

# **Системы контроля версий**

**Лабораторная работа №2**

Козомазов Владимир Романович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
3.0.1	Теоретическое введение по регистрации в GitHub . . . . .	7
3.0.2	1. <b>Что такое GitHub?</b> . . . . .	7
3.0.3	2. <b>Зачем нужна регистрация на GitHub?</b> . . . . .	8
3.0.4	3. <b>Основные понятия</b> . . . . .	8
3.0.5	4. <b>Процесс регистрации</b> . . . . .	9
3.0.6	5. <b>Настройка учетной записи</b> . . . . .	9
3.0.7	6. <b>Первый шаг после регистрации</b> . . . . .	9
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>10</b>
<b>5</b>	<b>Выводы</b>	<b>14</b>
	<b>Список литературы</b>	<b>15</b>

## Список иллюстраций

4.1	Проверка установки git . . . . .	10
4.2	Проверка установки gh . . . . .	10
4.3	Проверка имени владельца репозитория . . . . .	10
4.4	Проверка email владельца репозитория . . . . .	10
4.5	Настройка utf-8 в выводе сообщений git . . . . .	11
4.6	Проверка наличия SSH ключа . . . . .	11
4.7	Создание gpg ключа и выбор необходимых опций . . . . .	11
4.8	Вывод списка ключей . . . . .	12
4.9	Вывод на экран сгенерированного ключа и копирование его в буфер обмена . . . . .	12
4.10	Добавление ключа на github . . . . .	13
4.11	Добавление ключа на github . . . . .	13

## **Список таблиц**

# 1 Цель работы

- Изучить идеологию и применение средств контроля версий.
- 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

### 3.0.1 Теоретическое введение по регистрации в GitHub

GitHub — это одна из самых популярных платформ для совместной разработки программного обеспечения, основанная на системе контроля версий Git. Она предоставляет инструменты для управления кодом, отслеживания задач, совместной работы и автоматизации процессов разработки. Регистрация на GitHub — это первый шаг к использованию всех возможностей этой платформы.

---

### 3.0.2 1. Что такое GitHub?

- **GitHub** — это веб-сервис, который предоставляет хостинг для репозиториях Git. Он позволяет разработчикам хранить код, управлять версиями, сотрудничать с другими участниками и автоматизировать процессы разработки.
  - GitHub поддерживает как публичные (открытые для всех), так и приватные (доступные только для определенных пользователей) репозитории.
  - Платформа также предоставляет дополнительные функции, такие как Issues (для отслеживания задач), Wiki (для документации), Projects (для управления проектами) и Actions (для CI/CD).
-

### 3.0.3 2. Зачем нужна регистрация на GitHub?

Регистрация на GitHub открывает доступ к следующим возможностям: - **Хранение кода**: Создание репозитория для хранения и управления проектами. - **Совместная работа**: Возможность приглашать других разработчиков для работы над проектами. - **Контроль версий**: Использование Git для отслеживания изменений в коде. - **Участие в open-source**: Возможность вносить вклад в открытые проекты. - **Автоматизация**: Использование GitHub Actions для настройки CI/CD (непрерывной интеграции и доставки). - **Документирование**: Создание Wiki и README-файлов для описания проектов.

---

### 3.0.4 3. Основные понятия

- **Репозиторий**: Хранилище кода, которое содержит все файлы проекта, историю изменений и настройки.
  - **Коммит (Commit)**: Отдельное изменение в коде, которое сохраняется в истории репозитория.
  - **Ветка (Branch)**: Изолированная версия кода, которая позволяет работать над новыми функциями или исправлениями, не затрагивая основную версию.
  - **Пул-реквест (Pull Request)**: Запрос на объединение изменений из одной ветки в другую. Используется для проверки и обсуждения кода перед его внесением в основную ветку.
  - **Issues**: Задачи или баги, которые необходимо решить в рамках проекта.
  - **Wiki**: Документация проекта, которая может быть создана и отредактирована участниками.
-



### 3.0.5 4. Процесс регистрации

Регистрация на GitHub включает следующие шаги: 1. **Переход на сайт GitHub:** Открыть сайт `github.com`. 2. **Заполнение формы регистрации:** - Ввод имени пользователя (`username`), которое будет использоваться для доступа к репозиториям и профилю. - Ввод адреса электронной почты. - Создание пароля. 3. **Подтверждение учетной записи:** - Подтверждение электронной почты, указанной при регистрации. 4. **Настройка профиля:** - Добавление аватарки, имени и краткого описания. - Настройка двухфакторной аутентификации (2FA) для повышения безопасности.

---

### 3.0.6 5. Настройка учетной записи

После регистрации важно настроить учетную запись для удобства и безопасности: - **Двухфакторная аутентификация (2FA):** Добавление дополнительного уровня защиты. - **SSH-ключи:** Настройка SSH для безопасного взаимодействия с репозиториями без необходимости ввода пароля. - **Интеграция с IDE:** Настройка интеграции GitHub с популярными средами разработки, такими как Visual Studio Code, IntelliJ IDEA и другими.

---

### 3.0.7 6. Первый шаг после регистрации

После регистрации рекомендуется: - Создать первый репозиторий. - Изучить интерфейс GitHub (вкладки Repositories, Issues, Pull Requests, Actions и другие). - Попробовать выполнить базовые операции с Git (клонирование репозитория, создание коммитов, создание веток).

---

## 4 Выполнение лабораторной работы

Проверил установку git с помощью команды `git --version` (рис. 4.1).

```
Vladimir@DESKTOP-AI090TO MINGW64 ~  
$ git --version  
git version 2.48.1.windows.1
```

Рис. 4.1: Проверка установки git

Проверил установку gh с помощью команды `gh -version` (рис. 4.2).

```
Vladimir@DESKTOP-AI090TO MINGW64 ~  
$ gh --version  
gh version 2.68.1 (2025-03-06)  
https://github.com/cli/cli/releases/tag/v2.68.1
```

Рис. 4.2: Проверка установки gh

Проверил имя владельца репозитория командой `git config --global user.name` (рис. 4.3).

```
Vladimir@DESKTOP-AI090TO MINGW64 ~  
$ git config --global user.name  
Vladimir Kozomazov
```

Рис. 4.3: Проверка имени владельца репозитория

Проверил email владельца репозитория командой `git config --global user.email` (рис. 4.4).

```
Vladimir@DESKTOP-AI090TO MINGW64 ~  
$ git config --global user.email  
voffkoz@gmail.com
```

Рис. 4.4: Проверка email владельца репозитория

Настроил utf-8 в выводе сообщений git командой `git config --global core.quotePath false` (рис. 4.5).

```
Vladimir@DESKTOP-AI090T0 MINGW64 ~
$ git config --global core.quotePath false
Vladimir@DESKTOP-AI090T0 MINGW64 ~
$
```

Рис. 4.5: Настройка utf-8 в выводе сообщений git

Проверил наличие SSH ключа командой `ls ~/.ssh` (рис. 4.6).

```
Vladimir@DESKTOP-AI090T0 MINGW64 ~
$ ls ~/.ssh
id_ed25519 id_ed25519.pub known_hosts known_hosts.old
Vladimir@DESKTOP-AI090T0 MINGW64 ~
$
```

Рис. 4.6: Проверка наличия SSH ключа

Создал ключ gpg с помощью команды `gpg --full-generate-key` и выбрал необходимые опции (рис. 4.7).

```
Vladimir@DESKTOP-AI090T0 MINGW64 ~
$ gpg --full-generate-key
gpg (GnuPG) 2.4.5-unknown; Copyright (C) 2024 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.

Please select what kind of key you want:
(1) RSA and RSA
(2) DSA and Elgamal
(3) DSA (sign only)
(4) RSA (sign only)
(9) ECC (sign and encrypt) "default"
(10) ECC (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
    <n>w = key expires in n weeks
    <n>m = key expires in n months
    <n>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: vkozomazov
Email address: voffkoz@gmail.com
Comment:
You selected this USER-ID:
"vkozomazov <voffkoz@gmail.com>"
```

Рис. 4.7: Создание gpg ключа и выбор необходимых опций

Вывел список ключей командой `gpg --list-secret-keys --keyid-format LONG` (рис. 4.8).

```
Vladimir@DESKTOP-A1090TD MINGW64 ~
$ gpg --list-secret-keys --keyid-format LONG
gpg: checking the trustdb
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: depth: 0 valid: 2 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 2u
[keyboard]
-----
sec rsa4096/7816A9728D800CD0 2025-03-06 [SC]
      2CB06620D7F647C25527BD07816A9728D800CD0
uid          [ultimate] vkozomazov <i132246812@pfur.com>
ssb          rsa4096/CB06A4E48Z6D8280 2025-03-06 [E]

sec rsa4096/79CE826E4C1B5D91 2025-03-06 [SC]
      D92A766C938E47437DEAD01879CE826E4C1B5D91
uid          [ultimate] vkozomazov <voffk0z@gmail.com>
ssb          rsa4096/75A23896F9E9A124 2025-03-06 [E]

Vladimir@DESKTOP-A1090TD MINGW64 ~
$
```

Рис. 4.8: Вывод списка ключей

Вывел на экран сгенерированный gpg ключ командой `gpg --armor --export 7816A9728D800CD0` и скопировал в буфер обмена (рис. 4.10).

```
Vladimir@DESKTOP-A1090TD MINGW64 ~
$ gpg --armor --export 7816A9728D800CD0
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGFjInU8EAD1RkZyMpcQDQw6Zj1h60LFxPrc3k360/B7VRFVOPxZGT
P421cWpPmF3e118rpg1P+uLVZVpBkKw/F1unD9Om-mdu5KfCvZ
4EXDn0zXNZVH0ADBUkDPVWxP1STak2wu2UQ+eo5HaZ8y1VApT0FmucCKPq
h1qA6tL11MMQKQVn/n154h1JCSPYBcdGkXOR4VQZtV01Pa3Dg-dJA7i68497
Itoz-QH0Vms3YDk8W831E1Qz51QZ-m0j66P51Z6/g48k33ed0JmWm0mZ
Vd0Q2CycKRev5yewldFV6+N16V+pgQ50Lk1L12EEIZmDf653z1Fkq151346uN
Vier-mUkWX872GrYrMMPHGVL4Dw71JRM8AB8rps35vkhcKTUzjq8B3j37ho
CU0Vsp0377q4RFD038tCcmQbaaz-Fm1Dc0p0v418360z7z7j3QUP3GEX
3b41Z8Z7CVAmPmZxfr+k6Q9M2xm6LIP20LpHNVuf3Jw2VvR84L78knY2wQJ/GFu
QIHCA9y0gH/kbGCAKFRu5G41GvA/SgvwXf3mJNSEH578a1Udk/cntB+Skux7D
qubfF1u7fU19/TtG1N8E5xP502u4wJFANU7arXSPQVCqEqr4Q0ARQA08
KC82a96b21hem921DwMThyMjQ020EYQH8ndL1uY2P0kCUQTAgQ40xYH8Cy9
hmInF+R81V770HgwKXKgaZ0B0jyz/VahsD0Q3JCacA1ICBhKCGlAg0kAgB
h4h4h44440718gm0XNg4D0C-4Mhe8k32kvbv8cHhgyu11kai+V1VLXVme
kLwLHau2M9dYZ2Z-F8/7/00qfge0vOh881ReuN1-ON3ENXAB3Lz8Wu3-D3Y1Vhs
B31J3YRQDtv41ny5D1a3g5TOx60wy6QNY8cLR8w1VjGZL1TheGr3ohhh1PRkzZ
2Qk5p6h0M0Mret31T7Wp4s4Z20C/zMRuX0Jpcevmhg0JmWm0m03z6
+81Aluqzy6Bfpm7ZVUGFDMKvndph1rpgM1xcnZk0jkyMLt0JYd4kqk5Bsy8
t151ETQ4MF82UwbkHTXkCJjgd801Lvg1MD6j/O1ney6V0Mh5H2v0WzxtHT1
2YV3AMR+Rc7H0eKUS/KrphKAtXZ3IuEdw0D0MwX7ZDm9NkgZDs5sr
VcPHH1vZSN2+d1R1untA20RyFztrYXfD0YF6hct/ASQhK8Qxwxd11015rM10
C/9BJ6CNCumtG0uJCF8dxxYPP1pi2UFa2SZrX9LqgTRvRbXUVM093MvlecJH
h1170h0N1B1k01G3JWau0mcturxf/NL33AY0kdukmYvdb3w3Bm/Q2Gn
pdk+aaXpUSHu8esJzc0Te88CH9zadsYzGt7zctDRHweeqJ5K1V7Fh6k32
1Z6xG4RQ1NBGFjInU8EACvK8B6q0/V1moTVq/h2ZLMSCL1Eq/4T0hptZ
Y/4333ygrT1N401010V/4GZuLwR72UUCwK0C9r3Hh4VtEn1E4jg10
Wnffr/B8+NxkXtX0cym7+H1TMQNTfLbntjMWS1kWhuALB2g2paw77e0skdZuR
I2Z7r5Rf1BRu09u/VtttutuhfFbv8rnb1Pm87MHZN8Rumd979MeY6G6Z3ne
NLC1+sc40h14C10wP1/n5087F882pRq1V4WZC0MeytL5C51Z3g0uH13p
L37w8Tj2yPcZ73KLL1E1/Fg4Lq0N9aXLSZ5/ETvvdpcNzaBIZ1rnf6EQJdshM
Pnr44k36wFKX0Z++IAzz170/THZ2/g3V2JNGAYT6dsf6NhbIUDYd3aMAAmfw
QUD6U21scTZGP073zfc4K1S0p0u1ur3j2p2QmHMcgbW0XfP0u3j1u4o
gUfC140H1NqLeqP5r9ndjkmv0LSgeGfJAccd0JTu5qsmAS18vRg0Q1WZArB
1g0/v0TK8ahRu6jyzPq33nyplaz7ChKJfDphQhJr5n75V0jphlvagvpjgtU19
7ZLSU8QJH4E4C0G8Z3Z3V4VLEBYU7Fm3J2p1mE2L0qly5cE8FMR2
Z0ARQA081Q128BgCAAg1EEILL2CvG1/5Hw1UmvQeapco2ADNAfAmF3n9UCGwaA
CkQeapco2ADND11A/SASa7Xfng3DeYsX8QgnKsVzGqPW/M4svB1Gxzf0vBB
hcmNMfJno23JmVxX0M2z2D0Cvkh0ebUkexR1kVdJL0ysvvyRvge8Y
pA1L/Aaw5970j9mZyEubk1ozkEks/SvkhLSUjYKA/pgmBPx5ePxmN02GBAFu
18dheV1m1K01+VULGw0gEKJ+/ro9rveXwdB115qf9db6bhzTsmu1N57X/
4k8w4v423Mw70u1F463FyCQYV0P9PKEFEDHn1F0pgrFmkJ18P9ZQ07
Bee+814nkKp+swLYCF8gb71uL8Qc1Ip31N1RkrAQkZf772QacqR/I4adiZx
1sD08Kk4DezyPye1BC1t5cphK4c6G1aMcECL5Qops008w19ek4vPR0HEZ0P
hP1P4T22hNhb23PEZv0P0Mh1/guev61hC242w3J1c3d8d1wepm02w1
knev8UuKGu5zAsRdd406ULCL8h530tqHs3hsraU3adN7jEX8Dn51fkwo17h9
VcPy+3vkh+aa66Qv74eQNE10912Mc3MH68JESUYh3yblA07KvJ+030DuspKp
r4QMR44K103ZaExxQDZCYK1KnyAasUr4G829ZPwuk303jrUQ1as3yq4D-
9JyG
-----END PGP PUBLIC KEY BLOCK-----
```

Рис. 4.9: Вывод на экран сгенерированного ключа и копирование его в буфер обмена

Добавил gpg ключ на github (рис. 4.9).

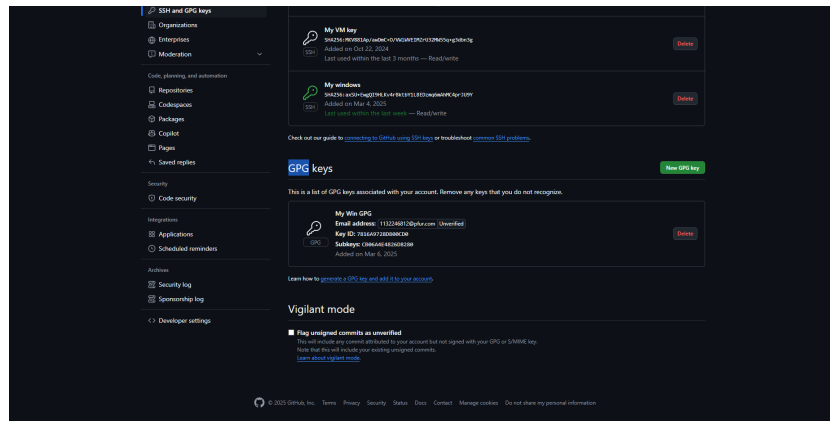


Рис. 4.10: Добавление ключа на github

Настроил автоматичесие подписи коммитов git (рис. 4.11).

```
ladywin@DESKTOP-AI0907D MINGW64 ~
$ git config --global user.signingkey 7816A9728D800C00

ladywin@DESKTOP-AI0907D MINGW64 ~
$ git config --global commit.gpgsign true

ladywin@DESKTOP-AI0907D MINGW64 ~
$ git config --global gpg.program $(which gpg)

ladywin@DESKTOP-AI0907D MINGW64 ~
$ |
```

Рис. 4.11: Добавление ключа на github

## 5 Выводы

Работа по регистрации и настройке GitHub позволила получить базовые навыки, необходимые для эффективного использования этой платформы. GitHub является неотъемлемой частью современной разработки, и освоение его функций открывает широкие возможности для участия в open-source проектах, командной работе и профессионального роста. Полученные знания и навыки могут быть применены как в учебных, так и в профессиональных целях.

## **Список литературы**