

Отчет по лабораторной работе №1

Дисциплина: Научное программирование

Выполнила Дяченко Злата Константиновна, НПМмд-02-22

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
4.1	Шаг 1	8
4.2	Шаг 2	9
4.3	Шаг 3	9
4.4	Шаг 4	10
4.5	Шаг 5	11
4.6	Шаг 6	11
4.7	Шаг 7	12
4.8	Шаг 8	12
5	Выводы	14
	Список литературы	15

Список иллюстраций

4.1	Задание имени и email владельца репозитория	8
4.2	Настройка utf-8 в выводе сообщений git	8
4.3	Задание имени начальной ветки, параметра autocrlf и параметра safecrlf	8
4.4	Создание ключа ssh по алгоритму rsa	9
4.5	Создание ключа ssh по алгоритму ed25519	9
4.6	Генерация ключа pgr	10
4.7	Список ключей	10
4.8	Экспорт ключа	11
4.9	Добавленный PGP ключ	11
4.10	Настройка автоматических подписей коммитов git	11
4.11	Авторизация	12
4.12	Сообщение	12
4.13	Создание папки	13
4.14	Создание репозитория	13
4.15	Создание репозитория	13
4.16	Отправка файлов на сервер	13

Список таблиц

1 Цель работы

Изучить идеологию и применение средств контроля версий и освоить умения по работе с git.

2 Задание

- Создать базовую конфигурацию для работы с git.
- Создать ключ SSH.
- Создать ключ PGP.
- Настроить подписи git.
- Зарегистрироваться на Github.
- Создать локальный каталог для выполнения заданий по предмету.

3 Теоретическое введение

Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их производить откат к любой более ранней версии проекта, если это требуется. Система контроля версий Git представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды `git` с различными опциями.

4 Выполнение лабораторной работы

4.1 Шаг 1

Создала ранее учётную запись на <https://github.com> и заполнила основные данные. С помощью команд, представленных на Рисунке 1 (рис. -fig. 4.1), в командной строке задала имя и email владельца репозитория.

```
D:\Учеба\workD>git config --global user.name "Zlata Dyachenko"  
D:\Учеба\workD>git config --global user.email "1132223497@pfur.ru"
```

Рис. 4.1: Задание имени и email владельца репозитория

С помощью команд, представленных на Рисунке 2 (рис. -fig. 4.2), настроила utf-8 в выводе сообщений git.

```
D:\Учеба\workD>git config --global core.quotepath false
```

Рис. 4.2: Настройка utf-8 в выводе сообщений git

С помощью команд, представленных на Рисунке 3 (рис. -fig. 4.3), задала имя начальной ветки, параметр autocrlf и параметр safecrlf.

```
D:\Учеба\workD>git config --global init.defaultBranch master  
D:\Учеба\workD>git config --global core.autocrlf input  
D:\Учеба\workD>git config --global core.safecrlf warn
```

Рис. 4.3: Задание имени начальной ветки, параметра autocrlf и параметра safecrlf

4.2 Шаг 2

Создала ключ ssh по алгоритму rsa размером 4096 бит, что показано на Рисунке 4 (рис. -fig. 4.4).

```
D:\Учеба\workD>ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\honor/.ssh/id_rsa):
C:\Users\honor/.ssh/id_rsa already exists.
Overwrite (y/n)? n
```

Рис. 4.4: Создание ключа ssh по алгоритму rsa

Создала ключ ssh по алгоритму ed25519, что показано на Рисунке 5 (рис. -fig. 4.5).

```
D:\Учеба\workD>ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (C:\Users\honor/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\honor/.ssh/id_ed25519.
Your public key has been saved in C:\Users\honor/.ssh/id_ed25519.pub.
The key fingerprint is:
SHA256:YdAwSdGx4i8a0r6+/NA/wezNtgAbHQsv+Ak9qvM0tDI honor@LAPTOP-MDMGQE00
The key's randomart image is:
+--[ED25519 256]--+
|      .+B=       |
|     ...O+       |
|      ... O      |
|    o +.O. .     |
| +.*++ S        |
|..*oB=          |
|E+.=..+         |
|*+. .O.+        |
|oX* .O..        |
+----[SHA256]-----+
```

Рис. 4.5: Создание ключа ssh по алгоритму ed25519

4.3 Шаг 3

Сгенерировала ключ pgr с помощью команды, показанной на Рисунке 6 (рис. -fig. 4.6).

```

D:\Учеба\workD>gpg --full-generate-key
gpg (GnuPG) 2.3.7; Copyright (C) 2021 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: Злата
Адрес электронной почты: 1132223497@pfur.ru
Примечание:
Используется таблица символов 'utf-8'.
Вы выбрали следующий идентификатор пользователя:
  "Злата <1132223497@pfur.ru>"

Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(O)Выход? 0
Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(O)Выход? 0
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: C:\Users\honor\AppData\Roaming\gnupg\trustdb.gpg: создана таблица доверия
gpg: создан каталог 'C:\Users\honor\AppData\Roaming\gnupg\openpgp-revocs.d'
gpg: сертификат отзыва записан в 'C:\Users\honor\AppData\Roaming\gnupg\openpgp-revocs.d\68E9AB933BAB39EC936085AE07463551AAC12A76'

```

Рис. 4.6: Генерация ключа pgp

4.4 Шаг 4

Вывела список ключей и скопировала отпечаток приватного ключа. На Рисунке 7 (рис. -fig. 4.7) показан результат выполнения команды.

```

D:\Учеба\workD>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
C:\Users\honor\AppData\Roaming\gnupg\pubring.kbx
-----
sec   rsa4096/07463551AAC12A76 2022-09-15 [SC]
      68E9AB933BAB39EC936085AE07463551AAC12A76
uid   [ абсолютно ] Злата <1132223497@pfur.ru>
ssb   rsa4096/0A8F137C85A2DFB7 2022-09-15 [E]

```

Рис. 4.7: Список ключей

4.5 Шаг 5

Экспортировала ключ с помощью команды, показанной на Рисунке 8 (рис. -fig. 4.8) и скопировала его.

```
D:\Учеба\workD>gpg --armor --export 07463551AAC12A76
-----BEGIN PGP PUBLIC KEY BLOCK-----
```

Рис. 4.8: Экспорт ключа

Перешла в настройки GitHub, нажала на кнопку New GPG key и вставила полученный ключ в поле ввода. Новый PGP ключ был добавлен, что подтверждает Рисунок 9 (рис. -fig. 4.9).

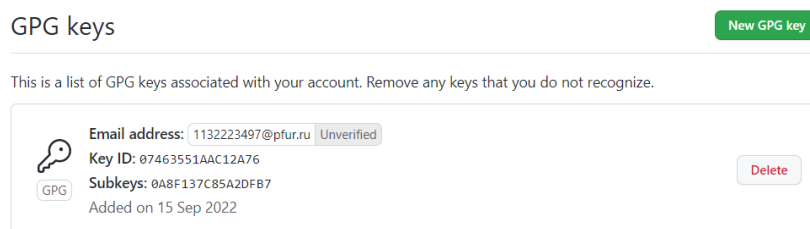


Рис. 4.9: Добавленный PGP ключ

4.6 Шаг 6

Используя введённый email, указала Git применять его при подписи коммитов с помощью команд, показанных на Рисунке 10 (рис. -fig. 4.10).

```
D:\Учеба\workD>git config --global user.signingkey 07463551AAC12A76
D:\Учеба\workD>git config --global commit.gpgsign true
D:\Учеба\workD>git config --global gpg.program $(which gpg2)
```

Рис. 4.10: Настройка автоматических подписей коммитов git

4.7 Шаг 7

Авторизовалась через браузер, выполнив сначала команду, показанную на Рисунке 11 (рис. -fig. 4.11), и ответив на несколько вопросов.

```
D:\Учеба\workD>gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations? HTTPS
? Authenticate Git with your GitHub credentials? Yes
? How would you like to authenticate GitHub CLI? Login with a web browser

! First copy your one-time code: 9ABA-DFF7
Press Enter to open github.com in your browser...
[✓] Authentication complete.
- gh config set -h github.com git_protocol https
[✓] Configured git protocol
[✓] Logged in as ZlataDyachenko
```

Рис. 4.11: Авторизация

В результате получила сообщение, продемонстрированное на Рисунке 12 (рис. -fig. 4.12).

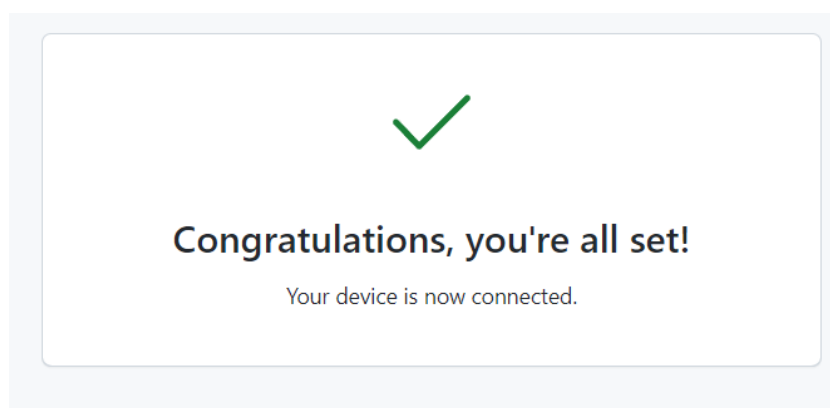


Рис. 4.12: Сообщение

4.8 Шаг 8

Создала папку, имеющую имя соответствующее названию курса и перешла в нее с помощью команд, показанных на Рисунке 13 (рис. -fig. 4.13).

```
D:\Учеба\workD\2022-2023> mkdir D:\Учеба\workD\2022-2023\Научное программирование"
D:\Учеба\workD\2022-2023>cd D:\Учеба\workD\2022-2023\Научное программирование"
```

Рис. 4.13: Создание папки

Создала репозиторий, скопировав шаблонный репозиторий с помощью команды, показанной на Рисунке 14 (рис. -fig. 4.14) и Рисунке 15 (рис. -fig. 4.15).

```
D:\Учеба\workD\2022-2023\Научное программирование>gh repo create study_2022-2023_sci-prog
--template=yamadharma/course-directory-student-template --public
Created repository ZlataDyachenko/study_2022-2023_sci-prog on GitHub
```

Рис. 4.14: Создание репозитория

```
D:\Учеба\workD\2022-2023\Научное программирование>git clone --recursive git@github.com:ZlataDyachenko/study_2022-2023_sci-prog.git sci-prog
Cloning into 'sci-prog'...
remote: Enumerating objects: 26, done.
remote: Counting objects: 100% (26/26), done.
remote: Compressing objects: 100% (25/25), done.
remote: Total 26 (delta 0), reused 17 (delta 0), pack-reused 0R
Receiving objects: 100% (26/26), 16.03 KiB | 2.00 MiB/s, done.
Submodule 'template/presentation' (https://github.com/yamadharma/academic-presentation-markdown-template.git) registered for path 'template/presentation'
Submodule 'template/report' (https://github.com/yamadharma/academic-laboratory-report-template.git) registered for path 'template/report'
Cloning into 'D:\Учеба\workD\2022-2023\Научное программирование\sci-prog\template\presentation'...
remote: Enumerating objects: 71, done.
remote: Counting objects: 100% (71/71), done.
remote: Compressing objects: 100% (49/49), done.
remote: Total 71 (delta 23), reused 68 (delta 20), pack-reused 0
Receiving objects: 100% (71/71), 88.89 KiB | 938.00 KiB/s, done.
Resolving deltas: 100% (23/23), done.
Cloning into 'D:\Учеба\workD\2022-2023\Научное программирование\sci-prog\template\report'...
remote: Enumerating objects: 78, done.
remote: Counting objects: 100% (78/78), done.
remote: Compressing objects: 100% (52/52), done.
remote: Total 78 (delta 31), reused 69 (delta 22), pack-reused 0
Receiving objects: 100% (78/78), 292.27 KiB | 612.00 KiB/s, done.
Resolving deltas: 100% (31/31), done.
Submodule path 'template/presentation': checked out '2703b47423792d472694aaf7555a5626dce51a25'
Submodule path 'template/report': checked out 'df7b2ef80f8def3b9a496f8095277469a1a7842a'
```

Рис. 4.15: Создание репозитория

Удалила файл *package.json*, создала необходимые каталоги и отправила файлы на сервер, что продемонстрировано на Рисунке 16 (рис. -fig. 4.16).

```
D:\Учеба\workD\2022-2023\Научное программирование\sci-prog>git add .
D:\Учеба\workD\2022-2023\Научное программирование\sci-prog>git commit -am "feat(main): make course structure"
[master 3c32779] feat(main): make course structure
2 files changed, 1 insertion(+), 14 deletions(-)
delete mode 100644 package.json
D:\Учеба\workD\2022-2023\Научное программирование\sci-prog>git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 955 bytes | 955.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:ZlataDyachenko/study_2022-2023_sci-prog.git
1253889..3c32779 master -> master
```

Рис. 4.16: Отправка файлов на сервер

5 Выводы

Я изучила идеологию и применение средств контроля версий, а также освоила умения по работе с git. Результаты работы находятся в репозитории на GitHub, а также есть скринкаст выполнения лабораторной работы.

Список литературы