Отчет по лабораторной работе №3

Смирнов Дмитрий Романович, НММбд-03-22

Содержание

# 1 Цель работы

Целью работы является изучить идеологию и применение средств контроля версий. Приобрести практические навыки по работе с системой git.

# 2 Выполнение лабораторной работы

Для начала проведу базовую настройку git (рис 1)

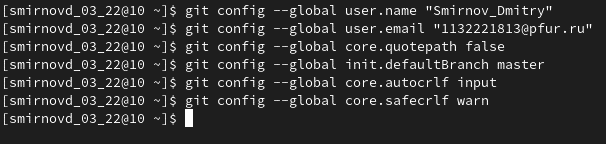


Рис. 1: Рис1

Сгенерирую ключ для последующей идентификации пользователя на сервере репозиториев (рис 2)

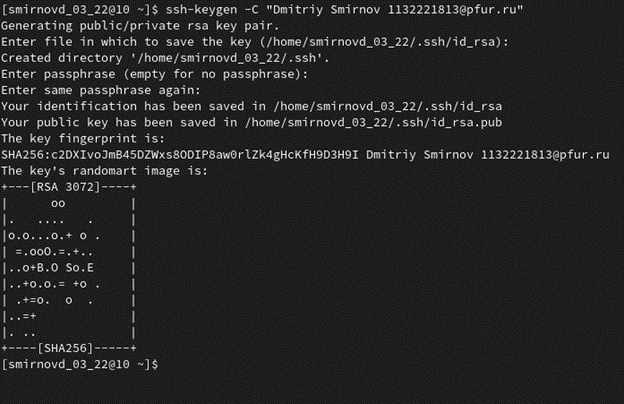


Рис. 2: Рис2

Скопирую ssh ключ и вставлю на GitHub (рис 3, рис 4)

Рис. 3: Рис3

Рис. 3: Рис3

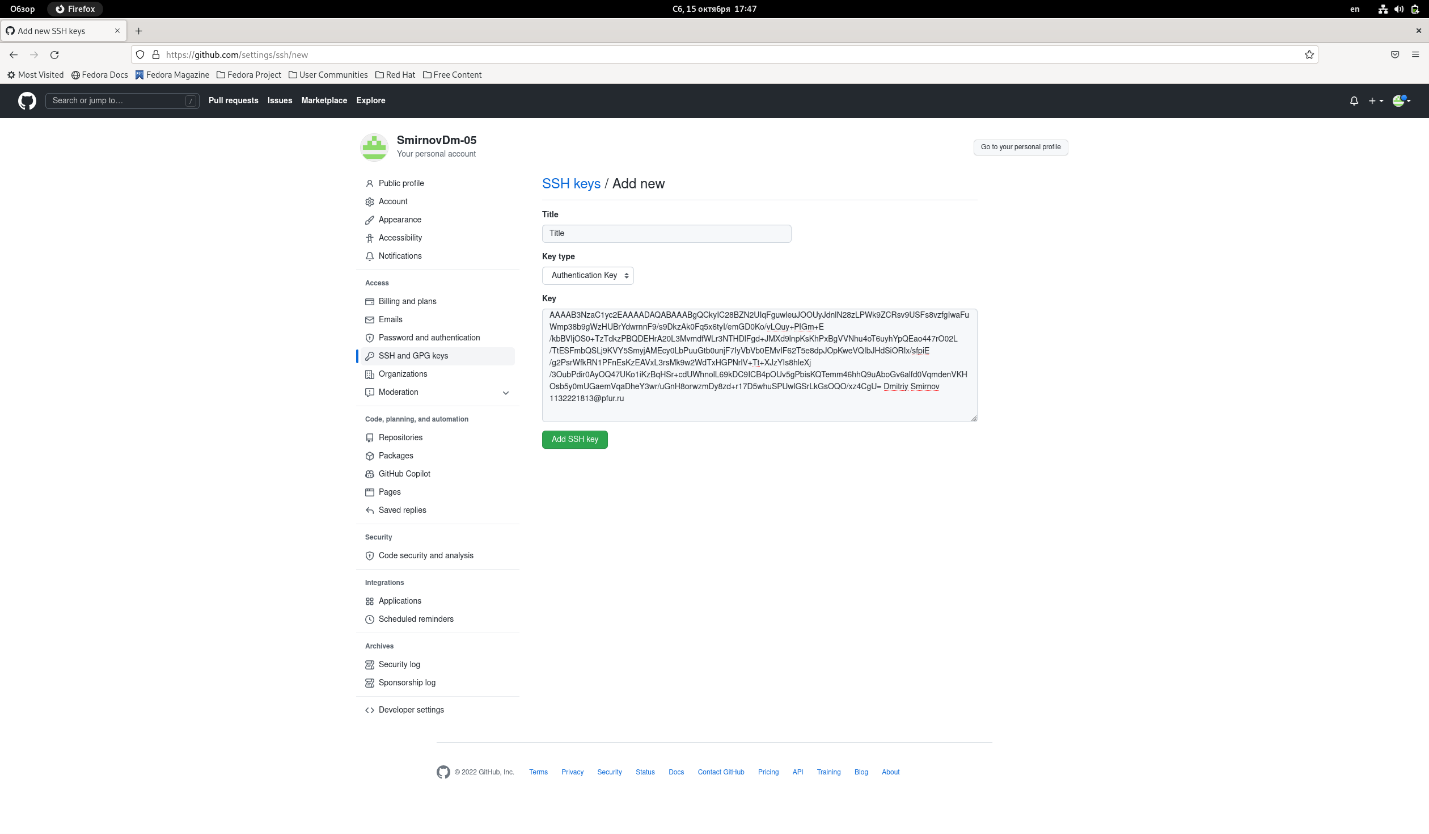


Рис. 4: Рис4

Создам каталог для предмета «Архитектура компьютера» (рис 5)

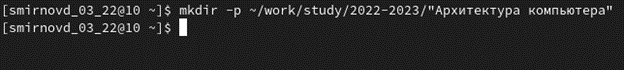


Рис. 5: Рис5

Создам репозиторий на основе предоставленного шаблона. Скопирую созданный репозиторий (рис 6)

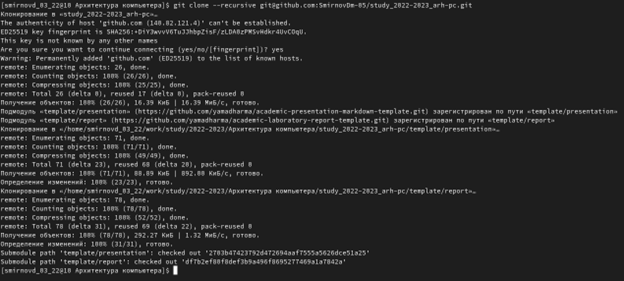


Рис. 6: Рис6

Перейду в каталог курса. Удалю файл package.json и создам COURSE (рис 7)

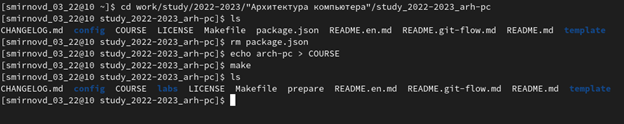


Рис. 7: Рис7

Отправлю файлы на сервер с помощью команд: git add . git commit -am ‘feat(main): make course structure’ git push

# 3 Контрольные вопросы для самопроверки

1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются? Система контроля версий — это система, регистрирующая изменения в одном или нескольких файлах с тем, чтобы в дальнейшем была возможность вернуться к определённым старым версиям этих файлов. Применяются при работе нескольких человек над одним проектом.
2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия. Хранилище, или репозитарий, — место хранения всех версий и служебной информации. Commit - зафиксированный набор изменений, который показывает, какие файлы изменились и что именно в них изменилось. Рабочая копия — текущее состояние файлов проекта, основанное на версии, загруженной из хранилища
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

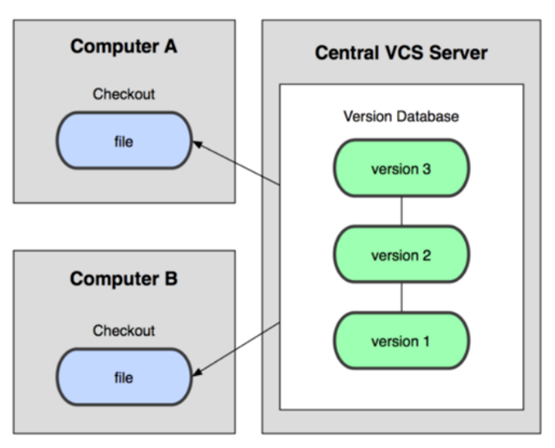


Рис. 8: Рис8

Децентрализованные VCS. У каждого пользователя свой вариант (возможно не один) репозитория. Присутствует возможность добавлять и забирать изменения из любого репозитория

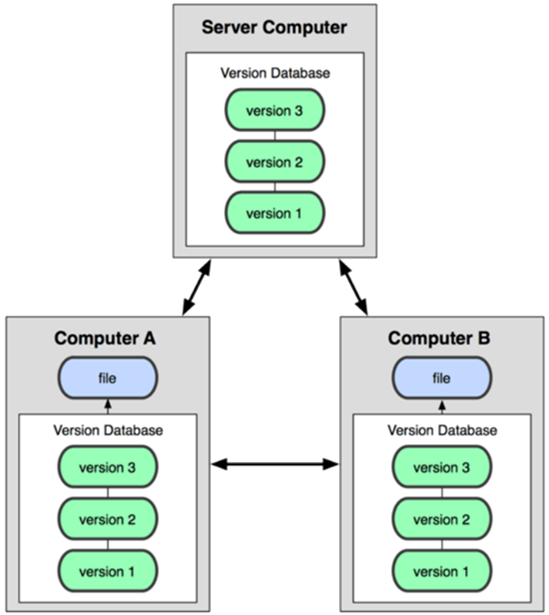


Рис. 9: Рис9

* 1. Опишите действия с VCS при единоличной работе с хранилищем.

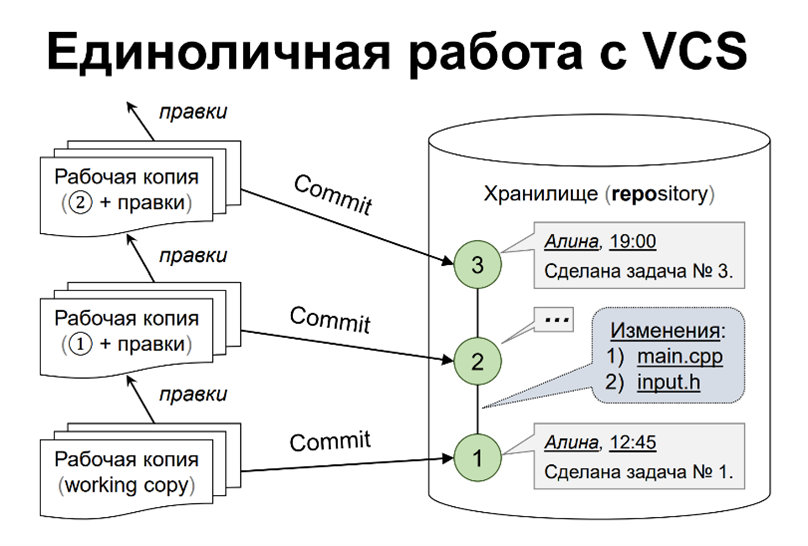


Рис. 10: Рис10

* 1. Опишите порядок работы с общим хранилищем VCS.

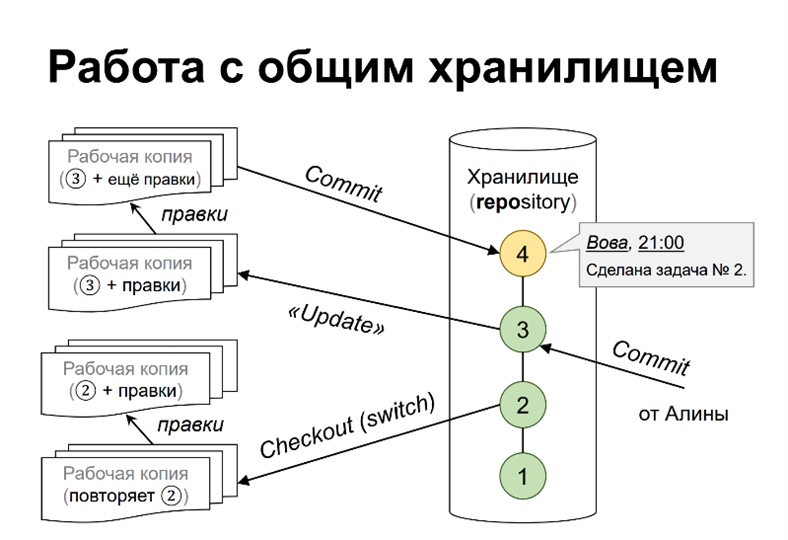


Рис. 11: Рис11

1. Каковы основные задачи, решаемые инструментальным средством git? У Git две основных задачи: первая — хранить информацию о всех изменениях в вашем коде, начиная с самой первой строчки, а вторая — обеспечение удобства командной работы над кодом.
2. Назовите и дайте краткую характеристику командам git. 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 изменения git add . добавить все изменённые и/или созданные файлы и/или каталоги git add имена\_файлов добавить конкретные изменённые и/или созданные файлы и/или каталоги git rm имена\_файлов удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории) git commit -am ‘Описание коммита’ сохранить все добавленные изменения и все изменённые файлы git checkout -b имя\_ветки создание новой ветки, базирующейся на текущей git checkout имя\_ветки переключение на некоторую ветку (при переключении на ветку, которой ещё нет в локальном репозитории, она будет создана и связана с удалённой) git push origin имя\_ветки отправка изменений конкретной ветки в центральный репозиторий git merge –no-ff имя\_ветки слияние ветки с текущим деревом git branch -d имя\_ветки удаление локальной уже слитой с основным деревом ветки git branch -D имя\_ветки принудительное удаление локальной ветки git push origin :имя\_ветки удаление ветки с центрального репозитория
3. Приведите примеры использования при работе с локальным и удалённым репозиториями. Команда git fetch связывается с удалённым репозиторием и забирает из него все изменения, которых у вас пока нет и сохраняет их локально. Команда git pull вначале забирает изменения из указанного удалённого репозитория, а затем пытается слить их с текущей веткой. Команда git push используется для установления связи с удалённым репозиторием, вычисления локальных изменений отсутствующих в нём, и собственно их передачи в вышеупомянутый репозиторий.

# 4 Выводы

Я изучил идеологию и применение средств контроля версий, а также приобрел практические навыки по работе с системой git.