

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.

2. В чем недостатки локальных и централизованных СКВ?

Недостатком локальных СКВ является отсутствие возможности взаимодействия с другими разработчиками. Проблемой же централизованной СКВ является единая точка отказа.

3. К какой СКВ относится Git?

Git относится к распределенным СКВ

4. В чем концептуальное отличие Git от других СКВ?

Git не хранит и не обрабатывает данные в виде списка изменений в файлах. Вместо этого, подход Git к хранению данных больше похож на набор снимков миниатюрной файловой системы. Каждый раз, когда делается коммит, система запоминает, как выглядит каждый файл в этот момент, и сохраняет ссылку на этот снимок. Git представляет свои данные как «поток снимков»

5. Как обеспечивается целостность хранимых 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. Данная функциональность встроена в Git на низком уровне и является неотъемлемой частью его философии

6. В каких состояниях могут находиться файлы в Git? Как связаны эти состояния?

Зафиксированное, изменённое и подготовленное

7. Что такое профиль пользователя в GitHub?

Профиль - это публичная страница на GitHub

8. Какие бывают репозитории в GitHub?

Локальные и удаленные

9. Укажите основные этапы модели работы с GitHub.

- Клонирование репозитория

- Локальное изменение содержимого
- Запрос на извлечение для переноса локальных данных в исходный репозиторий

10. Как осуществляется первоначальная настройка Git после установки?

К первоначальной настройке Git относятся указания имени профиля GitHub и адреса электронной почты

11. Опишите этапы создания репозитория в GitHub.

- Переход в соответствующий раздел на GitHub
- Указание имени репозитория (должно быть уникальным для репозитория на аккаунте)
- Заполнение описания (можно оставить пустым)
- Выбор публичности репозитория
- Оформление .gitignore

12. Какие типы лицензий поддерживаются GitHub при создании репозитория?

GitHub поддерживает лицензии OpenSource

13. Как осуществляется клонирование репозитория GitHub? Зачем нужно клонировать репозиторий?

Ссылка на репозиторий копируется на самом GitHub, после чего в командной строке или терминале вводится команда `git clone` с ссылкой на репозиторий.

Клонировать репозиторий необходимо для внесения локальных изменений.

14. Как проверить состояние локального репозитория Git?

Состояние локального репозитория проверяется командой `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`?

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`.

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

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

