Лабораторная работа №1

1. Что такое СКВ и каково ее назначение?

Система контроля версий (СКВ) — это система, регистрирующая изменения в одном или нескольких файлах с тем, чтобы в дальнейшем была возможность вернуться к определённым старым версиям этих файлов

1. В чем недостатки локальных и централизованных СКВ?

Необходимость взаимодействовать с другими разработчиками

1. К какой СКВ относится Git?

РСКВ-распределенные системы контроля версий

4. В чем концептуальное отличие Git от других СКВ?

Основное отличие Git от любой другой СКВ (включая Subversion и её собратьев) — это подход к работе со своими данными. Концептуально, большинство других систем хранят информацию в виде списка изменений в файлах. Каждый раз, когда вы делаете коммит, то есть сохраняете состояние своего проекта в Git, система запоминает, как выглядит каждый файл в этот момент, и сохраняет ссылку на этот снимок

5.Как осуществляется целостность данных в GIT

В Git для всего вычисляется хеш-сумма, и только потом происходит сохранение. В дальнейшем обращение к сохранённым объектам происходит по этой хеш-сумме. Это значит, что невозможно изменить содержимое файла или директории так, чтобы Git не узнал об этом.

6.В каких состояниях могут находиться файлы в GID? как связаны эти состояния?

Существует 3 состояния, зафиксированный, изменены и подготовленный

7. Что такое профиль пользователя в GITHUB?

Профиль— это ваша публичная страница в которой вы делитесь своим опытом.

8. какие бывают какие бывают репозитории?  
 Публичные и закрытые

9. Укажите основные этапы модели работы с GIT?

Локальный контроль версий, централизованный контроль версий и распределенной контроль версий

10. Как осуществляет это первоначальная настройка GIT после установки?

Регаемся на GitHub,создаем репозиторий на их ресурсе после чего качаем утилиту.  
После скачки утилиты даем имя и привязываем почту к утилите.

Копируем ссылку на репозиторий и прописываем путь к нужному источнику.  
 Привязываем наш репозиторий к рабочей ветке

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Автоматически созданное описание

gИзображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Автоматически созданное описание

12. Какие типы лицензий поддерживается в GitHub при создании репозитория

AFL 3.0, Apache, artistic, BSL, BSD, WTFPL, итд   
 13. Как осуществляется клонирование репозитория? Зачем нужно клонировать репозиторий?

Зайти на гитхаб, скопировать ссылку на источник, прописать путь к репозиторию с помощью утилиты GIT и проверить статус, это нужно делать для того что бы проект который мы реализуем мог сохраниться с нужной нам версией в репозитории (ресурсе гитхаба) с целью последующего использования более ранних версий нашего кода, а так же для обсуждения или доработки при необходимости при помощи сообщества которое так же делится знаниями на ресурсе.

14. Как проверить состояние локального репозитори?,

Прописать команду в терминале git status

15. . Как изменяется состояние локального репозитория Git после выполнения следующих операций: добавления/изменения файла в локальный репозиторий Git; добавления нового/ измененного файла под версионный контроль с помощью команды git add ; фиксации (коммита) изменений с помощью команды git commit и отправки изменений на сервер с помощью команды git push

При добавлении изменений файла в репозитории добавляется в наш локальный репозиторий ,прик помощи команды git add добавляем сразу все измененные файлы, при помощи команды git commit фиксируем изменения локальном репозиторий с комментарием

При помощи команды push вновь им изменения в исходный репозиторий  
 16. У Вас имеется репозиторий на GitHub и два рабочих компьютера, с помощью которых Вы можете осуществлять работу над некоторым проектом с использованием этого репозитория. Опишите последовательность команд, с помощью которых оба локальных репозитория, связанных с репозиторием GitHub будут находиться в синхронизированном состоянии. Примечание: описание необходимо начать с команды git clone .

Копируем ссылку на репозиторий после чего в терминале прописываем git clone   
 С помощью команды мы имеем локальный репозиторий, все изменения которые будут проводится в коде, будут сохраняться в локальном источнике

Порядок команд   
 git status  
 git add README.md  
git add (для внесения всех изменений в локальный репозиторий)  
мы внесли изменения в локальный репозиторий, теперь вносим в исходный репозиторий   
git push –set-upstream origin edit-readme   
мы создали локальную ветку в удаленном источнике (РЕПОЗИТОРИИ)  
после этой команды для изменения и сохранения команды вводим команду:  
git push и все сохраняется уже удаленном репозитории.

17. GitHub является не единственным сервисом, работающим с Git. Какие сервисы еще Вам известны? Приведите сравнительный анализ одного из таких сервисов с GitHub.

CVS, Subversion, Perforce, Bazaar

18. Интерфейс командной строки является не единственным и далеко не самым удобным способом работы с Git. Какие Вам известны программные средства с графическим интерфейсом пользователя для работы с Git? Приведите как реализуются описанные в лабораторной работе операции Git с помощью одного из таких программных средств  
 gitignore