Отчет по лабораторной работе №2

Первоначальна настройка git

Акопян Сатеник Манвеловна

Содержание

# 1 Цель работы

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

# 2 Выполнение лабораторной работы

1.Установка программного обеспечения

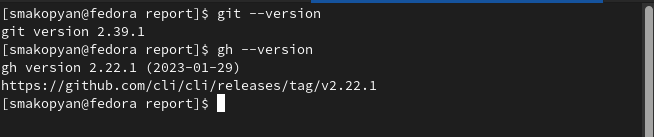


Figure 1: рисунок 1

2.Базовая настройка git.Сделаем предварительную конфигурацию git, а так же настроим utf-8 в выводе сообщений git и зададим имя начальной ветки (будем называть её master)

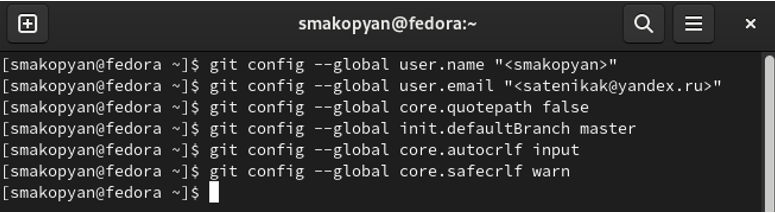


Figure 2: рисунок 2

3.Создание ключей ssh.



Figure 3: рисунок 3

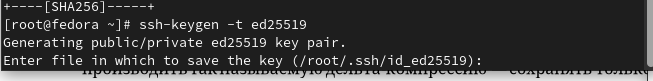


Figure 4: рисунок 4

4.Создание ключей pgp

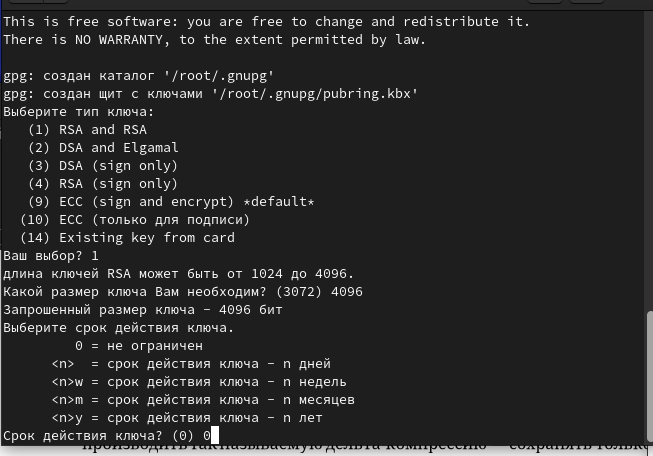


Figure 5: рисунок 5

5.Настройка github

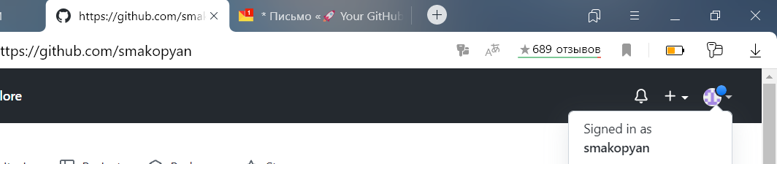


Figure 6: рисунок 6

6.Добавление PGP ключа в GitHub

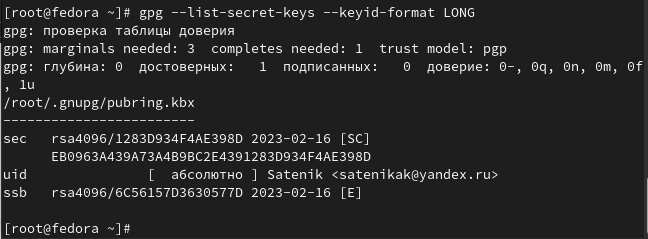


Figure 7: рисунок 7

Figure 8: рисунок 8

Figure 8: рисунок 8

7.Настройка автоматических подписей коммитов git

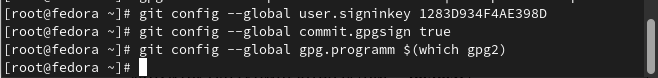


Figure 9: рисунок 9

8.Настройка gh

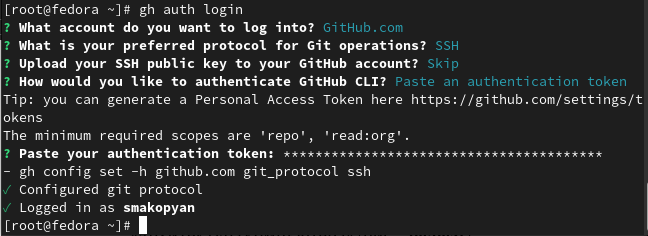


Figure 10: рисунок 10

9.Создание репозитория курса на основе шаблона

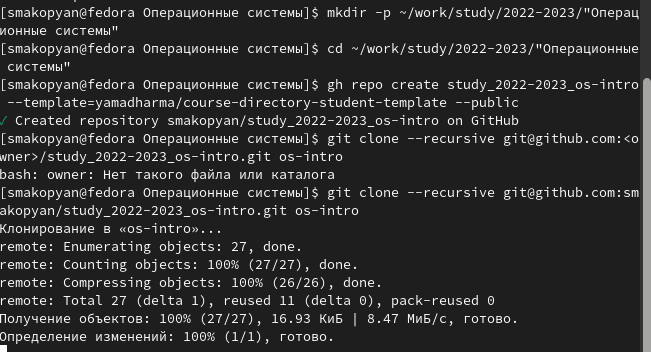


Figure 11: рисунок 11

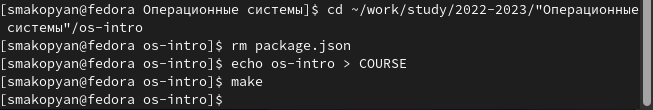
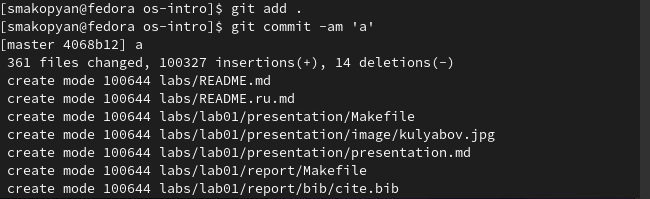


Figure 12: рисунок 12

 # Контрольные вопросы

1.Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются?  
Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом  
  
2.Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.  
Репозиторий - хранилище версий - в нем хранятся все документы вместе с историей их

изменения и другой служебной информацией

commit -редактирует репозиторий, заносит туда изменения

Рабочая копия - копия проекта, связанная с репозиторием

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 и, затем, добавляет свои изменения обратно  
Децентрализованные VCS: У каждого пользователя свой вариант (возможно не

один) репозитория, присутствует возможность добавлять и забирать изменения из любого репозитория

4.Опишите действия с VCS при единоличной работе с хранилищем.  
Cначала создаем и подключаем удаленный репозиторий, затем по мере изменения проекта отправляем проведенные изменения на сервер.   
  
5.Опишите порядок работы с общим хранилищем VCS.  
Пользователь перед началом работы получает нужную ему версию файлов, после внесения изменений оптравляет их на сервер, предыдущие версии не удаляются, к ним можно вернуться в любой момент.  
  
6.Каковы основные задачи, решаемые инструментальным средством git?  
Основными задачами, решаемыми инструментальным средством git являются: хранение информации о всех изменениях в коде, обеспечение удобства командной работы над кодом.  
  
7.Назовите и дайте краткую характеристику командам git.  
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.Приведите примеры использования при работе с локальным и удалённым репозиториями.  
  
9.Что такое и зачем могут быть нужны ветви (branches)?  
  
Ветви нужны для совместной работы над проектом.  
  
10.Как и зачем можно игнорировать некоторые файлы при commit?  
  
Т.к. во время работы над проектом могут создаваться файлы, которые не должны попасть в репозиторий (результаты сборки программ например), то можно создать файл .gitignore с перечислением шаблонов соответствующих таким файлам.

# 3 Выводы

В результате данной лабораторной работы, мы изучили идеологию и применение средств контроля версий, а также освоили умения по работе с git.

# Список литературы