РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Отчет

Лабораторная работа 2

*дисциплина: Архитектура компьютеров и операционные системы*

Накова Амина Михайловна

НПИбд-02-23

Москва

2023

Содержание:

1. Цель работы
2. Теоретическое введение
3. Выполнение лабораторной работы
4. Вывод
5. **Цель работы:**

Изучить идеологию и применение средств контроля версий, а так же приобрести навыки по работе с системой git.

1. **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.

# 3. Выполнение лабораторной работы

3.1 Настройка github

Существует несколько доступных серверов репозиториев с возможностью бесплатного размещения данных. Например, http://bitbucket.org/, https://github.com/ и https://gitflic.ru. Для выполнения лабораторных работ предлагается использовать Github. Создайте учётную запись на сайте https://github.com/ и заполните основные данные.

Сначала сделаем предварительную конфигурацию git. Откройте терминал и введите следующие команды, указав имя и email владельца репозитория:



Рис. 3.1.1 предварительная конфигурация git

Настроим utf-8 в выводе сообщений git.

Зададим имя начальной ветки (будем называть её master), параметр autocrlf и параметр safecrlf:

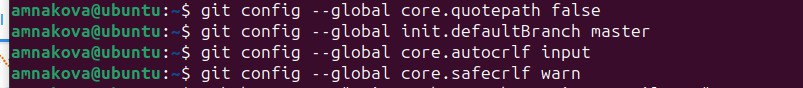


Рис. 3.1.2 настройка utf-8

3.2 Создание SSH ключа

Для последующей идентификации пользователя на сервере репозиториев необходимо сгенерировать пару ключей (приватный и открытый).

Ключи сохраняться в каталоге ~/.ssh/. Далее необходимо загрузить сгенерённый открытый ключ. Для этого зайти на сайт http: //github.org/ под своей учётной записью и перейти в меню Setting . После этого выбрать в боковом меню SSH and GPG keys и нажать кнопку New SSH key . Скопировав из локальной консоли ключ в буфер обмена, вставляем ключ в появившееся на сайте поле и указываем для ключа имя (Title).

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рис. 3.1.3 генерация ключа

Изображение выглядит как текст, снимок экрана, Шрифт, информация

Автоматически созданное описание

Рис. 3.1.4 копирование ключа

Название проекта на хостинге git имеет вид: study\_\_ Например, для 2023–2024 учебного года и предмета «Архитектура компьютера» (код предмета arch-pc) название проекта примет следующий вид: study\_2023–2024\_arch-pc

Изображение выглядит как текст, Шрифт, чек, снимок экрана

Автоматически созданное описание

3.1.5 создание ключа

Откройте терминал и создайте каталог для предмета «Архитектура компьютера»:



Рис. 3.1.6 создание каталога

Репозиторий на основе шаблона можно создать через web-интерфейс github.

Перейдите на станицу репозитория с шаблоном курса https://github.com/yamadharma/cour se-directory-student-template.

Далее выберите Use this template

В открывшемся окне задайте имя репозитория (Repository name) study\_2023–2024\_arhpc и создайте репозиторий (кнопка Create repository from template).

Изображение выглядит как текст, Шрифт, число, снимок экрана

Автоматически созданное описание

Рис. 3.1.7 создание репозитория

Откройте терминал и перейдите в каталог курса и клонируйте созданный репозиторий:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рис. 3.1.8 клонирование репозитория

3.2 Настройка каталога курса

Перейдите в каталог курса и удалите лишние файлы, создайте необходимые каталоги и отправьте файлы на сервер:

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рис. 3.2.0 настройка каталога курса

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рис. 3.2.1 настройка каталога курса