Отчёт по лабораторной работе № 2

Королёв Иван Андреевич

Содержание

Список иллюстраций

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 Теоретическое введение

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

- Установка git (рис. 4.1).

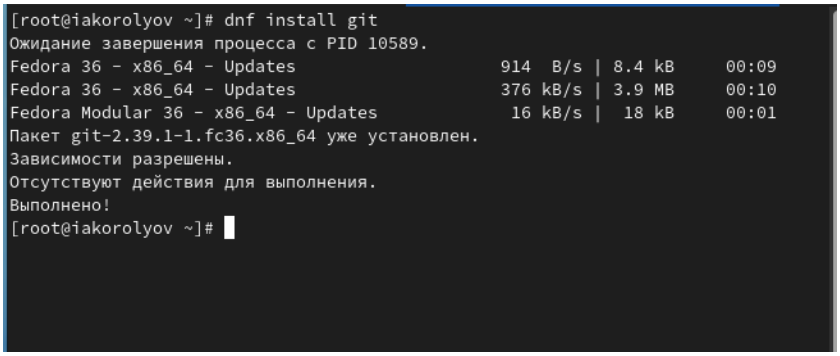

Рис. 4.1: git

- Установка gh (рис. 4.2).

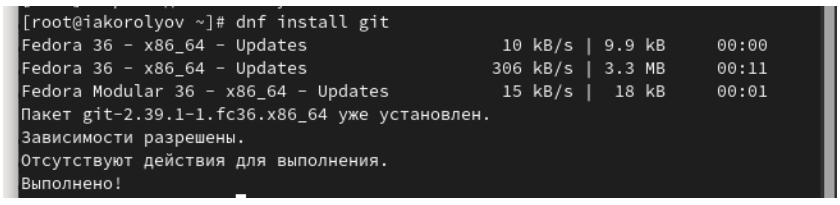

Рис. 4.2: gh

4.2 Базовая настройка git

- Зададим имя и email владельца репозитория (рис. 4.3).

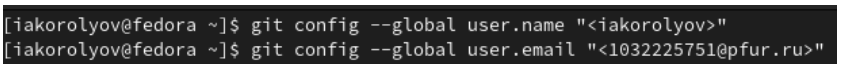

Рис. 4.3: name and email

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

Рис. 4.4: utf-8

- Зададим имя начальной ветки (будем называть её master) (рис. 4.5).

Рис. 4.5: master

- Параметр autocrlf (рис. 4.6).

Рис. 4.6: autocrlf

- Параметр safecrlf (рис. 4.7).

Рис. 4.7: safecrlf

4.3 Создайте ключи ssh

- Мы уже умеем создавать ssh из прошлого курса Архитектуры компьютеров. (рис. 4.8).

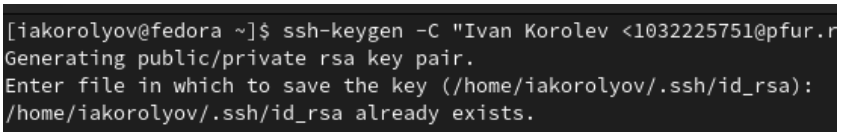

Рис. 4.8: ssh

4.4 Создайте ключи pgp

- Генерируем ключ (рис. 4.9), (рис. 4.10)

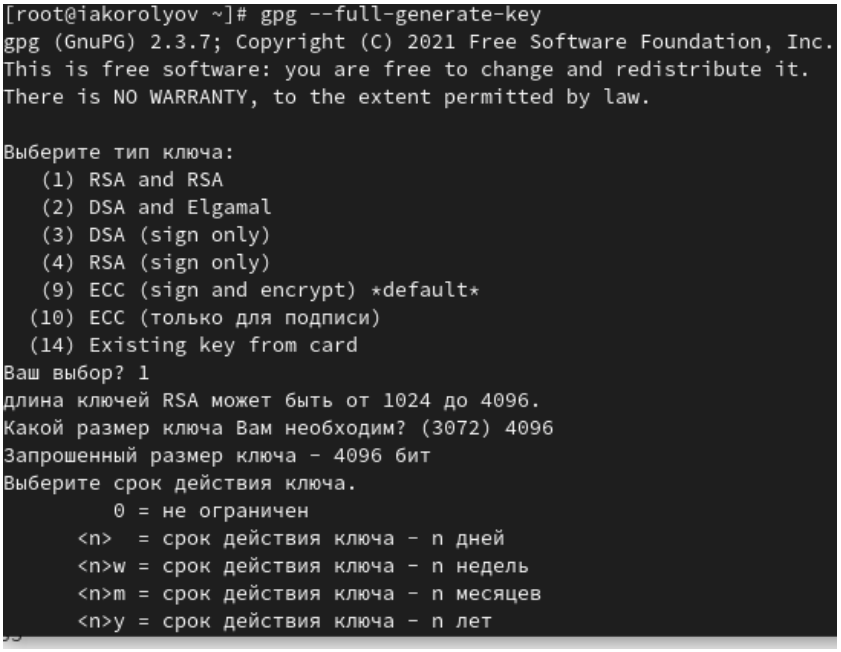

Рис. 4.9: pgp

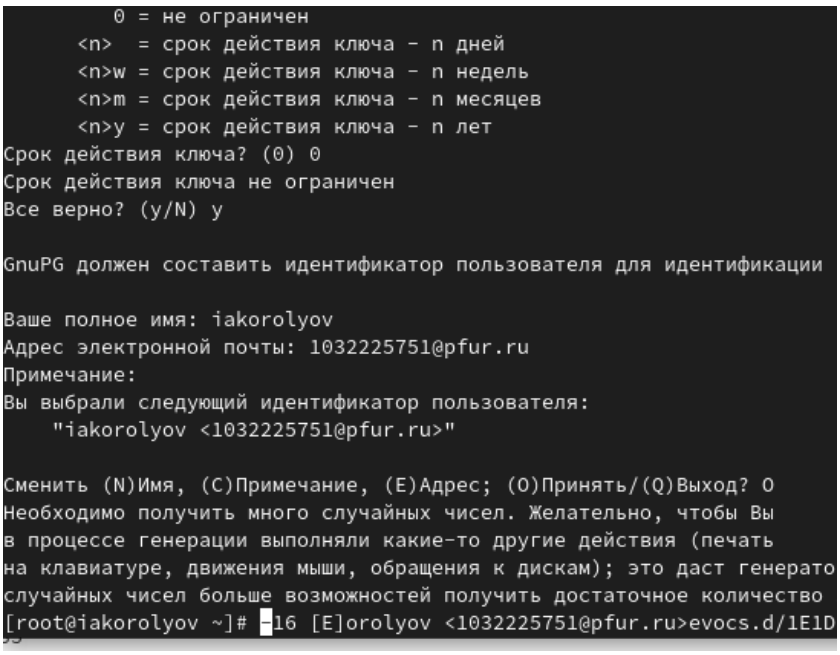

Рис. 4.10: pgp

4.5 Настройка github

- У меня создан репозиторий. (рис. 4.11)

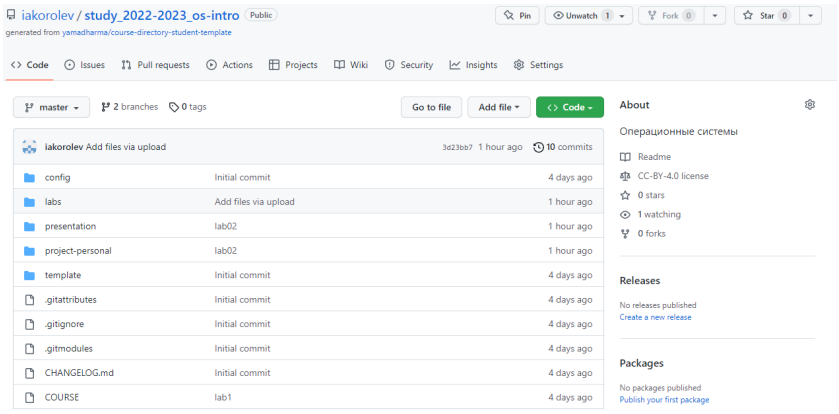

Рис. 4.11: github

4.6 Добавление PGP ключа в GitHub

- Выводим список ключей и копируем отпечаток приватного ключа (рис. 4.12)

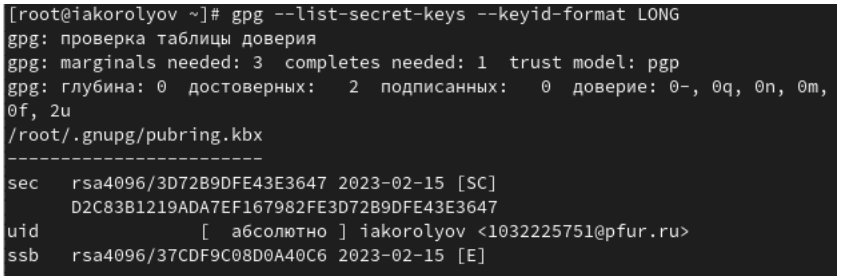

Рис. 4.12: key pgp

- Скопируйте ваш сгенерированный PGP ключ в буфер обмена (рис. 4.13)

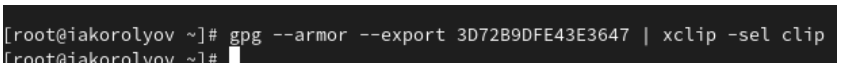

Рис. 4.13: copy pgp

- Перейдите в настройки GitHub (https://github.com/settings/keys), нажмите на кнопку New GPG key и вставьте полученный ключ в поле ввода (рис. 4.14)

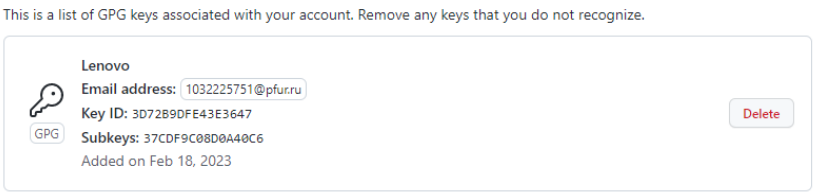

Рис. 4.14: pgp

4.7 Настройка автоматических подписей коммитов git

Используя введёный email, укажите Git применять его при подписи коммитов (рис. 4.15)

```
[root@iakorolyov ~]# git config --global user.signingkey 3D72B9DFE43E3647
[root@iakorolyov ~]# git config --global commit.gpgsign true
[root@iakorolyov ~]# git config --global gpg.program $(which gpg2)
```

Рис. 4.15: подписи коммитов

4.8 Настройка gh

- Авторизоваться в gh (рис. 4.16)

```
[root@iakorolyov ~]# gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations? SSH
? Generate a new SSH key to add to your GitHub account? No
? 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/t
okens
The minimum required scopes are 'repo', 'read:org'.
? Paste your authentication token: ****************************************
- gh config set -h github.com git_protocol ssh
✓ Configured git protocol
✓ Logged in as iakorolev
[root@iakorolyov ~]#
```

Рис. 4.16: gh

4.9 Создание репозитория курса на основе шаблона

- Создать шаблон рабочего пространства (рис. 4.17)

```
[root@iakorolyov ~]# gh repo create study_2022-2023_os-intro --template=yamadhar
ma/course-directory-student-template --public
```

Рис. 4.17: gh

- Репозиторий “Операционные системы” создан.

4.10 Настройка каталога курса

- Файлы на сервере (рис. 4.18), (рис. 4.19)

```
[iakorolyov@iakorolyov os-intro]$ rm package.json
rm: невозможно удалить 'package.json': Нет такого файла или каталога
[iakorolyov@iakorolyov os-intro]$ ls
CHANGELOG.md  labs        prepare           README.en.md       template
config        LICENSE     presentation      README.git-flow.md
COURSE        Makefile    project-personal  README.md
```

Рис. 4.18: Файлы

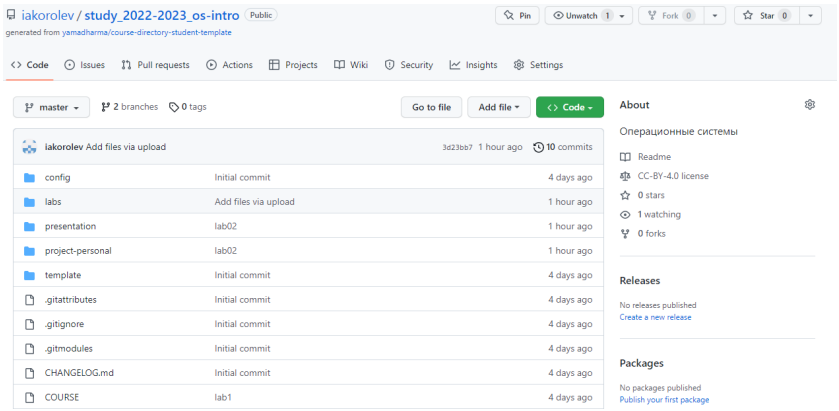

Рис. 4.19: Файлы

5 Выводы

Я освоил умения использования git.