

Отчет по лабораторной работе №2

Операционные системы

Серебрякова Дарья Ильинична

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Настройка git	7
3.2	Создание ключа SSH	7
3.3	Создание ключа GPG	8
3.4	Добавление ключа GPG в гитхаб	9
3.5	Настройка подписи git	10
3.6	Настройка gh	10
3.7	Создание репозитория курса на основе шаблона	11
4	Выводы	13
5	Ответы на контрольные вопросы	14
6	Список литературы	17

Список иллюстраций

3.1	1	7
3.2	2	7
3.3	5	8
3.4	6	8
3.5	7	9
3.6	8	9
3.7	9	10
3.8	10	10
3.9	11	10
3.10	12	10
3.11	13	11
3.12	14	11
3.13	15	12
3.14	16	12
3.15	17	12

Список таблиц

1 Цель работы

Цель данной работы - изучение идеологии и применения средств контроля версий, освоение умения работы с git

2 Задание

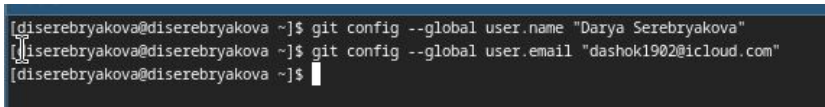
1. Создать базовую конфигурацию для работы с git
2. создать ключи SSH и GPG
3. Создать локальный каталог для выполнения заданий по предмету

3 Выполнение лабораторной работы

3.1 Настройка git

Устанавливаю необходимое программное обеспечение и перехожу к настройке git.

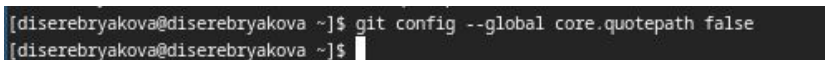
Задаю в качестве имени и адреса почты владельца репозитория свои данные (рис. 3.1).

A terminal window showing two git configuration commands. The first command sets the global user name to "Darya Serebryakova". The second command sets the global user email to "dashok1902@icloud.com".

```
[diserebryakova@diserebryakova ~]$ git config --global user.name "Darya Serebryakova"
[diserebryakova@diserebryakova ~]$ git config --global user.email "dashok1902@icloud.com"
[diserebryakova@diserebryakova ~]$
```

Рис. 3.1: 1

Настраиваю utf-8 в выводе сообщений гит для их корректного отображения (рис. 3.2).

A terminal window showing a git configuration command to set the core.quotePath to false.

```
[diserebryakova@diserebryakova ~]$ git config --global core.quotePath false
[diserebryakova@diserebryakova ~]$
```

Рис. 3.2: 2

3.2 Создание ключа SSH

Создаю ключ ssh размером 4096 бит по алгоритму rsa (рис. 3.3).

```
[diserebryakova@diserebryakova ~]$ ssh-keygen -C "Darya Serebryakova <dashok1902@icloud.com>"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/diserebryakova/.ssh/id_ed25519): ssh
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ssh
Your public key has been saved in ssh.pub
The key fingerprint is:
SHA256:SgPc4t915z85H8gJbpVYFAhzJlw15N2Lx3uToLM3zpw Darya Serebryakova <dashok1902@icloud.com>
The key's randomart image is:
+--[ED25519 256]--+
|      .++*o+      |
|      .+*+      |
|      .o+      |
|      .o+      |
|      .o+      |
|      .o+      |
|      .o+      |
|      .o+      |
|      .o+      |
|      .o+      |
+-----[SHA256]-----+
[diserebryakova@diserebryakova ~]$
```

Рис. 3.3: 5

Ключ создан (рис. 3.4).

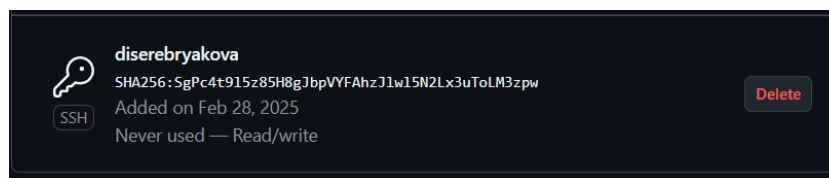


Рис. 3.4: 6

3.3 Создание ключа GPG

Генерирую ключ gpg, затем выбираю тип ключа, задаю максимальную его длину и ставлю неограниченный срок действия ключа (рис. 3.5).


```
serebryakova@diserebryakova ~]$ gpg --full-generate-key
(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.

: создан каталог '/home/diserebryakova/.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
: выбор? 1
: на ключей RSA может быть от 1024 до 4096.
: Какой размер ключа Вам необходим? (3072) 4096
: предложенный 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
: Верно? (y/N) y

GPG должен составить идентификатор пользователя для идентификации ключа.
: Ваше полное имя: DashaSerebryakova
: Ваш адрес электронной почты: dashok1902@icloud.com
: Примечание:
: выбрали следующий идентификатор пользователя:
: "DashaSerebryakova <dashok1902@icloud.com>"
: Введите (N)Имя, (C)Примечание, (E)Адрес; (O)Принять/(Q)Выход?
```

Рис. 3.5: 7

3.4 Добавление ключа GPG в гитхаб

Вывожу список созданных ключей в терминал, копирую в буфер обмена отпечаток ключа (рис. 3.6).

```
serebryakova@diserebryakova ~]$ gpg --list-secret-keys --keyid-format LONG
: проверка таблицы доверия
: marginals needed: 3 completes needed: 1 trust model: pgp
: глубина: 0 достоверных: 1 подписанных: 0 доверие: 0-, 0q, 0n, 0m, 0f, 1u
: [yboxd]
-----
: rsa4096/45F95FD3A29A8A75 2025-02-28 [SC]
: 5FF55F7BE87002BC3F08F2745F95FD3A29A8A75
: [ абсолютно ] DashaSerebryakova <dashok1902@icloud.com>
: rsa4096/1A8833A65BB60A3B 2025-02-28 [E]
serebryakova@diserebryakova ~]$
```

Рис. 3.6: 8

Ввожу в терминале команду, с помощью которой копирую сам ключ в буфер обмена (рис. 3.7).

```
serebryakova@diserebryakova ~]$ gpg --armor --export 45F95FD3A29A8A75 | xclip -sel clip
serebryakova@diserebryakova ~]$
```

Рис. 3.7: 9

Ключ создан (рис. 3.8).

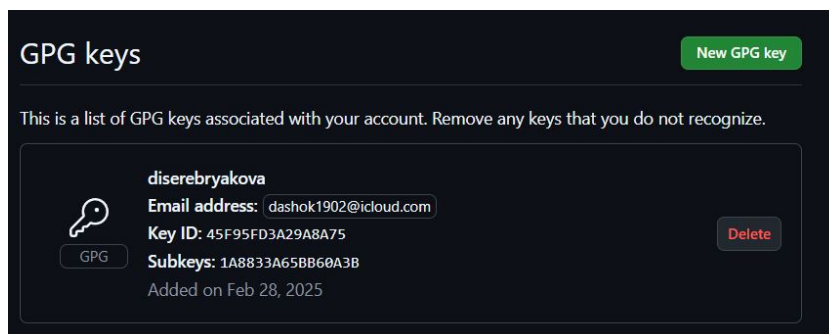


Рис. 3.8: 10

3.5 Настройка подписи git

Настраиваю автоматические подписи коммитов гит (рис. 3.9).

```
serebryakova@diserebryakova ~]$ git config --global user.signingkey 45F95FD3A29A8A75
serebryakova@diserebryakova ~]$ git config --global commit.gpgsign true
serebryakova@diserebryakova ~]$ git config --global gpg.program $(which gpg2)
serebryakova@diserebryakova ~]$
```

Рис. 3.9: 11

3.6 Настройка gh

Начинаю авторизацию в gh, авторизируюсь через браузер (рис. 3.10).

```
[diserebryakova@diserebryakova ~]$ 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? Login with a web browser

! First copy your one-time code: 1368-5D07
Press Enter to open https://github.com/login/device in your browser...
```

Рис. 3.10: 12

Авторизация успешно выполнена (рис. 3.11).

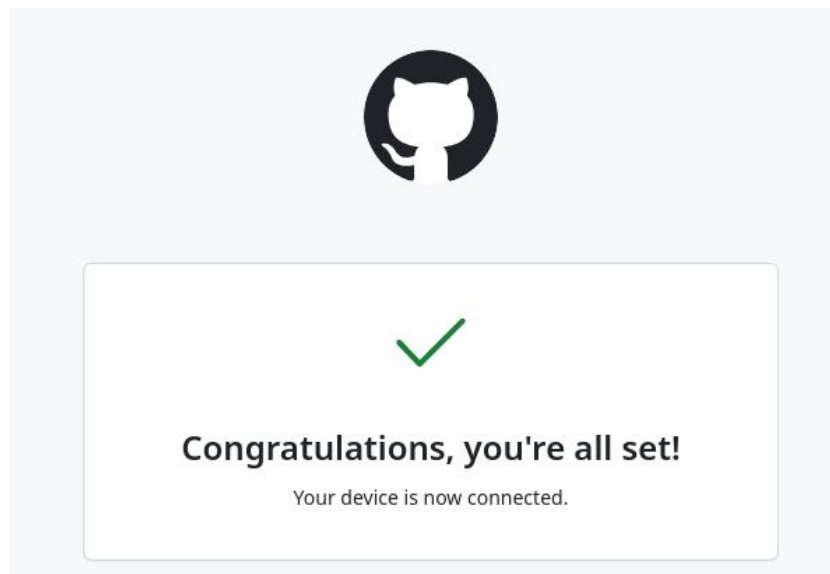


Рис. 3.11: 13

Видю сообщение об окончании авторизации под именем dserebryakova (рис. 3.12).

```
✓ Authentication complete.  
- gh config set -h github.com git_protocol https  
✓ Configured git protocol  
✓ Logged in as dserebryakova  
[dserebryakova@dserebryakova ~]$
```

Рис. 3.12: 14

3.7 Создание репозитория курса на основе шаблона

Создаю директорию, перехожу в нее, создаю репозиторий на основе шаблона репозитория и клонирую его к себе в директорию (рис. 3.13).

```
[diserebryakova@diserebryakova ~]$ mkdir -p ~/work/study/2024-2025/"Операционные системы"
[diserebryakova@diserebryakova ~]$ cd ~/work/study/2024-2025/"Операционные системы"
[diserebryakova@diserebryakova Операционные системы]$ gh repo create study_2024-2025_os-intro --template=yamadharma/course-directory-student-t
/ Created repository dserebryakova/study_2024-2025_os-intro on GitHub
https://github.com/dserebryakova/study_2024-2025_os-intro
[diserebryakova@diserebryakova Операционные системы]$ git clone --recursive https://github.com/dserebryakova/study_2024-2025_os-intro.git os-i
Клонирование в «os-intro»...
fatal: https://github.com/dserebryakova/study_2024-2025_os-intro.git/ недоступно: Could not resolve host: github.comdserebryakova
[diserebryakova@diserebryakova Операционные системы]$ git clone --recursive https://github.com/dserebryakova/study_2024-2025_os-intro.git os-i
Клонирование в «os-intro»...
remote: Enumerating objects: 36, done.
remote: Counting objects: 100% (36/36), done.
remote: Compressing objects: 100% (35/35), done.
Получение объектов: 100% (36/36), 19.37 Киб | 314.00 Киб/с, готово.
remote: Total 36 (delta 1), reused 21 (delta 0), pack-reused 0 (from 0)
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yamadharma/academic-presentation-markdown-template.git) зарегистрирован по пути «templat
Подмодуль «template/report» (https://github.com/yamadharma/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/home/diserebryakova/work/study/2024-2025/Операционные системы/os-intro/template/presentation»...
remote: Enumerating objects: 111, done.
remote: Counting objects: 100% (111/111), done.
remote: Compressing objects: 100% (77/77), done.
remote: Total 111 (delta 42), reused 100 (delta 31), pack-reused 0 (from 0)
Получение объектов: 100% (111/111), 102.17 Киб | 692.00 Киб/с, готово.
Определение изменений: 100% (42/42), готово.
Клонирование в «/home/diserebryakova/work/study/2024-2025/Операционные системы/os-intro/template/report»...
remote: Enumerating objects: 142, done.
remote: Counting objects: 100% (142/142), done.
remote: Compressing objects: 100% (97/97), done.
```

Рис. 3.13: 15

Перехожу в каталог курса, удаляю лишние файлы и создаю необходимые каталоги (рис. 3.14).

```
[diserebryakova@diserebryakova Операционные системы]$ cd os-intro
[diserebryakova@diserebryakova os-intro]$ ls
CHANGELOG.md config COURSE LICENSE Makefile package.json README.en.md README.git-flow.md README.md template
[diserebryakova@diserebryakova os-intro]$ rm package.json
[diserebryakova@diserebryakova os-intro]$ echo os-intro > COURSE
[diserebryakova@diserebryakova os-intro]$ make
Usage:
  make <target>

Targets:
  list           List of courses
  prepare        Generate directories structure
  submodule      Update submodules
[diserebryakova@diserebryakova os-intro]$
```

Рис. 3.14: 16

Добавляю все новые файлы и отправляю их на сервер (рис. 3.15).

```
[diserebryakova@diserebryakova os-intro]$ git add .
[diserebryakova@diserebryakova os-intro]$ git commit -am 'feat(main): make course structure'
[master 620d048] feat(main): make course structure
2 files changed, 1 insertion(+), 14 deletions(-)
delete mode 100644 package.json
[diserebryakova@diserebryakova os-intro]$ git push
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 954 байта | 954.00 Киб/с, готово.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/dserebryakova/study_2024-2025_os-intro.git
d9e6ad1..620d048 master -> master
[diserebryakova@diserebryakova os-intro]$
```

Рис. 3.15: 17

4 Выводы

При выполнении данной работы я изучила идеологию и применение средств контроля версий, освоила работу 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.

6 Список литературы

1. Лабораторная работа №2