

Отчет по лабораторной работе №2

Простейший вариант

Платонов Иван Георгиевич

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
3.1	Системы контроля версий. Общие понятия	7
3.2	Система контроля версий git	8
4	Выполнение лабораторной работы	9
4.1	Техническое обеспечение	9
4.2	Регистрация на GitHub	9
4.3	Локальная настройка git	9
5	Выводы	11
	Список литературы	12

Список иллюстраций

Список таблиц

1 Цель работы

Целями данной работы являются теоретическое и практическое ознакомление с системой контроля версиями (VSC – Version control system), а также наработка навыков в работе VSC git в связке с git сервер провайдером GitHub.

2 Задание

1. Техническое обеспечение;
2. Регистрация на GitHub;
3. Локальная настройка git;
4. Создание SSH-ключа для авторизации на GitHub;
5. Создание копии репозитория (форка – fork) на основе уже существующего репозитория;
6. Настройка каталога курса;
7. Задание для самостоятельной работы.

3 Теоретическое введение

3.1 Системы контроля версий. Общие понятия

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. Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно доступна информация о том, кто из участников, когда и какие изменения вносил. Обычно такого рода информация хранится в журнале изменений, доступ к которому можно ограничить. В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным. Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.

3.2 Система контроля версий git

Система контроля версий Git представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды `git` с различными опциями. Благодаря тому, что Git является распределённой системой контроля версий, резервную копию локального хранилища можно сделать простым копированием или архивацией.

4 Выполнение лабораторной работы

4.1 Техническое обеспечение

Так как работа производится на домашнем ПК, то необходимо установить git командой `sudo apt install git` (Linux Ubuntu).

4.2 Регистрация на GitHub

Для того, чтобы зарегистрировать новый аккаунт, переходим на официальный веб- сайт GitHub: <https://github.com>, и нажимаем кнопку sign up. Далее заполняем форму, проходим опрос и получаем свой аккаунт. Однако стоит заметить, что, так как GitHub принадлежит Microsoft и Microsoft запретила регистрацию на почтовые аккаунты доменных зон России (.ru, .bk, *.su и т.д.), то приходится регистрироваться на личную почту – на корпоративную почту РУДН в данный момент не получится. После успешной регистрации появится окно нового аккаунта.

Рисунок 3. Окно после успешной регистрации на GitHub

4.3 Локальная настройка git

Настройка осуществляется при помощи редактирования файла конфигурации git путем ввода команды `git config`. Ключ `-global` означает редактирование общего конфига; если необходимо редактировать только конфиг отдельно взятого репозитория, то стоит использовать ключ `-local`. Далее стоит рассмотреть

изменяемые параметры

5 Выводы

VCS git – одна из самых необходимых вещей в разработке. Именно с ее помощью большие команды разработчиков могут вместе работать над одним проектом. Стоит также заметить, что с помощью git и GitHub в связке можно не только работать одновременно над одним проектом, но и создавать сложные Developers Operations (DevOps), настраивать CI/CD операции и pipeline рабочего проекта, однако это уже не тема данной лабораторной работы. В данной лабораторной работе были изучены и проделаны базовые операции работы с git и GitHub, что, естественно, очень полезно для студента, не имевшего раньше опыт работы с данной VCS.

Список литературы

Edwards-Curve Digital Signature Algorithm (EdDSA), Internet Research Task Force (IRTF). (1 2017 г.). Получено из datatracker: <https://datatracker.ietf.org/doc/html/rfc8032>

Software Freedom Conservancy Inc. (2024). git documentation. Получено из git-scm: <https://git-scm.com/docs/git-config>

Демидова А. В. (б.д.). Лабораторная работа №1. Основы интерфейса командной строки ОС GNU Linux. В Д. А.В., Архитектура ЭВМ (стр. 1-13). Москва. Колисниченко Д.Н. (2023). Командная строка Linux. Санкт-Петербург: ВХВ-Петербург. Левицкий, Д. Н., & Завьялов, А. В. (2023). Сервер на Windows и Linux. Администрирование и виртуализация. Санкт-Петербург: Издательство Наука и Техника. Фишерман, Л. В. (2022). Git. Практическое руководство. Управление и контроль версий в разработке программного обеспечения. Санкт-Петербург: Издательство “Наука и Техника”. Чакон, С., & Штрауб, Б. (2022). Git для профессионального разработчика. Санкт- Петербург: Издание “Питер”.