Лабораторная работа №3. Система контроля версий Git

Российский университет дружбы народов

Максимиано Герра

Содержание

1	Цель работы	1
	Задание	
	Теоретическое введение	
	Выполнение лабораторной работы	
	Выводы	
Спи	сок литературы	7

1 Цель работы

Целью работы является изучить идеологию и применение средств контроля версий. Приобрести практические навыки по работе с системой git

2 Задание

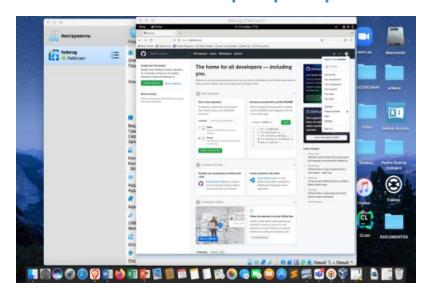
необходимо создать репозиторий в github, где находятся папки с соответствующими лабораториями, которые необходимо провести

3 Теоретическое введение

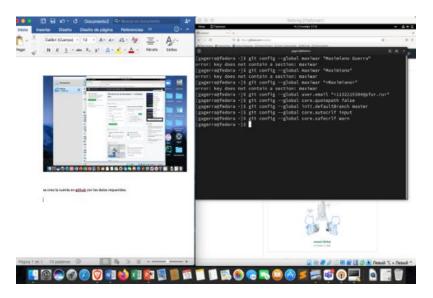
Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево про- екта хранится в локальном или удалённом репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля 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 со- хранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных. Системы контроля версий поддерживают возможность отслеживания и разре- шения 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 зависи- мости 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 изме- нений до точки ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно доступна информация о том, кто из участников, ко- гда 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 цен- тральный 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

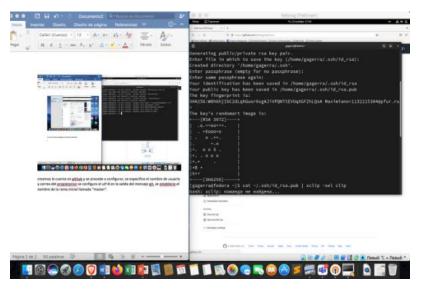
4 Выполнение лабораторной работы

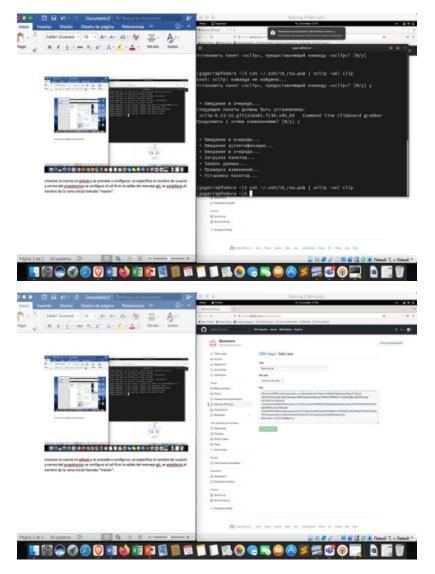


учетная запись github была создана с необходимыми данными

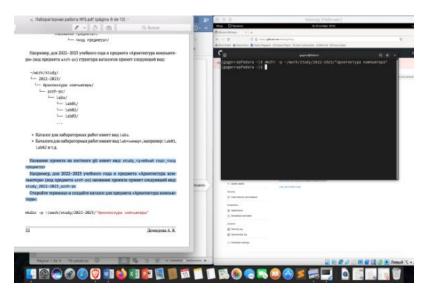


Мы создали учетную запись на github и приступили к ее настройке, указав имя пользователя и электронную почту владельца, настроив utf-8 в выводе сообщений git, установив имя начальной ветки под названием «master».

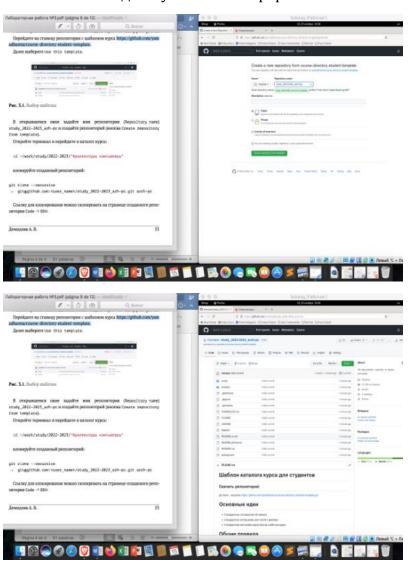




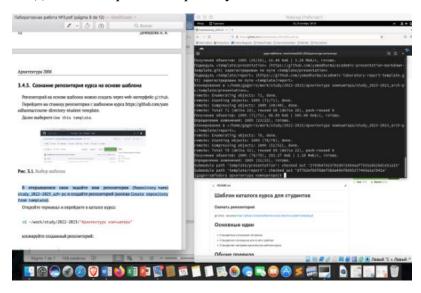
ключ был сгенерирован, скопирован с помощью команды <> и успешно вставлен на сайт github



Каталог был создан с указанной иерархией и именами.

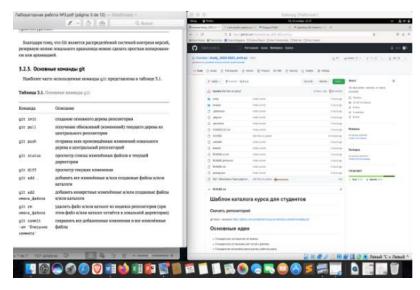


создан новый репозиторий с указанным именем



репозиторий был клонирован

- 1. Создайте отчет по выполнению лабораторной работы в соответствующем каталоге рабочего пространства (labs>lab03>report).
- 2. Скопируйте отчеты по выполнению предыдущих лабораторных работ в соответствующие каталоги созданного рабочего пространства.
- 3. Загрузите файлы на github.



файл был добавлен

5 Выводы

Github — это полезный инструмент, когда требуется групповая работа, он имеет то преимущество, что предоставляет ряд функций, облегчающих выполнение одной и той же работы, поэтому его могут использовать те, кто заслуживает выполнения важной работы, их информация хранятся одновременно

Список литературы

Не было вспомогательного материала