Отчет по лабораторной работе 2

Операционные системы

Быкова Алина Александровна

Содержание

1 Цель работы

Изучить идеологию и применение средств контроля версий. Освоить умения по работе с git.

2 Выполнение лабораторной работы

Выполнила базовую настройку git.

Создала ssh ключ и вставила его в свой репозиторий.

Создаю ключ gpg: генерирую ключ(тип RSA and RSA;размер 4096;срок действия; значение по умолчанию — 0 (срок действия не истекает никогда).)

Настраиваю github (выполнила вход в свою учетную запись)

Вывела список ключей и скопировала отпечаток приватного ключа.

Скопировала свой сгенерированный PGP ключ в буфер обмена. Перешла в настройки GitHub и вставила полученный ключ в поле ввода.

Настроила автоматические подписи коммитов git.

Настроила gh (авторизовалась).

Создала репозиторий курса на основе шаблона. (“Операционные системы”)

Настроила каталог курса (перешла в каталог курса, удалила лишние файлы, создала необходимые каталоги, отправила файлы на сервер)

3 Выводы

Я научилась работать с git.

4 Ответы на контрольные вопросы

1. Системы контроля версий позволяют хранить файлы не только на локальном компьютере, но и не локально. Это помогает в совместной работе с файлами или если нужно работать с ними с нескольких устройств. Так же с их помощью можно создавать ветки (branches) которые позволяют сохранить рабочую версию и внести изменения только в копию.

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

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

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

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

6. 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.

Список литературы