

Отчет по лабораторной работе №2

Дисциплина: Операционные системы

Кириянова Екатерина Андреевна

Содержание

1	Цель работы	4
2	Задание	5
3	Теоретическое введение	6
4	Выполнение лабораторной работы	8
4.1	Установка программного обеспечения	8
4.2	Базовая настройка git	8
4.3	Создание ключей ssh	9
4.4	Создание ключей pgr	10
4.5	Добавление PGP ключа в GitHub	10
4.6	Настройка автоматических подписей коммитов git	11
4.7	Настройка gh	12
4.8	Шаблон для рабочего пространства	12
5	Выводы	14
	Список литературы	15

Список иллюстраций

4.1	Установка	8
4.2	Установка	8
4.3	Имя и email	8
4.4	Настройка	9
4.5	Параметры	9
4.6	SSH	9
4.7	SSH	10
4.8	PGP	10
4.9	Установка	10
4.10	Генерация	11
4.11	GPG	11
4.12	Подписи коммитов	11
4.13	Авторизация	12
4.14	Репозиторий	12
4.15	Каталог	12
4.16	Удаление	12
4.17	Создание	13
4.18	Установка	13

1 Цель работы

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

2 Задание

1. Установка программного обеспечения
2. Базовая настройка git
3. Создание ключей ssh
4. Создание ключей gpg
5. Добавление PGP ключа в GitHub
6. Настройка автоматических подписей коммитов git
7. Настройка gh
8. Шаблон для рабочего пространства

3 Теоретическое введение

Системы контроля версий. Общие понятия

Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещать изменения, произведённые разными участниками проекта, производить откат к любой более ранней версии проекта, если это требуется.

В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную ему версию файлов. После внесения изменений, пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент. Сервер может сохранять не полную версию изменённых файлов, а производить так называемую дельта-компрессию — сохранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных.

Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Можно объединить (слить) изменения, сделанные разными

участниками (автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменения. В зависимости от настроек блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая таким образом, привилегированный доступ только одному пользователю, работающему с файлом.

Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно доступна информация о том, кто из участников, когда и какие изменения вносил. Обычно такого рода информация хранится в журнале изменений, доступ к которому можно ограничить.

В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным.

Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.

4 Выполнение лабораторной работы

4.1 Установка программного обеспечения

Устанавливаю git (рис. 4.1).

```
eakiryanova@eakiryanova:~$ sudo dnf install git
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет "git-2.48.1-1.fc41.x86_64" уже установлен.
```

Рис. 4.1: Установка

Устанавливаю gh (рис. 4.2).

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

Рис. 4.2: Установка

4.2 Базовая настройка git

Задаю свои имя и email (рис. 4.3).

```
eakiryanova@eakiryanova:~$ git config --global user.name "Екатерина Кирьянова"
eakiryanova@eakiryanova:~$ git config --global user.email "scdddt27@gmail.com"
```

Рис. 4.3: Имя и email

Настраиваю utf-8 в выводе сообщений git (рис. 4.4).


```
eakiryanova@eakiryanova:~$ git config --global core.quotepath false
```

Рис. 4.4: Настройка

Задаю параметры autocrlf и safecrlf (рис. 4.5).

```
eakiryanova@eakiryanova:~$ git config --global core.autocrlf input
eakiryanova@eakiryanova:~$ git config --global core.safecrlf warn
```

Рис. 4.5: Параметры

4.3 Создание ключей ssh

Создаю ssh ключ по алгоритму rsa с размером 4096 бит (рис. 4.6).

```
eakiryanova@eakiryanova:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/eakiryanova/.ssh/id_rsa):
Created directory '/home/eakiryanova/.ssh'.
Enter passphrase for "/home/eakiryanova/.ssh/id_rsa" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/eakiryanova/.ssh/id_rsa
Your public key has been saved in /home/eakiryanova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:x4a9EoGa/8JnNREpd5D6tBD+1rI0YHnMJC1Z59izVhc eakiryanova@eakiryanova
The key's randomart image is:
+---[RSA 4096]---+
|      +o+.  E  |
|      .* B=.  .|
|      ...%.o+  .|
|      o B=*   + .|
|      o .SB=+o  |
|      .  +@o.   |
|      .. .+. =  |
|      o.o..     |
|      +.        |
+-----[SHA256]-----+
```

Рис. 4.6: SSH

Теперь по алгоритму ed25519 (рис. 4.7).

```
eakiryanova@eakiryanova:~$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/eakiryanova/.ssh/id_ed25519):
Enter passphrase for "/home/eakiryanova/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/eakiryanova/.ssh/id_ed25519
Your public key has been saved in /home/eakiryanova/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:zILLxzC4tFnAKL40g8998gCDte/BpbdTr/Yx7yqT70w eakiryanova@eakiryanova
The key's randomart image is:
+--[ED25519 256]--+
|  o.                |
| . . .o             |
| ... o +            |
| + o = B .          |
| o = + + S          |
| +. = + o.          |
| o+ .* o. oE         |
| .o.o+o.=o.+        |
| .+oo.B*oo          |
+----[SHA256]-----+
```

Рис. 4.7: SSH

4.4 Создание ключей pgr

Генерирую ключ (рис. 4.8).

```
eakiryanova@eakiryanova:~$ 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.
```

Рис. 4.8: PGP

4.5 Добавление PGP ключа в GitHub

Вывожу список ключей (рис. 4.9).

```
eakiryanova@eakiryanova:~$ 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
[keyboard]
-----
sec   rsa4096/24AFF74F98B82271 2025-03-02 [SC]
      A0D3D03CFDA1ED344881CA0D24AFF74F98B82271
uid   [ абсолютно ] Екатерина <scdddt27@gmail.com>
ssb   rsa4096/AC3BD6264C3C4FC8 2025-03-02 [E]
```

Рис. 4.9: Установка

Генерирую ключ (рис. 4.10).

```
eakiryanova@eakiryanova:~$ gpg --armor --export 24AFF74F98B82271
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGfEVA8BEADnjLU+iBF5M21pTKMaDvZKxQVWltBKIGWQVqqIDqX3JNzt+0rp
TB8mGVq5iuPG0umVyenHE6tCQPaEz66pLxm2w4hmD2xXtW1eTAbyYajPzeqRpmZn
VX0AzVAnn9VUZln4p5QWrK10JN0giRpv+yW/Sy03XeMm9eyNv2nKoN7oWz21b9pa
mrzWDAiAPjTGJAHvot3wLE0fkkdE0jp+QAwMWqyOM+1EADS2uayy3I6e/axLMsRY
JSovMi4YePCM0CqjMQKJsWS67ZNPvc937LdbP7mwYYkP4eqKpCZV+YfLPE6mHzk
GozGc0FkgS7aHd8yq+ZNo5xW74xd179gptk7wQwCot/m2CGVD+NrfBT1Fwh4joR/
qm0UzNo6rTy7zDMn6Lj05CzorYcQhLQp0Fi1GmCcQKTEWxGwKigxbhzHg59bfHhG
WBrMdh1B1FeoBywV44t+lzyRe8HwR5LRc33uQJ0oB4ueWqmyl/yjFFv7ZcNkLCjG
4c+kiVXpnWUk5GG3od6adt4sLIggY2eaimWl3v+ft2tnQAMGUI7kWr+74F/ZCYf0
lbmITvkiVCutFMESeyg/DdL0cuBau8KqNHgpHz011A4I66MEenEbwLCirmbtcQ2r
ajfViUamTG/JR8afs3YqubtgDyNzjNBkqpKzmTYizfI5w5WByGY32CvuVwARAQAB
tG50LjE6G6NLD8e+C10YDQwNC06LAcRHNiZG9keD13OGdtYXN0LmNub3R6LjE1
F5Sw5T
```

Рис. 4.10: Генерация

Создаю новый GPG ключ на GitHub (рис. 4.11).

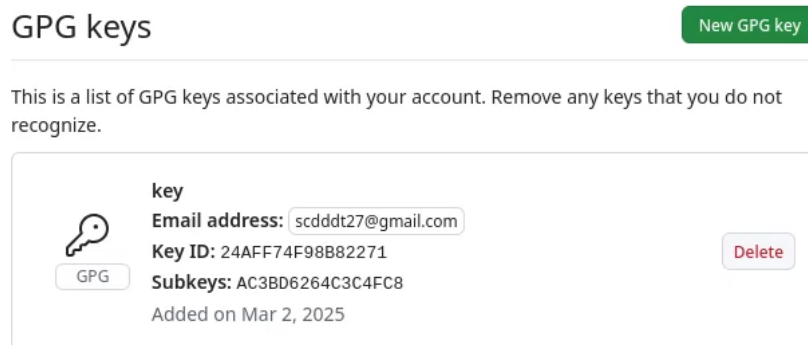


Рис. 4.11: GPG

4.6 Настройка автоматических подписей коммитов git

Указываю Git применять введенный email при подписи коммитов (рис. 4.12).

```
eakiryanova@eakiryanova:~$ git config --global user.signingkey 24AFF74F98B82271
eakiryanova@eakiryanova:~$ git config --global commit.gpgsign true
eakiryanova@eakiryanova:~$ git config --global gpg.program $(which gpg2)
eakiryanova@eakiryanova:~$ gh auth login
? Where do you use GitHub? [Use arrows to move, type to filter]
> GitHub.com
Other
```

Рис. 4.12: Подписи коммитов

4.7 Настройка gh

Авторизуюсь с помощью `gh auth login` (рис. 4.13).

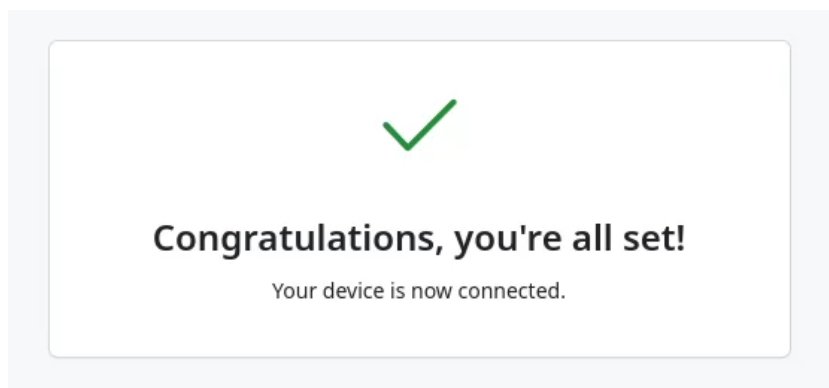


Рис. 4.13: Авторизация

4.8 Шаблон для рабочего пространства

Создаю репозиторий (рис. 4.14).

```
eakiryanova@eakiryanova:~$ mkdir -p ~/work/study/2024-2025/"Операционные системы"
eakiryanova@eakiryanova:~$ cd ~/work/study/2024-2025/"Операционные системы"
eakiryanova@eakiryanova:~/work/study/2024-2025/Операционные системы$ gh repo create study_2024-2025_os-intro --template=yamadharma/course-directory-student-template --public
✓ Created repository scxkl/study_2024-2025_os-intro on GitHub
https://github.com/scxkl/study_2024-2025_os-intro
eakiryanova@eakiryanova:~/work/study/2024-2025/Операционные системы$ git clone -recursive git@github.com:scxkl/study_2024-2025_os-intro.git os-intro
Клонирование в «os-intro»...
```

Рис. 4.14: Репозиторий

Перехожу в каталог курса (рис. 4.15).

```
eakiryanova@eakiryanova:~/work/study/2024-2025/Операционные системы$ cd study_2024-2025_os-intro
```

Рис. 4.15: Каталог

Удаляю лишние файлы (рис. 4.16).

```
eakiryanova@eakiryanova:~/work/study/2024-2025/Операционные системы/study_2024-2025_os-intro$ rm package.json
```

Рис. 4.16: Удаление

Создаю необходимые каталоги (рис. 4.17).

```
eakiryanova@eakiryanova:~/work/study/2024-2025/Операционные системы/study_2024-2025_os-intro$ echo os-intro > COURSE
eakiryanova@eakiryanova:~/work/study/2024-2025/Операционные системы/study_2024-2025_os-intro$ make
Usage:
  make <target>

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

eakiryanova@eakiryanova:~/work/study/2024-2025/Операционные системы/study_2024-2025_os-intro$ make prepare
```

Рис. 4.17: Создание

Отправляю файлы на сервер (рис. 4.18).

```
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/__init__.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/core.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/main.py
create mode 100644 project-personal/stage6/report/pandoc/filters/pandocxnos/pandocattributes.py
create mode 100644 project-personal/stage6/report/report.md
eakiryanova@eakiryanova:~/work/study/2024-2025/Операционные системы/study_2024-2025_os-intro$ git push
Перечисление объектов: 40, готово.
Подсчет объектов: 100% (40/40), готово.
Сжатие объектов: 100% (30/30), готово.
Запись объектов: 100% (38/38), 342.36 КиБ | 4.56 МиБ/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.
```

Рис. 4.18: Установка

5 Выводы

В ходе выполнения данной лабораторной работы я изучила идеологию и применение средств контроля версий и освоила умения по работе с git.

Список литературы

1.Лабораторная работа №2