

Отчёт по лабораторной работе №2

Операционные системы

Морозова Мария Вячеславовна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Ответы на контрольные вопросы	11
5	Выводы	13

Список иллюстраций

3.1	Установка программного обеспечения	7
3.2	Базовая настройка git и создание SSH ключа	7
3.3	Создаём ключ gpg	8
3.4	Добавление GPG ключа в GitHub	8
3.5	Полученный ключ	9
3.6	Настройка автоматических подписей коммитов git, настройка gh	9
3.7	Создание репозитория	10
3.8	Настройка каталога курса	10

Список таблиц

1 Цель работы

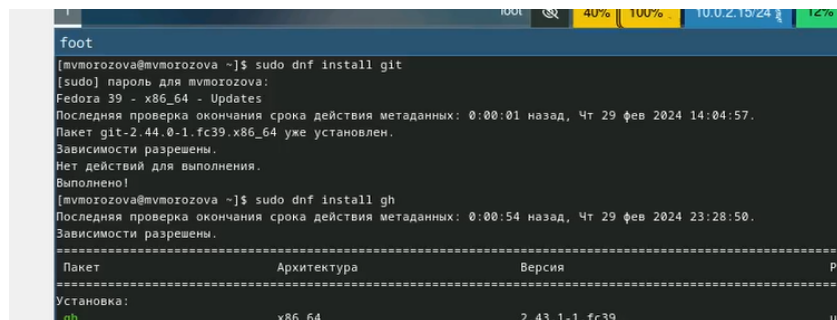
Изучить идеологию и применение средств контроля версий. Освоить умения по работе с git.

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 Выполнение лабораторной работы

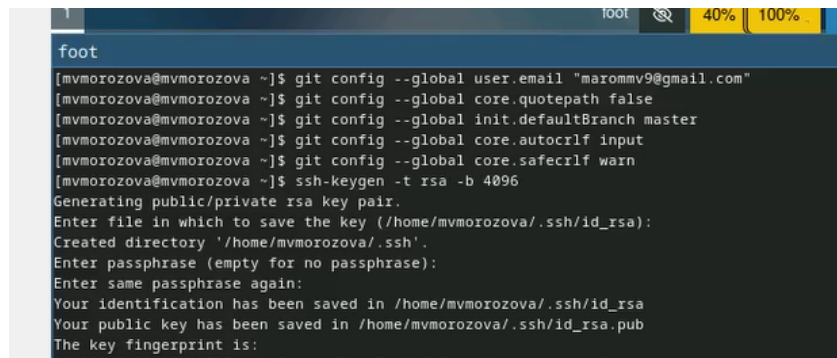
Установка git, установка gh. (рис. 3.1).



```
foot
[mvmorozova@mvmorozova ~]$ sudo dnf install git
[sudo] пароль для mvmorozova:
Fedora 39 - x86_64 - Updates
Последняя проверка окончания срока действия метаданных: 0:00:01 назад, Чт 29 фев 2024 14:04:57.
Пакет git-2.44.0-1.fc39.x86_64 уже установлен.
Зависимости разрешены.
Нет действий для выполнения.
Выполнено!
[mvmorozova@mvmorozova ~]$ sudo dnf install gh
Последняя проверка окончания срока действия метаданных: 0:00:54 назад, Чт 29 фев 2024 23:28:50.
Зависимости разрешены.
=====
Пакет      Архитектура      Версия      Р
-----
Установка:
gh          x86_64           2.43.1-1.fc39  up
```

Рис. 3.1: Установка программного обеспечения

Задаём имя и почту владельца репозитория, задаём имя начальной ветки, параметр autocrlf, параметр safecrlf, создаём ключ SSH по алгоритму rsa (рис. 3.2).



```
foot
[mvmorozova@mvmorozova ~]$ git config --global user.email "marommv9@gmail.com"
[mvmorozova@mvmorozova ~]$ git config --global core.quotepath false
[mvmorozova@mvmorozova ~]$ git config --global init.defaultBranch master
[mvmorozova@mvmorozova ~]$ git config --global core.autocrlf input
[mvmorozova@mvmorozova ~]$ git config --global core.safecrlf warn
[mvmorozova@mvmorozova ~]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/mvmorozova/.ssh/id_rsa):
Created directory '/home/mvmorozova/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/mvmorozova/.ssh/id_rsa
Your public key has been saved in /home/mvmorozova/.ssh/id_rsa.pub
The key fingerprint is:
```

Рис. 3.2: Базовая настройка git и создание SSH ключа

Генерируем ключ (рис. 3.3).

```
+---[SHA256]-----+
[mvmorozova@mvmorozova ~]$ 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/mvmorozova/.gnupg'
Выберите тип ключа:
  (1) RSA and RSA
  (2) DSA and Elgamal
```

Рис. 3.3: Создаём ключ gpg

Выводим список ключей и копируем сгенерированный ключ. (рис. 3.4).

```
foot
[mvmorozova@mvmorozova ~]$ gpg --list-secret-keys --keyid-format LONG
gpg: проверка таблицы доверия
gpg: removing stale lockfile (created by 2806)
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: глубина: 0 достоверных: 1 подписанных: 0 доверие: 0-, 0q, 0n
, 0m, 0f, 1u
[keyboard]
-----
sec   rsa4096/C9736AEF795170BA 2024-02-29 [SC]
      B715B619646C35EE24178E0BC9736AEF795170BA
uid           [ абсолютно ] Maria Morozova <marommv9@gmail.com>
ssb   rsa4096/8704286860860084 2024-02-29 [E]

[mvmorozova@mvmorozova ~]$ gpg --armor --export marommv9@gmail.com | xcl
ip -sel clip
[mvmorozova@mvmorozova ~]$ cat
```

Рис. 3.4: Добавление GPG ключа в GitHub

Вставляем полученный ключ в поле ввода. (рис. 3.5).

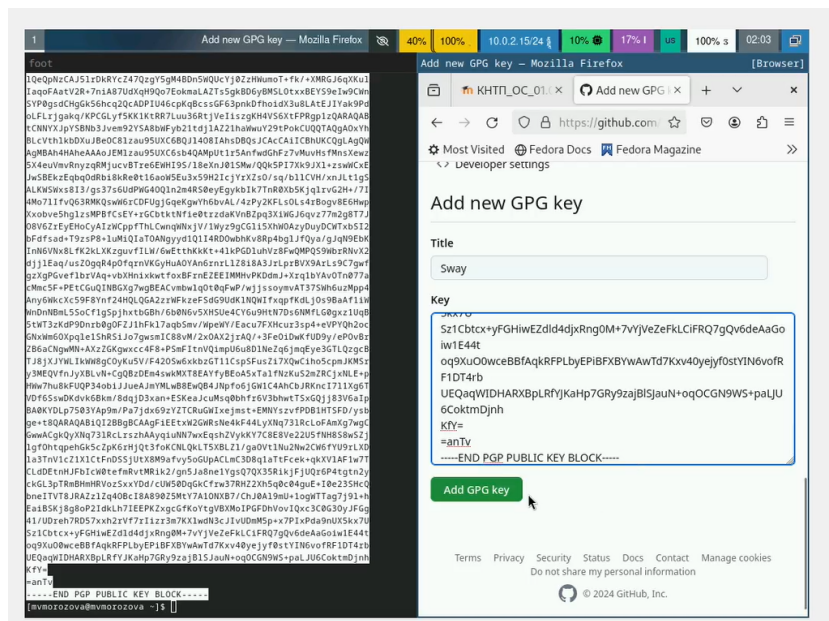


Рис. 3.5: Полученный ключ

Используя введённую почту, указываю применять её при подписи коммитов, авторизация (рис. 3.6).

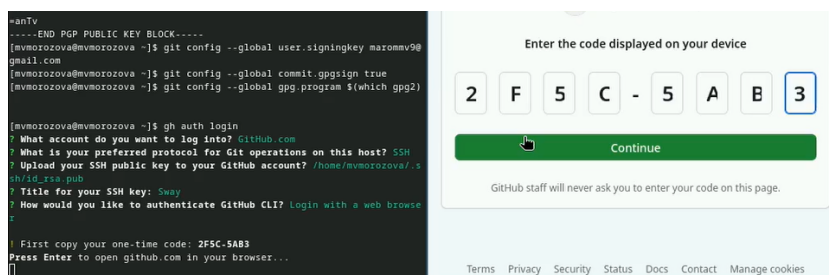


Рис. 3.6: Настройка автоматических подписей коммитов git, настройка gh

Создаю репозиторий курса на основе шаблона.(рис. 3.7).

```
[mvmorozova@mvmorozova ~]$ mkdir -p ~/work/study/2023-2024/"Операционные системы"
[mvmorozova@mvmorozova ~]$ cd ~/work/study/2023-2024/"Операционные системы"
[mvmorozova@mvmorozova Операционные системы]$ gh repo create study_2022-2023_os-intro --template=yamadharma/course-directory-student-template --public
✓ Created repository chitapr/study_2022-2023_os-intro on GitHub
https://github.com/chitapr/study_2022-2023_os-intro
[mvmorozova@mvmorozova Операционные системы]$ gh repo create study_2023-2024_os-intro --template=yamadharma/course-directory-student-template --public
✓ Created repository chitapr/study_2023-2024_os-intro on GitHub
https://github.com/chitapr/study_2023-2024_os-intro
[mvmorozova@mvmorozova Операционные системы]$ git clone --recursive git@github.com:chitapr/study_2023-2024_os-intro.git os-intro
Клонирование в «os-intro»...
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0aPMSvHdkr4UvC0qU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

Рис. 3.7: Создание репозитория

Перехожу в каталог курса, удаляю лишние файлы, создаю необходимые каталоги, отправляю файлы на сервер. (рис. 3.8).

```
a19ef8028ced88e'
[mvmorozova@mvmorozova Операционные системы]$ cd ~/work/study/2023-2024/"Операционные системы"/os-intro
[mvmorozova@mvmorozova os-intro]$ rm package.json
[mvmorozova@mvmorozova os-intro]$ echo os-intro > COURSE
[mvmorozova@mvmorozova os-intro]$ make
Usage:
  make <target>

Targets:
  list           List of courses
  prepare        Generate directories structure
  submodule      Update submodules
[mvmorozova@mvmorozova os-intro]$ make prepare
[mvmorozova@mvmorozova os-intro]$ git add .
[mvmorozova@mvmorozova os-intro]$ git commit -am 'feat(main): make course structure'
```

Рис. 3.8: Настройка каталога курса

4 Ответы на контрольные вопросы

1. Системы контроля версий - это программное обеспечение для облегчения работы с изменяющейся информацией. Применяются при работе нескольких человек над одним проектом. 2. Хранилище - репозиторий, хранилище версий, в нём хранятся все документы и прочая служебная информация. Commit - отслеживание изменений, сохраняет разницу в изменениях. История - хранит изменения в проекте и позволяет вернуться к нужным данным. Рабочая копия - копия проекта, основанная на версии из хранилища. 3. Централизованные - одно основное хранилище всего проекта (CVS, TFS). Децентрализованные - у каждого пользователя свой вариант репозитория. (Git, Bazaar) 4. Сначала создаётся и подключается удаленный репозиторий, затем по мере изменения проекта эти изменения отправляются на сервер. 5. Участник проекта перед работой получает нужную версию проекта в хранилище, с помощью определенных команд после внесения изменений пользователь размещает новую версию в хранилище. Предыдущие версии не удаляются и к ним всегда можно вернуться. 6. Основная задача - это хранение информации об изменениях в коде, обеспечение удобной командной работы. 7. git init создание основного дерева репозитория git diff просмотр текущих изменений git status просмотр списка изменённых файлов в директории 8. git push -all отправляем из локального репозитория все сохранённые изменения в центральный репозиторий, предварительно создав локальный репозиторий и сделав предварительную конфигурацию. 9. Ветвление - используется для разработки новых функций, один из параллельных участков в одном хранилище, исходящих из одной версии. 10. Во время работы могут создаваться фай-

лы, которые не следует добавлять в репозиторий, например, временные файлы. Можно прописать шаблоны игнорируемых при добавлении в репозиторий типов файлов в файл `.gitignore` с помощью сервисов.

5 Выводы

Была изучена идеология и применение средств контроля версий, освоены умения по работе с git.