd ~/work/study/2023-2024/“Операционные системы”/os-intro
rm package.json
```

```
[alnos05@alnos05 ~]$ cd ~/work/study/2023-2024/"Операционные системы"/os
-intro
[alnos05@alnos05 os-intro]$ ls
CHANGELOG.md  COURSE  Makefile  README.git-flow.md  template
config        LICENSE  README.en.md  README.md
[alnos05@alnos05 os-intro]$ rm package.json
rm: невозможно удалить 'package.json': Нет такого файла или каталога
[alnos05@alnos05 os-intro]$
```

Рис. 3.10: Удаляем лишние файлы

Создаём необходимые каталоги (рис. 3.11):

echo os-intro > COURSE

make

```
[alnos05@alnos05 os-intro]$ echo os-intro > COURSE
[alnos05@alnos05 os-intro]$ make
Usage:
  make <target>

Targets:
  list           List of courses
  prepare       Generate directories structure
  submodule      Update submodules
[alnos05@alnos05 os-intro]$
```

Рис. 3.11: Создаём необходимые каталоги

Фиксируем изменения на github (рис. 3.12):

git add .

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

git push

```
[alnos05@alnos05 labs]$ git add .
[alnos05@alnos05 labs]$ git commit -am 'feat(main): make course structure'
[alnos05@alnos05 labs]$ git push
```

Рис. 3.12: Фиксируем изменения на github

4 Выводы

В ходе работы мы научились оформлять отчёты с помощью легковесного языка разметки Markdown.

Список литературы

1. Kulyabov. Архитектура компьютеров и операционные системы. Раздел "Операционные системы" (09.03.03, НПИбд).