

Лабораторная работа №2

Дисциплина: Архитектура компьютеров и операционные системы

Люкшина Влада Алексеевна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	15

Список иллюстраций

4.1	Установка git	8
4.2	Установка gh	9
4.3	Задаем имя и email	9
4.4	Настраиваем git	9
4.5	Создаем ключ rsa размером 4096 бит	10
4.6	Создаем ключ ed25519	10
4.7	Генерируем ключ pgr	11
4.8	Копируем отпечаток	11
4.9	Добавляем ключ	12
4.10	Настраиваем подписи	12
4.11	Генерируем код	12
4.12	Авторизируемся	13
4.13	Создаем шаблон	13
4.14	Работа с каталогом	14
4.15	Отправляем файлы	14

Список таблиц

1 Цель работы

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

2 Задание

- 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.

3 Теоретическое введение

Система контроля версий Git представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды `git` с различными опциями.

Благодаря тому, что Git является распределённой системой контроля версий, резервную копию локального хранилища можно сделать простым копированием или архивацией.

4 Выполнение лабораторной работы

Первым пунктом лабораторной работы №2 является установка git. Устанавливаем.

```
foot
[valyukshina@valyukshina ~]$ sudo dnf install git
[sudo] пароль для valyukshina:
Обновление и загрузка репозитория:
Fedora 41 - x86_64 - Updates      100% | 36.5 KiB/s | 22.1 KiB | 00m01s
Fedora 41 - x86_64 - Updates      100% | 3.4 MiB/s | 2.8 MiB | 00m01s
Репозитории загружены.
Пакет "git-2.48.1-1.fc41.x86_64" уже установлен.

Нечего делать.
[valyukshina@valyukshina ~]$
```

Рис. 4.1: Установка git

Далее устанавливаем gh.


```
[valyukshina@valyukshina ~]$ sudo dnf install gh
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет      Арх.  Версия      Репозиторий  Размер
Установка:
gh         x86_64  2.65.0-1.fc41  updates      42.6 MiB

Сводка транзакции:
Установка:      1 пакета

Общий размер входящих пакетов составляет 10 MiB. Необходимо загрузить 10 MiB.
После этой операции будут использоваться дополнительные 43 MiB (установка 43 MiB, удаление 0 B).
Is this ok [y/N]: y
[1/1] gh-0:2.65.0-1.fc41.x86_64      100% | 23.3 MiB/s | 10.3 MiB | 00m00s
-----
[1/1] Total                          100% | 8.8 MiB/s | 10.3 MiB | 00m01s
Выполнение транзакции
[1/3] Проверить файлы пакетов 100% | 20.0 B/s | 1.0 B | 00m00s
[2/3] Подготовить транзакцию 100% | 1.0 B/s | 1.0 B | 00m01s
[3/3] Установка gh-0:2.65.0-1.fc41 100% | 16.8 MiB/s | 42.7 MiB | 00m03s
Завершено!
[valyukshina@valyukshina ~]$
```

Рис. 4.2: Установка gh

Следующим пунктом идет базовая настройка git. Задаем свое имя и email как владельца репозитория.

```
[valyukshina@valyukshina ~]$ git config --global user.name "valyukshina"
[valyukshina@valyukshina ~]$ git config --global user.email "liuckshinava@gmail.com"
[valyukshina@valyukshina ~]$
```

Рис. 4.3: Задаем имя и email

Настраиваем utf-8 в выводе сообщения git. Задаем имя начальной ветки, называем ее master. Задаем два параметра: autocrlf и safecrlf.

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

Рис. 4.4: Настраиваем git

По алгоритму создаем ключи ssh.

```
[valyukshina@valyukshina ~]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/valyukshina/.ssh/id_rsa):
Created directory '/home/valyukshina/.ssh'.
Enter passphrase for "/home/valyukshina/.ssh/id_rsa" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/valyukshina/.ssh/id_rsa
Your public key has been saved in /home/valyukshina/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:f372cnXFdNH61Z+yICCJWJ+pWDB4s04Kf0yIwPT81pA valyukshina@valyukshina
The key's randomart image is:
+---[RSA 4096]-----+
|o+o...oB|
|..+..oo..o+|
|+.o.+o..o+|
|o.oo.E...+|
|.+o.o$.o.o|
|o..o...o|
|..o...oo|
|o..o..o+o|
|..o...o|
+---[SHA256]-----+
[valyukshina@valyukshina ~]$
```

Рис. 4.5: Создаем ключ rsa размером 4096 бит

```
[valyukshina@valyukshina ~]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/valyukshina/.ssh/id_ed25519):
Enter passphrase for "/home/valyukshina/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/valyukshina/.ssh/id_ed25519
Your public key has been saved in /home/valyukshina/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:1DFr8U5sB9w2IinegYfNu0GJq4T1PIFrxRp74CGPRU valyukshina@valyukshina
The key's randomart image is:
+--[ED25519 256]--+
|      +E*+...      |
|      o 0+X0o.+     |
|      o ^=B=+=o...  |
|      .o++*+++      |
|      .o=...So.     |
|      ... o.        |
|                    |
|                    |
+---[SHA256]-----+
[valyukshina@valyukshina ~]$
```

Рис. 4.6: Создаем ключ ed25519

Генерируем ключи pgr. Выбираем предложенные опции, вводим личную информацию.

```
[valyukshina@valyukshina ~]$ gpg --full-generate-key
gpg (GnuPG) 2.4.5; 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.

gpg: создан каталог '/home/valyukshina/.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
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: Vlada Lyukshina
Адрес электронной почты: liuckshinava@gmail.com
Примечание:
Вы выбрали следующий идентификатор пользователя:
  "Vlada Lyukshina <liuckshinava@gmail.com>"

Сменить (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход? 
```

Рис. 4.7: Генерируем ключ gpg

Созданный ключ нам необходимо добавить в github. Выводим список ключей и копируем отпечаток приватного ключа.

```
[valyukshina@valyukshina ~]$ 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, 0m, 0f, 1u
[keyboard]
-----
sec  rsa4096/000D238A62563FDB 2025-03-07 [SC]
    77F6093F168AF961B3D410A3000D238A62563FDB
uid          [ абсолютно ] Vlada Lyukshina <liuckshinava@gmail.com>
ssb  rsa4096/D89564874FE8887A 2025-03-07 [E]
```

Рис. 4.8: Копируем отпечаток

Копируем сгенерированный PGP ключ в буфер обмена и вставляем в поле ввода в github.

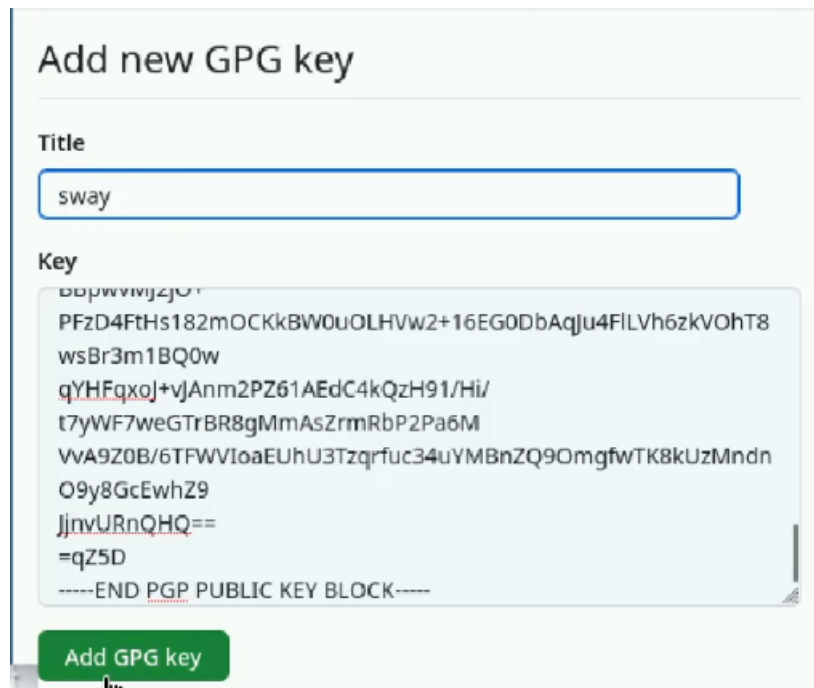


Рис. 4.9: Добавляем ключ

Изначально коммиты неподписаны, они отмечаются как неподтвержденные. Поэтому нам необходимо настроить подписи, используя нашу почту.

```
[valyukshina@valyukshina ~]$ git config --global user.signingkey liuckshinava@gmail.com
[valyukshina@valyukshina ~]$ git config --global commit.gpgsign true
[valyukshina@valyukshina ~]$ git config --global gpg.program $(which gpg2)
```

Рис. 4.10: Настраиваем подписи

Авторизируемся в gh с помощью браузера. Генерируется код для входа.

```
[valyukshina@valyukshina ~]$ gh auth login
? Where do you use GitHub? GitHub.com
? What is your preferred protocol for Git operations on this host? SSH
? Upload your SSH public key to your GitHub account? /home/valyukshina/.ssh/id_rsa.pub
? Title for your SSH key: sway
? How would you like to authenticate GitHub CLI? Login with a web browser

! First copy your one-time code: FF7D-3AED
Press Enter to open https://github.com/login/device in your browser...
```

Рис. 4.11: Генерируем код

Копируем сгенерированный код, вставляем его в открывшемся окне и авторизируемся.

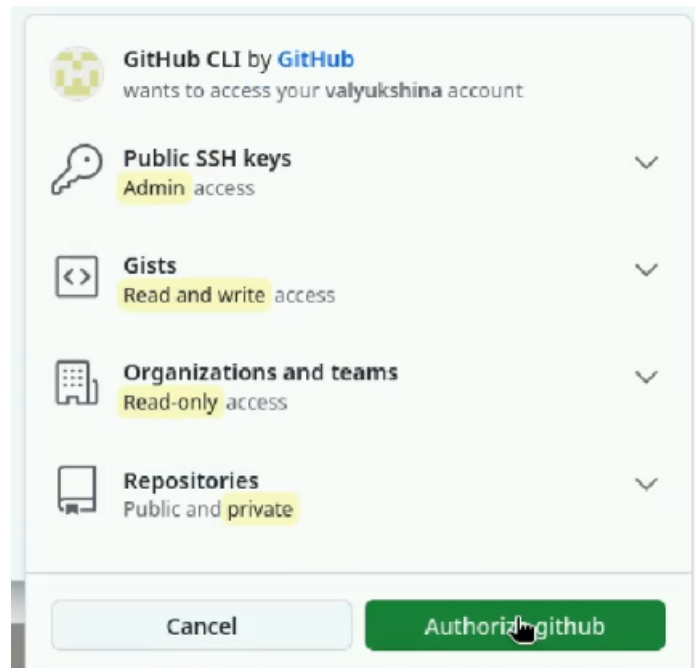


Рис. 4.12: Авторизируемся

После авторизации нам необходимо создать шаблон рабочего пространства. Создаем необходимые папки, переходим туда и копируем шаблон с github.

```
[valyukshina@valyukshina Операционные системы]$ git clone --recursive git@github.com:valyukshina/study_2024-2025_os-intro.git os-intro
Клонирование в «os-intro»...
```

Рис. 4.13: Создаем шаблон

Переходим в каталог созданного курса, удаляем лишние файлы и создаем необходимые каталоги.

```
[valyukshina@valyukshina Операционные системы]$ cd ~/work/study/2024-2025/"Операционные
системы"/os-intro
[valyukshina@valyukshina os-intro]$ rm package.json
[valyukshina@valyukshina os-intro]$ echo os-intro > COURSE
[valyukshina@valyukshina os-intro]$ make
Usage:
  make <target>

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

[valyukshina@valyukshina os-intro]$ make prepare
[valyukshina@valyukshina os-intro]$
```

Рис. 4.14: Работа с каталогом

Отправляем файлы на сервер.

```
[valyukshina@valyukshina os-intro]$ git push
Перечисление объектов: 40, готово.
Подсчет объектов: 100% (40/40), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (30/30), готово.
Запись объектов: 100% (38/38), 342.31 КиБ | 2.50 МБ/с, готово.
Total 38 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:valyukshina/study_2024-2025_os-intro.git
   d394bc3..6e33aa1  master -> master
[valyukshina@valyukshina os-intro]$
```

Рис. 4.15: Отправляем файлы

5 Выводы

В ходе лабораторной работы мы научились базовым навыкам работы с github. Научились создавать репозитории, шаблоны, работать с ключами.