

Отчёт по лабораторной работе №2

Борунов Семён Сергеевич

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Контрольные вопросы	7
4	Выводы	9
	Список литературы	10

Список иллюстраций

Список таблиц

1 Цель работы

получить навыки по работе с удаленным репозиторием git.

2 Выполнение лабораторной работы

Всю практическую часть я выполнял в первом семестре и детально изложил в файле `report_old.pdf`. Здесь посмотрим только на основные вехи: 1. установка пакета `git` на компьютер 2. Создание аккаунта на `github.com` 3. Создание и подключение `ssh`-ключа 4. создание и развертка репозитория из шаблона.

3 Контрольные вопросы

1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначены? 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. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия. 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. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида. централизованные VCS позволяют пользователю подключиться ко всему хранилищу и запросить только одну конкретную версию децентрализованные хранят в себе всю историю коммитов
4. Опишите действия с VCS при единоличной работе с хранилищем. можно отправить измененную версию в репозиторий, клонировать существующий репозиторий, внести изменения, создать или удалить ветку
5. Опишите порядок работы с общим хранилищем VCS. пользователь сконирует себе последнюю (или нужную ему) версию, редактирует ее, отправляет

обратно в репозиторий

6. Каковы основные задачи, решаемые инструментальным средством git? возможность работать командой, возможность вернуться к старым версиям, можно иметь несколько “путей развития” проекта (на разных ветках) и выбирать оптимальный по итогу
7. Назовите и дайте краткую характеристику командам git. `git add .` – сохранить изменения в текущем каталоге `git commit -am 'anything'` – создание коммита и с комментарием `git push` – отправка коммита
8. Приведите примеры использования при работе с локальным и удалённым репозиториями. В большом проекте разработчики будут клонировать локальный репозиторий себе на устройства, благодаря 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. Что такое и зачем могут быть нужны ветви (branches)? Ветки (branches) которые позволяют сохранить рабочую версию и внести изменения только в копию. Это может быть нужно для избежания конфликта версий.
10. Как и зачем можно игнорировать некоторые файлы при commit? обычно в игнорируемых файлах хранятся данные, которые не стоит выкладывать в общий доступ: пароли, ssh-ключи, базы данных.

4 Выводы

Были получены теоритические и практические навыки по работе с git и github.

Список литературы