
Лабораторная работа №2

"Отчёт по лабораторной работе №2" "Система контроля версий Git" **Выполнил:** Куашев Бетал Муратович

Группа: НПИбд-02-24

Архитектура ЭВМ

Год: 2024

Цель работы

Цели работы - приобрести навыки по работе с системой контроля версий git и научиться пользоваться онлайн платформой GitHub.

Выполнение лабораторной работы

Благодаря ней, мы настроили вывод сообщений git в кодировке utf8. Теперь мы должны задать имя для начальной ветки. Мы назовем ее master :

Теперь введем следующую команду

Таким образом мы указываем git автоматически конвертировать CRLF окончания строк в LF во время commit'a. Теперь выполним вот эту команду (рис. 2.5). Она будет отвечать за то, чтобы печатать предупреждение в случае, если преобразования из CRLF будут необратимыми.

Для того, чтобы сервер мог идентифицировать пользователя, необходимо сгенерировать несколько ssh ключей.

Начнем с генерации открытого ключа. Для этого мы введем следующую команду, указав имя, фамилию пользователя и его адрес электронной почты в качестве аргумента

Зайдем на сайт GitHub Здесь мы должны перейти в пункт settings Находим раздел SSH and GPG keys и нажимаем New SSH key В предложенное поле "Key" нам необходимо вставить ключ, который мы только что сгенерировали

Для того, чтобы скопировать ключ для последующей вставки, нам необходимо ввести следующую команду

Команда "cat" прочитает данные из файла id_rsa.pub, а команда xclip вгрузит их в буфер обмена. Остается лишь вставить содержимое буфера обмена, то есть наш ключ, в предложенное поле "Key" и указать имя ключа в поле "Title"

Теперь нам необходимо организовать наше рабочее пространство. Для этого создадим каталог "Архитектура компьютера" по следующему адресу

После этого нам нужно будет создать репозиторий. Мы будем его создавать на основе шаблона.

Нажимаем на кнопку "Use this template", и в предложенных опциях выбираем "Create new repository". Далее нас переносит на следующую страницу. Здесь задаём имя нашего репозитория. Он будет называться так: study_2023-2024_arh-pc

Теперь нам нужно клонировать репозиторий на наш компьютер. Для этого перейдем в папку, в которую мы хотим скопировать репозиторий. В нашем случае это ранее созданная папка

Теперь перейдем непосредственно к клонированию. Для этого воспользуемся командой `git clone`, в аргументе указав ссылку на репозиторий. Ссылку можно найти при нажатии на кнопку код на странице нашего репозитория

Теперь перейдем к настройке клонированного каталога. Для начала перейдем в него с помощью команды `"cd"`

Удалим с помощью команды `"rm"` лишний файл

Теперь создадим необходимые файлы. В нашем случае, это будет файл `COURSE`. Чтобы его создать, мы воспользуемся командой `echo`, которая запишет в файл строку `"acrh-pc"` и автоматически создаст его, т.к. этого файла раньше не существовало

Теперь нам остается лишь отправить файлы на сервер. Для этого с помощью команды `git add` мы добавим каталоги, которые должны отправляться на сервер. В качестве аргумента мы возьмем точку, которая укажет на то, что мы должны отправить на сервер все файлы и каталоги, которые по иерархии находятся ниже нашего текущего расположения

Теперь с помощью команды `git commit` мы сохраним изменения и укажем комментарий, в котором будет поясняться, какие изменения мы сделали. В данном случае в комментарии мы напишем, что создали структуру курса

Теперь нам осталось окончательно загрузить изменения на сервер. Для этого мы воспользуемся командой `git push`

Остается лишь проверить, сохранились ли файлы на сервере. Сравним файлы, которые находятся на GitHub с теми файлами, что находятся на нашем компьютере

Как видим, все совпало.

Задание для самостоятельной работы

Теперь приступим к выполнению самостоятельной работы. Для начала мы создадим файл отчета для нашей лабораторной работы в папке `labs/lab02/report` с помощью LibreOffice

После этого скопируем отчет по нашей предыдущей лабораторной работе в соответствующую папку созданного нами рабочего пространства, то есть в папку `labs/lab01/report`. Для копирования воспользуемся командой `"cp"`

Теперь нам осталось лишь загрузить изменения на GitHub. Для этого нам нужно повторить те шаги, которые мы выполнили в предыдущих пунктах лабораторной работы, а именно, воспользоваться командой `"git add ."` для того, чтобы указать, что мы хотим сохранить изменения во всех файлах, находящихся в нашем каталоге

После этого с помощью команды `"git commit"` мы укажем комментарий и сохраним изменения. В комментарии мы укажем, что мы загрузили первую и вторую лабораторные работы

После этого вводим команду `"git push"` для того, чтобы загрузить наши файлы на GitHub

Остается лишь проверить, правильно ли мы все загрузили. Для этого посмотрим время обновления файлов в папке `labs` в GitHub

Выводы

В результате выполнения лабораторной работы появились практические навыки работы с системой контроля версий Git, была произведена её первоначальная настройка в Linux. Было изучено, как создавать репозитории, сохранять изменения и добавлять к ним комментарии, а также как выгружать файлы на сервер. Были приобретены навыки работы с платформой GitHub