Отчёт по лабораторной работе №2

Дисциплина: Архитектура компьютера.Операционные системы

Буриева Шахзода Акмаловна

Содержание

1 Цель работы

Целью данной работы является изучение идеологии и применение средств контроля версий,а также освоение умения по работе с git.

2 Задание

1. Создать базовую конфигурацию для работы с git.
2. Создать ключ SSH.
3. Создать ключ PGP.
4. Настроить подписи git.
5. Зарегистрироваться на Github.
6. Создать локальный каталог для выполнения заданий по предмету.

3 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.

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.

4 Выполнение лабораторной работы

Сперва делаем базовую настройку git,поэтому для этого задала имя и email владельца репозитория.

Настроила utf-8 в выводе сообщений git.

Задала имя начальной ветки (будем называть её master):

Далее вввела параметры autocrlf и safecrlf.

Потом создала ssh-ключ по алгоритму rsa с ключём размером 4096 бит.

Дальше по алгоритму ed25519.

Копирую ssh-ключ ее данной командой в буфер обмена, чтобы вставить на github аккаунт.

Вхожу в созданный акканут на github и создаю новый ssh-ключ для дальнейшей работы.

Для создания PGP ключа,сперва генерирую его определеннной командой.Из предложенных опций выбираю: 1. тип RSA and RSA; 2. размер 4096; 3. срок действия; значение по умолчанию — 0

Потом ввела: 1. Имя (не менее 5 символов). 2. Адрес электронной почты. 3. Комментарий.

Вывела список ключей и скопировала отпечаток приватного ключа

Скопировала сгенерированный PGP ключ в буфер обмена.

Вхожу в созданный акканут на github и создаю PGP-ключ для дальнейшей работы.

Используя введёный email, указала Git применять его при подписи коммитов

Далее для настройки gh,авторизовалась там,выбирая опции,которые предлагаются.Далее нажала Enter для авторизации через Браузер.

Потом ввела код,который требовался ввести и успешно авторизовалась.

Следующим шагом создала репозиторий курса на основе шаблона для 2023–2024 учебного года и предмета «Операционные системы» (код предмета os-intro).

Создала необходимые каталоги.

Последним шагом,я отправила файлы на сервер(github) при помощи данных команд.

5 Ответы на контрольные вопросы

1)Системы контроля позволяют хранить файлы не только на локальнлм компьютере,но и не на локальном.Это помогает в совместной работе с файлами или если нужно работать с ними на несколькоих устройствах.Также с помощью них можно создавать ветви,позволяющие сохранить рабочую версию и вносить изменения только в реставрацию.

2)Хранилище содержит в себе всю информацию о проекте: истории,коммиты,все файлы коммит-набор изменений и информация о них,а история-запись обо всех коммитах и рабочая копия-последняя версия,в которую вносят изменения.

3)Централизованные VCS позволяют позльзователю подключиться ко всему хранилищу и запросить только одну конкретную версию децентрализованные хранят в себе всю историю комммитов.

4)Сперва можно отправить измененную версию в репозиторий, склонировать существующий репозиторий, внести изменения, создать или удалить ветку.

5)Сначала пользователь сконирует себе последнюю (или нужную ему) версию, редактирует ее, отправляет обратно в репозиторий.

6)Основные задачи: возможность работать командой, возможность вернуться к старым версиям, можно иметь несколько “путей развития” проекта (на разных ветках) и выбирать оптимальный по итогу.

7)git add . – сохранить изменения в текущем каталоге git commit -am ‘anything’ – создание коммита и с коментарием git push – отправка коммита.

8)В большом проекте разработчики будут клонировать локальный репозиторий себе на устройства, благодоря чему смогут работать параллельно.Если же репозиторий слишком большой, разработчики могут подключаться к нему удаленно и править код в новых ветках и т.п.

9)Ветки (branches), которые позволяют сохранить рабочую версию и внести изменения только в копию. Это может быть нужно для избежания конфликта версий.

10)Обычно в игнорируемых файлах хранятся данные, которые не стоит выкладывать в общий доступ: пароли, ssh-ключи, базы данных.

6 Выводы

Изучили идеологии и применение средств контроля версий,а также освоили умения по работе с git.

Список литературы