

Лабораторная работа №2

Операционные системы

Краснова Камилла Геннадьевна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Установка программного обеспечения	7
3.2	Базовая настройка git	7
3.3	Создание ключей ssh	8
3.4	Создание ключей gpg	9
3.5	Настройка github	11
3.6	Добавление gpg ключа в Github	12
3.7	Настройка автоматических подписей коммитов git	13
3.8	Настройка gh	13
3.9	Создание репозитория курса на основе шаблона	13
3.10	Настройка каталога курса	14
4	Выводы	16
5	Ответы на контрольные вопросы.	17
	Список литературы	20

Список иллюстраций

3.1	Установка git и gh	7
3.2	Данные владельца	7
3.3	Настройка utf-8	7
3.4	Имя начальной ветки	8
3.5	Параметры autocrlf и safecrlf	8
3.6	Алгоритм rsa	8
3.7	Алгоритм ed25519	9
3.8	Генерация ключа	10
3.9	Генерация ключа	11
3.10	Учетная запись github	11
3.11	Список ключей	12
3.12	New GPG key	12
3.13	Готовый gpg ключ	12
3.14	Автоматические подписи	13
3.15	Авторизация	13
3.16	Репозиторий курса	14
3.17	Каталог курса	15

Список таблиц

1 Цель работы

Целью данной лабораторной работы является изучение идеологии и применения средств контроля версий, а также освоение умений по работе с git.

2 Задание

1. Создать базовую конфигурацию для работы с git
2. Создать ключ SSH
3. Создать ключ GPG
4. Настроить подписи Git
5. Зарегистрироваться на GitHub
6. Создать локальный каталог для выполнения заданий по предмету.

3 Выполнение лабораторной работы

3.1 Установка программного обеспечения

Захожу в терминал и устанавливаю git, следом устанавливаю gh (рис. 3.1).

```
[root@kkrasnova ~]# dnf install git
Обновление и загрузка репозитория:
Fedora 41 - x86_64 - Updates 100% | 6.8 KiB/s | 18.5 KiB | 00m03s
Fedora 41 - x86_64 - Updates 100% | 1.0 MiB/s | 2.6 MiB | 00m03s
Репозитории загружены.
Пакет "git-2.45.1-1.fc41.x86_64" уже установлен.

Нечего делать.
[root@kkrasnova ~]# dnf install gh
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет
Установка:
gh
Арх.
x86_64
Версия
2.65.0-1.fc41
Репозиторий
updates
Размер
42.6 MiB
Сводка транзакции:
Установка: 1 пакета
Общий размер входящих пакетов составляет 10 MiB. Необходимо загрузить 10 MiB.
После этой операции будет использоваться дополнительное 43 MiB (установка 43 MiB, удаление 0 B).
```

Рис. 3.1: Установка git и gh

3.2 Базовая настройка git

Задаю имя и email владельца репозитория (рис. 3.2).

```
[root@kkrasnova ~]# git config --global user.name "kakras"
[root@kkrasnova ~]# git config --global user.email "lzip184@gmail.com"
```

Рис. 3.2: Данные владельца

Настраиваю utf-8 в выводе сообщений git (рис. 3.3).

```
[root@kkrasnova ~]# git config --global core.quotepath false
[root@kkrasnova ~]#
```

Рис. 3.3: Настройка utf-8

Задаю имя начальной ветки (рис. 3.4).

```
[root@kakrasnova ~]# git config --global init.defaultBranch master
[root@kakrasnova ~]#
```

Рис. 3.4: Имя начальной ветки

Далее задаю параметры autocrlf и safecrlf (рис. 3.5).

```
[root@kakrasnova ~]# git config --global core.autocrlf input
[root@kakrasnova ~]# git config --global core.safecrlf warn
[root@kakrasnova ~]#
```

Рис. 3.5: Параметры autocrlf и safecrlf

3.3 Создание ключей ssh

Создаю ключ размером 4096 бит по алгоритму rsa (рис. 3.6).

```
[root@kakrasnova ~]# ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase for "/root/.ssh/id_rsa" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:8rRzDcipAV8YZfAn/1QrxSaTMTXc1RaR2oWyn03Ep44 root@kakrasnova.net
The key's randomart image is:
+---[RSA 4096]-----+
|      ooo  ooo=B|
|      =  .oo=|
|    . . + ..+*B..|
|    o o *  +B+o.|
|    + S o oo.o |
|    * . = .o |
|    . + . oo |
|    o E . |
|
+-----[SHA256]-----+
[root@kakrasnova ~]#
```

Рис. 3.6: Алгоритм rsa

Дальше создаю ключ по алгоритму ed25519 (рис. 3.7).


```

[root@kakrasnova ~]# ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519):
Enter passphrase for "/root/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_ed25519
Your public key has been saved in /root/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:nLFA0pf88y14cegWJmLKz1AkGM5JSUd08nI8B1elp8s root@kakrasnova.net
The key's randomart image is:
+--[ED25519 256]--+
|o=+*.+oo.o      |
|=.+ 0o. =       |
| + + *.o.o .    |
|   = +o++* o .  |
|  o o oSo B +   |
| . o . . . = .  |
| +   E   . .    |
|  o           |
+-----[SHA256]-----+
[root@kakrasnova ~]#

```

Рис. 3.7: Алгоритм ed25519

3.4 Создание ключей gpg

Генерирую ключ (рис. 3.8).

```

[root@kakrasnova ~]# 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 '/root/.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)
Срок действия ключа не ограничен
Все верно? (y/N) y

```

Рис. 3.8: Генерация ключа

И выбираю нужные опции. Подтверждаю и придумываю кодовую фразу (рис. 3.9).

```

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)
Срок действия ключа не ограничен
Все верно? (y/N) y

GnuPG должен составить идентификатор пользователя для идентификации ключа.

Ваше полное имя: KrasnovaKamilla
Адрес электронной почты: lzip184@gmail.com
Примечание:
Вы выбрали следующий идентификатор пользователя:
    "KrasnovaKamilla <lzip184@gmail.com>"

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

```

Рис. 3.9: Генерация ключа

3.5 Настройка github

Учетная запись у меня уже была создана, а также были заполнены основные данные (рис. 3.10).

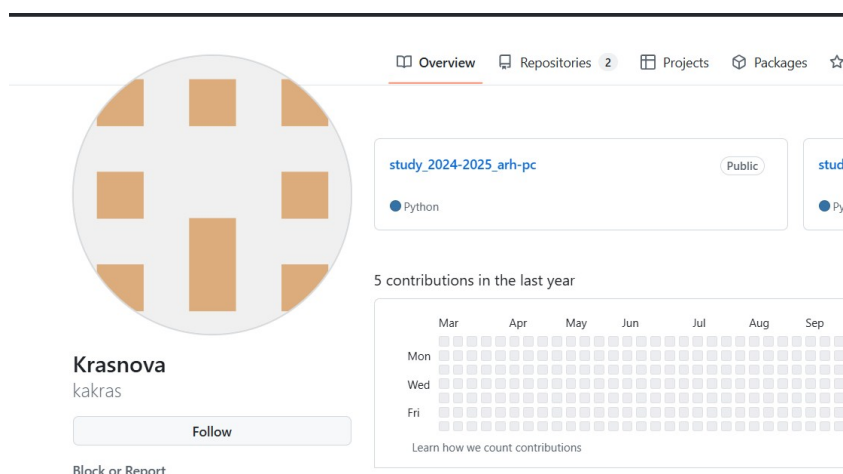


Рис. 3.10: Учетная запись github

3.6 Добавление gpg ключа в Github

Вывожу список ключей и копирую отпечаток приватного ключа (рис. 3.11).

```
[root@kaktasnova ~]# 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
[keyboxd]
-----
sec   rsa4096/F78D901A9F10C190 2025-02-24 [SC]
      3B252CE12E8F8A598A347222F78D901A9F10C190
uid   [ абсолютно ] KrasnovaKamilla <lzip184@gmail.com>
ssb   rsa4096/B55D86D50E05639E 2025-02-24 [E]

[root@kaktasnova ~]#
```

Рис. 3.11: Список ключей

Копирую сгенерированный gpg ключ в буфер обмена и вставляю полученный ключ в поле ввода в настройках GitHub (рис. 3.12).

Add new GPG key

Title

Key

```
ZmbYkbG90VK6pP2LIN2M5Ob/kVX4Jymd7H8Pcsa1YsBOT+twrQ9ayjDAeVkmvk0g
9zMQPfAYCyKgx99NKd/zkTMEkLzj3FRdJfFor8JGQ4t3psPC8aaPv7Po5jyq022
kLy/8tpDM6mQPTZpFu44ujVHQHy+9znxM+Bdwkp6t+TgW+m4+933XfDR2KtB+BD
KAMqT8NVBv4BXjB1AJQWjugUpKYINfa9+fy1X1TEf8T1JaiKONS7eU22itB+bPe6
ufuVEHaf1zDHRhMONt8ByMFCOh4L79/C58eUvKjWjih17zqEI155thOQy9JHwj9
xRw=
=rZ4C
-----END PGP PUBLIC KEY BLOCK-----
```


Add GPG key

Рис. 3.12: New GPG key

GPG ключ добавился в GitHub (рис. 3.13).

GPG keys New GPG key

This is a list of GPG keys associated with your account. Remove any keys that you do not recognize.



Email address: lzip184@gmail.com
Key ID: 56EC4F9819A21849
Subkeys: 3A6B9858174C61C1
Added on Feb 24, 2025

Delete

[Learn how to generate a GPG key and add it to your account.](#)

Vigilant mode

Рис. 3.13: Готовый gpg ключ

3.7 Настройка автоматических подписей коммитов git

Используя введенный email, указываю Git применять его при подписи коммитов (рис. 3.14).

```
root@fedora:~# git config --global user.signingkey 56EC4F9819A21849
root@fedora:~# git config --global commit.gpgsign true
root@fedora:~# git config --global gpg.program $(which gpg2)
```

Рис. 3.14: Автоматические подписи

3.8 Настройка gh

Авторизуюсь и отвечаю на несколько наводящих вопросов от утилиты. В конце выбираю авторизацию через токен. Создаю токен на GitHub и вставляю его в терминал. Получаю сообщение об успешной авторизации (рис. 3.15).

```
kamilla@fedora:~/.gnupg$ gh auth login

? Where do you use GitHub? GitHub.com

? What is your preferred protocol for Git operations on this host? HTTPS

? Authenticate Git with your GitHub credentials? Yes

? How would you like to authenticate GitHub CLI? Paste an authentication token

Tip: you can generate a Personal Access Token here https://github.com/settings/tokens
The minimum required scopes are 'repo', 'read:org', 'workflow'.

? Paste your authentication token: *****
*****- gh config set -h github.com git_protocol https
✓ Configured git protocol
✓ Logged in as kakras
```

Рис. 3.15: Авторизация

3.9 Создание репозитория курса на основе шаблона

У меня уже был создан репозиторий курса (рис. 3.16).



Рис. 3.16: Репозиторий курса

3.10 Настройка каталога курса

Также у меня уже был настроен каталог курса (рис. 3.17).

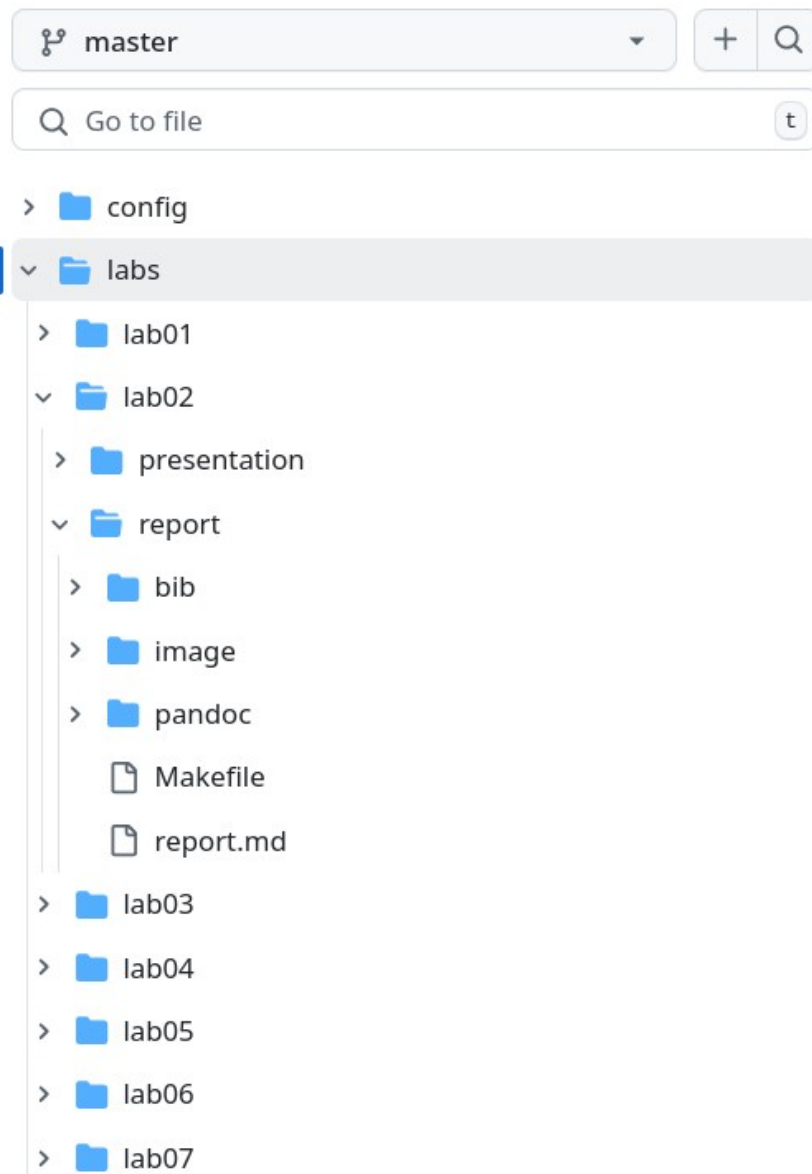


Рис. 3.17: Каталог курса

4 Выводы

При выполнении данной лабораторной работы я изучила идеологию и применение средств контроля версий, а также освоила умения по работе с git.

5 Ответы на контрольные вопросы.

1. Системы контроля версий (VCS) - программное обеспечение для облегчения работы с изменяющейся информацией. Они позволяют хранить несколько версий изменяющейся информации, одного и того же документа, может предоставить доступ к более ранним версиям документа. Используется для работы нескольких человек над проектом, позволяет посмотреть, кто и когда внес какое-либо изменение и т. д. VCS применяются для: Хранения полной истории изменений, сохранения причин всех изменений, поиска причин изменений и совершивших изменение, совместной работы над проектами.
2. Хранилище – репозиторий, хранилище версий, в нем хранятся все документы, включая историю их изменения и прочей служебной информацией. commit – отслеживание изменений, сохраняет разницу в изменениях. История – хранит все изменения в проекте и позволяет при необходимости вернуться/обратиться к нужным данным. Рабочая копия – копия проекта, основанная на версии из хранилища, чаще всего последней версии.
3. Централизованные VCS (например: CVS, TFS, AccuRev) – одно основное хранилище всего проекта. Каждый пользователь копирует себе необходимые ему файлы из этого репозитория, изменяет, затем добавляет изменения обратно в хранилище. Децентрализованные VCS (например: Git, Bazaar) – у каждого пользователя свой вариант репозитория (возможно несколько вариантов), есть возможность добавлять и забирать изменения из любого

репозитория. В отличие от классических, в распределенных (децентрализованных) системах контроля версий центральный репозиторий не является обязательным.

4. Сначала создается и подключается удаленный репозиторий, затем по мере изменения проекта эти изменения отправляются на сервер.
5. Участник проекта перед началом работы получает нужную ему версию проекта в хранилище, с помощью определенных команд, после внесения изменений пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются. К ним можно вернуться в любой момент.
6. Хранение информации о всех изменениях в вашем коде, обеспечение удобства командной работы над кодом.
7. Создание основного дерева репозитория: `git init`

Получение обновлений (изменений) текущего дерева из центрального репозитория: `git pull`

Отправка всех произведённых изменений локального дерева в центральный репозиторий: `git push`

Просмотр списка изменённых файлов в текущей директории: `git status`

Просмотр текущих изменений: `git diff`

Сохранение текущих изменений: добавить все изменённые и/или созданные файлы и/или каталоги: `git add` .

добавить конкретные изменённые и/или созданные файлы и/или каталоги: `git add имена_файлов`

удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории): `git rm имена_файлов`

Сохранение добавленных изменений:

сохранить все добавленные изменения и все изменённые файлы: `git commit -am 'Описание коммита'`

сохранить добавленные изменения с внесением комментария через встроенный редактор: `git commit`

создание новой ветки, базирующейся на текущей: `git checkout -b имя_ветки`

переключение на некоторую ветку: `git checkout имя_ветки` (при переключении на ветку, которой ещё нет в локальном репозитории, она будет создана и связана с удалённой)

отправка изменений конкретной ветки в центральный репозиторий: `git push origin имя_ветки`

слияние ветки с текущим деревом: `git merge --no-ff имя_ветки`

Удаление ветки:

удаление локальной уже слитой с основным деревом ветки: `git branch -d имя_ветки`

принудительное удаление локальной ветки: `git branch -D имя_ветки`

удаление ветки с центрального репозитория: `git push origin :имя_ветки`

8. `git push -all` отправляем из локального репозитория все сохраненные изменения в центральный репозиторий, предварительно создав локальный репозиторий и сделав предварительную конфигурацию.
9. Ветвление - один из параллельных участков в одном хранилище, исходящих из одной версии, обычно есть главная ветка. Между ветками, т. е. их концами возможно их слияние. Используются для разработки новых функций.
10. Во время работы над проектом могут создаваться файлы, которые не следуют добавлять в репозиторий. Например, временные файлы. Можно прописать шаблоны игнорируемых при добавлении в репозиторий типов файлов в файл `.gitignore` с помощью сервисов.

Список литературы

1. Лабораторная работа № 2 [Электронный ресурс] URL: <https://esystem.rudn.ru/mod/page/view>