

Лабораторная работа №2

Операционные системы

Александрова Ульяна Вадимовна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	Контрольные вопросы	13
6	Выводы	16

Список иллюстраций

4.1	Установка git	9
4.2	Установка gh	9
4.3	Генерация ключа ssh	10
4.4	Импорт ключа	10
4.5	Генерация gpg ключа	10
4.6	Вывод списка ключей	11
4.7	Копирование отпечатка ключа	11
4.8	Импорт	11
4.9	Настройка commit	11
4.10	Создание репозитория	11
4.11	Копирование репозитория	12
4.12	Название рисунка	12

Список таблиц

3.1	основные команды git	8
-----	--------------------------------	---

1 Цель работы

Целью работы является изучение идеологии и применение средств контроля версий, а также освоение 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.

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.

В табл. tbl. 3.1 приведены основные команды git.

Таблица 3.1: основные команды git

Команда	Описание
git init	Создание основного дерева репозитория
git pull	Получение обновлений (изменений) текущего дерева из центрального репозитория
git push	Отправка всех произведённых изменений локального дерева в центральный репозиторий
git status	Просмотр списка изменённых файлов в текущей директории
git diff	Просмотр текущих изменений
git add .	Добавить все изменённые и/или созданные файлы и/или каталоги
git commit	Сохранить добавленные изменения с внесением комментария через встроенный редактор
git checkout	Создание новой ветки, базирующейся на текущей

4 Выполнение лабораторной работы

Я установила git (рис. fig. 4.1).

```
[root@10 ~]# dnf install git
Последняя проверка окончания срока действия метаданных: 0:09:32 назад, Сб 18 фев 2023 13:15:22.
Пакет git-2.35.1-1.fc36.x86_64 уже установлен.
Зависимости разрешены.
Отсутствуют действия для выполнения.
Выполнено!
```

Рис. 4.1: Установка git

Далее установила gh (рис. fig. 4.2).

```
root@10:~
[root@10 ~]# dnf install git
Последняя проверка окончания срока действия метаданных: 0:09:32 назад, Сб 18 фев 2023 13:15:22.
Пакет git-2.35.1-1.fc36.x86_64 уже установлен.
Зависимости разрешены.
Отсутствуют действия для выполнения.
Выполнено!
[root@10 ~]# dnf install gh
Последняя проверка окончания срока действия метаданных: 0:09:48 назад, Сб 18 фев 2023 13:15:22.
Зависимости разрешены.
=====
Пакет      Архитектура  Версия      Репозиторий  Размер
=====
Установка:
gh          x86_64       2.23.0-1.fc36  updates      8.2 М
=====
Результат транзакции
=====
Установка 1 Пакет

Объем загрузки: 8.2 М
Объем изменений: 41 М
Продолжить? [д/н]:
```

Рис. 4.2: Установка gh

Так как я уже осуществила базовую настройку git в прошлом семестре, то я сразу перехожу к созданию ключа ssh (рис. fig. 4.3).

```
[root@10 ~]# ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (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:
SHA256:3XJ4T40o6530tQZgAqmKI/n5eooSSvFz+SeA0/M/7nk root@10.0.2.15
The key's randomart image is:
+---[RSA 4096]-----+
|
| . o
| . o .
|o.o . ..oo . o
|ooo. . So.= o .
|..o+ + *..o
|o. .+ o . .o
|o. +o +.E o o..
|o o+++=. .=.
+---[SHA256]-----+
```

Рис. 4.3: Генерация ключа ssh

И импортирую его в свои настройки github (рис. fig. 4.4).



Рис. 4.4: Импорт ключа

Затем я создаю gpg ключ (рис. fig. 4.5).

```
[yana@10 ~]$ gpg --full-generate-key
gpg (GnuPG) 2.3.4; Copyright (C) 2021 Free Software Foundation, Inc.
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/yana/.gnupg'
gpg: создан щит с ключами '/home/yana/.gnupg/pubring.kbx'
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)
```

Рис. 4.5: Генерация gpg ключа

Копирую ключ через специальные утилиты (рис. fig. 4.6), (рис. fig. 4.7).

```
[yana@10 ~]$ 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
/home/yana/.gnupg/pubring.kbx
-----
sec   rsa4096/006CEBDCD3C0A79C 2023-02-18 [SC]
      F96FB496DE2D4F151DE3BA57006CEBDCD3C0A79C
uid           [ абсолютно ] Uliana (meow) <nearrmee@gmail.com>
ssb   rsa4096/7E0C7E5C6BF90E7C 2023-02-18 [E]
```

Рис. 4.6: Вывод списка ключей

```
[yana@10 ~]$ gpg --armor --export F96FB496DE2D4F151DE3BA57006CEBDCD3C0A79C | xclip -sel clip
```

Рис. 4.7: Копирование отпечатка ключа

И экспортирую на сайт (рис. fig. 4.8).

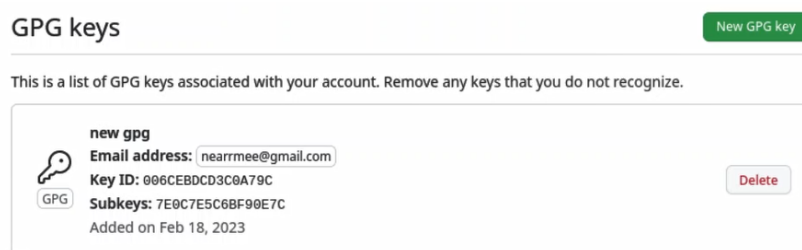


Рис. 4.8: Импорт

Я настроила автоматические подписи коммитов git (рис. fig. 4.9).

```
[yana@10 ~]$ git config --global user.signinkey F96FB496DE2D4F151DE3BA57006CEBDCD3C0A79C
[yana@10 ~]$ git config --global commit.gpgsign true
[yana@10 ~]$ git config --global gpg.program $(which gpg2)
```

Рис. 4.9: Настройка commit

Так как я уже работала с сервисом github, мне остается только создать репозиторий курса на основе шаблона (рис. fig. 4.10), (рис. fig. 4.11).

```
[yana@10 ~]$ mkdir -p ~/work/study/2022-2023/"Операционные системы"
[yana@10 ~]$ cd ~/work/study/2022-2023/"Операционные системы"
```

Рис. 4.10: Создание репозитория

```
[yana@10 Операционные системы]$ git clone --recursive git@github.com:Aleksandrov
aUV/study_2022-2023_os-intro.git os-intro
Клонирование в «os-intro»...
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 27 (delta 1), reused 11 (delta 0), pack-reused 0
Получение объектов: 100% (27/27), 16.94 КиБ | 199.00 КиБ/с, готово.
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yamadharma/academic-presen
tation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharma/academic-laboratory-r
eport-template.git) зарегистрирован по пути «template/report»
Клонирование в «/home/yana/work/study/2022-2023/Операционные системы/os-intro/te
mplate/presentation»...
```

Рис. 4.11: Копирование репозитория

Далее я настраиваю каталог курса и отправлю файлы на сервер при помощи утилит *git add .*, *git commit -am 'feat(main): make course structure*, *git push* (рис. fig. 4.12).

```
[yana@10 os-intro]$ rm package.json
[yana@10 os-intro]$ echo os-intro > COURSE
[yana@10 os-intro]$ make
```

Рис. 4.12: Название рисунка

5 Контрольные вопросы

1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются?

Системы контроля версий (VCS) применяются при работе нескольких человек над одним проектом. 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.

2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.

Хранилище - репозиторий, где хранится история изменений, а также док-ты, связанные с работой рабочего пространства.

Commit - изменения, внесенные пользователями, работающими в одном пространстве.

История - данные обо всех изменениях, внесенных в проект.

Рабочая копия - последняя версия проекта.

3. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.

Централизованные VCS - одно основное хранилище всего проекта. Каждый пользователь копирует себе необходимые ему файлы из этого репозитория, изменяет, затем добавляет изменения обратно в хранилище.

Децентрализованные VCS - у каждого пользователя свой вариант репозитория, есть возможность добавлять и забирать изменения из любого репозитория. В отличие от классических, в децентрализованных системах контроля версий центральный репозиторий не является обязательным.

4. Опишите действия с VCS при единоличной работе с хранилищем.

Как и при совместной работе, создается репозиторий, куда по мере продвижения проекта отправляются изменения.

5. Опишите порядок работы с общим хранилищем VCS.

Пользователю предоставляется доступ к одной из версий проекта, которую он может редактировать и сохранять изменения, доступные другим участникам проекта.

6. Каковы основные задачи, решаемые инструментальным средством git?

Предоставление удобства работы нескольких человек над одним проектом.

7. Назовите и дайте краткую характеристику командам git.

Команда	Описание
git init	Создание основного дерева репозитория
git pull	Получение обновлений (изменений) текущего дерева из центрального репозитория

Команда	
да	Описание
git push	Отправка всех произведённых изменений локального дерева в центральный репозиторий
git status	Просмотр списка изменённых файлов в текущей директории
git diff	Просмотр текущих изменений
git add .	Добавить все изменённые и/или созданные файлы и/или каталоги
git commit	Сохранить добавленные изменения с внесением комментария через встроенный редактор
git checkout	Создание новой ветки, базирующейся на текущей

10. Что такое и зачем могут быть нужны ветви (branches)?

Ветви нужны для того, чтобы программисты могли вести совместную работу над одним проектом вместе, но при этом не мешали друг другу.

6 Выводы

Я изучила идеологию и применение средств контроля версий, а также освоение умения по работе с git.