МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет информационных технологий и робототехники

Кафедра программного обеспечения информационных систем и технологий

**Отчет**

**по лабораторной работе № 1**

**«Системы контроля версий»**

по дисциплине: «Конструирование программного обеспечения»

|  |  |
| --- | --- |
| Выполнили**:** | ст.гр. 10701322 |
|  | Махонина Ю. В. |
|  | Свиржевская В.В |
|  |  |
|  |  |
| Приняла: | Станкевич С.Н. |

Минск 2024

**Лабораторная работа №1a**

**СИСТЕМЫ КОНТРОЛЯ ВЕРСИЙ GIT GUI**

**Цель работы**

Ознакомиться с системами контроля версий для разработки программного обеспечения. Освоить способы организации работы в системах контроля версий. Получить навыки командной работы.

**УПРАЖНЕНИЯ**

*Упражнение №1. Установка Git, работа с Git GU*

1. Установка Git

Git был успешно установлен:

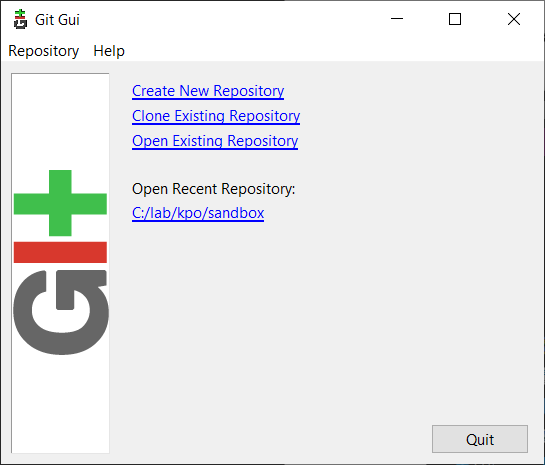


Рисунок 1 – Git Gui

1. Создание хранилища

С помощью «Create New Repository» хранилище «sandbox» создано. Папка хранилища:

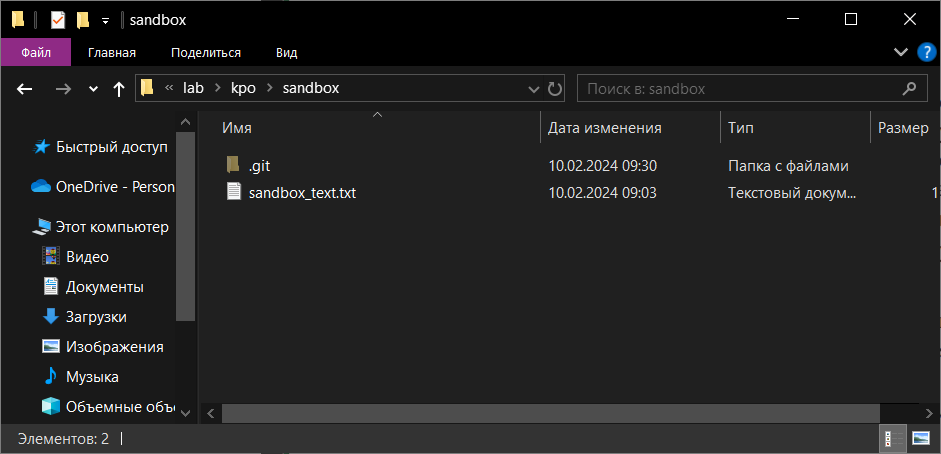


Рисунок 2 – Папка хранилища

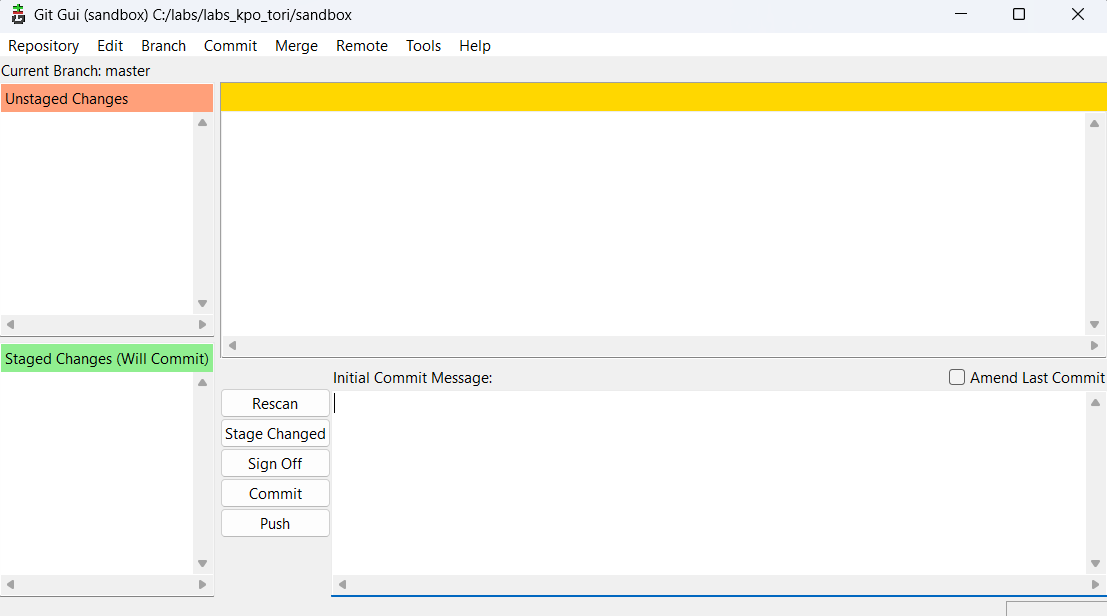
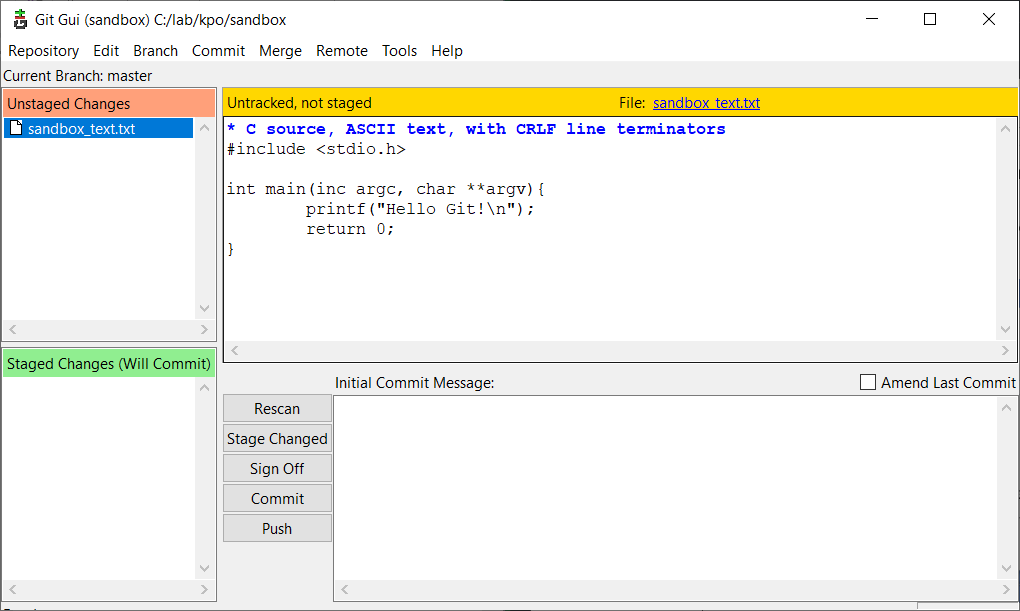


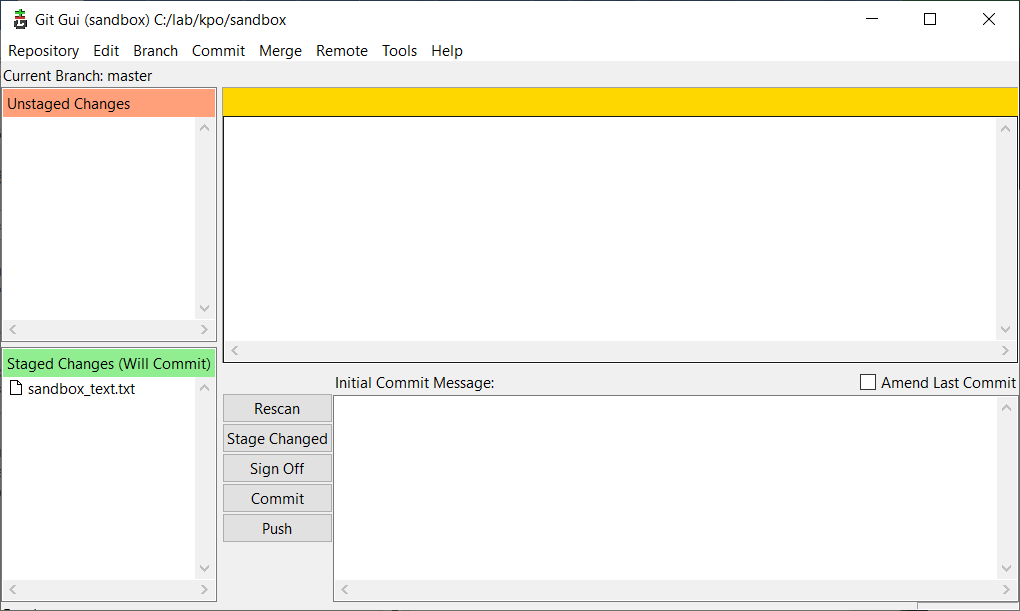
Рисунок 3 – Главный интерфейс

1. Закрепление изменений (commiting)

Хранилище создано, поэтому теперь в папке создается текстовый файл с некоторым содержанием. Редактирование было выполнено в блокноте. После нажатия кнопки «Rescan» мы получили следующее:

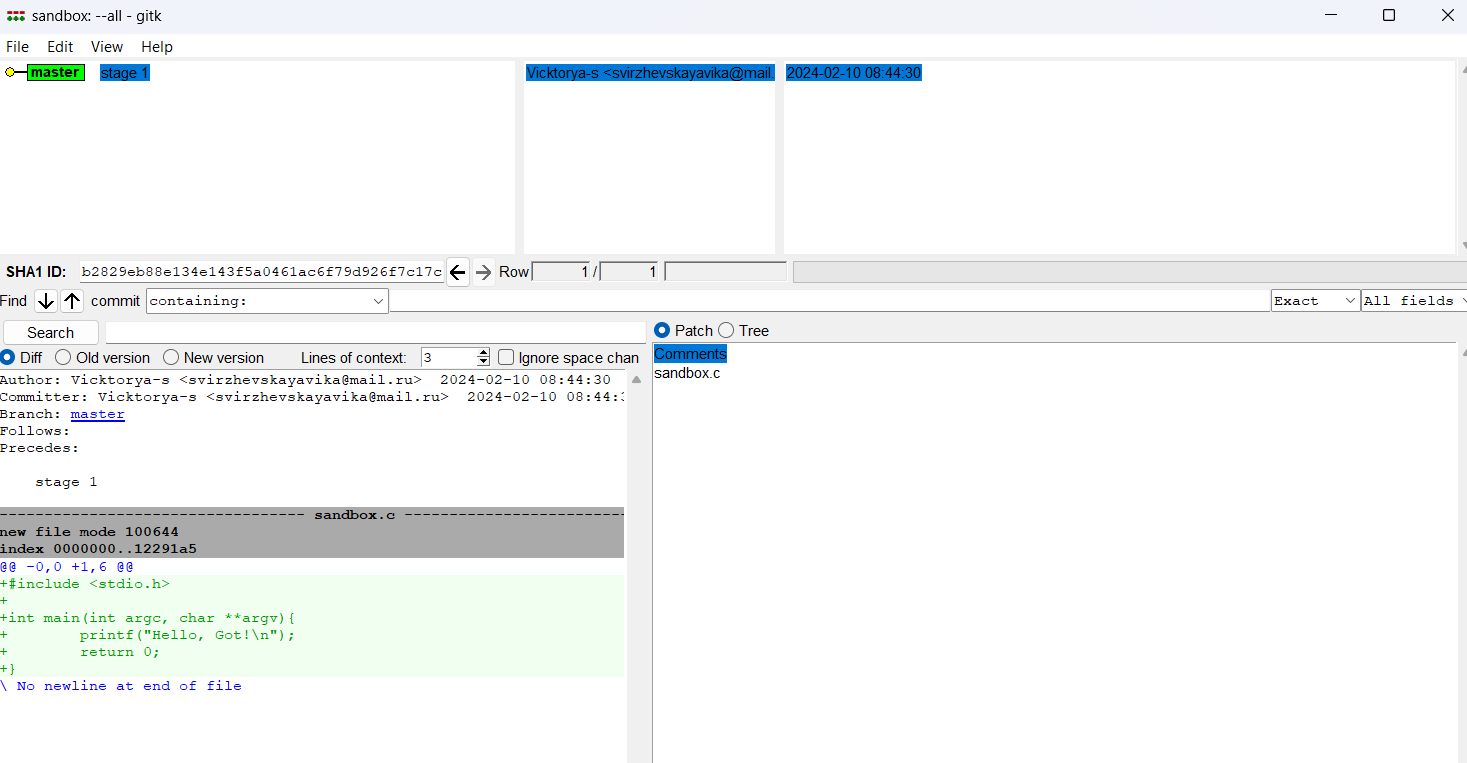


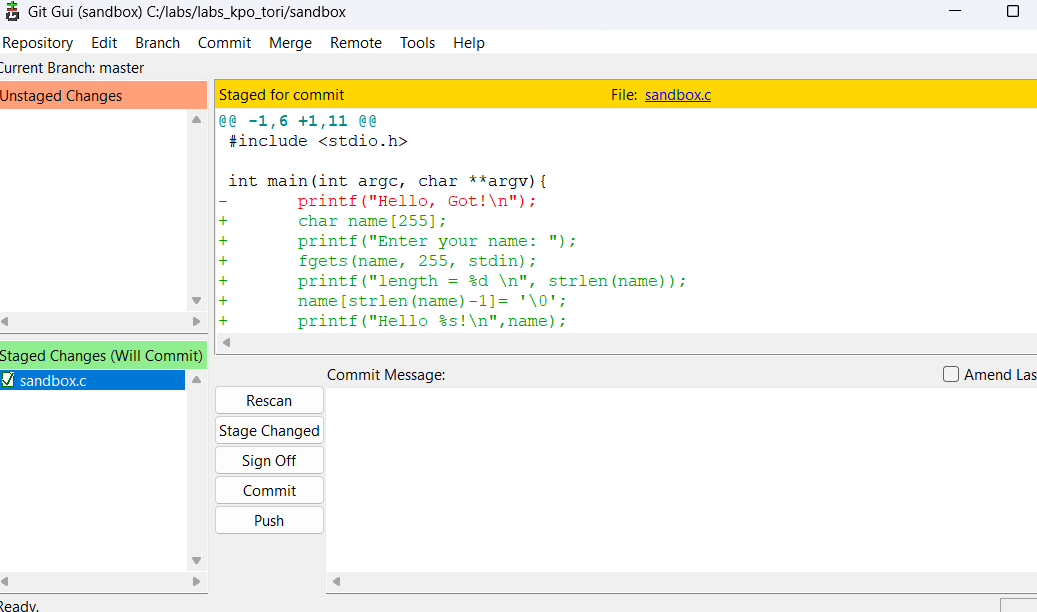
Перенесем файл в нижнее окно:

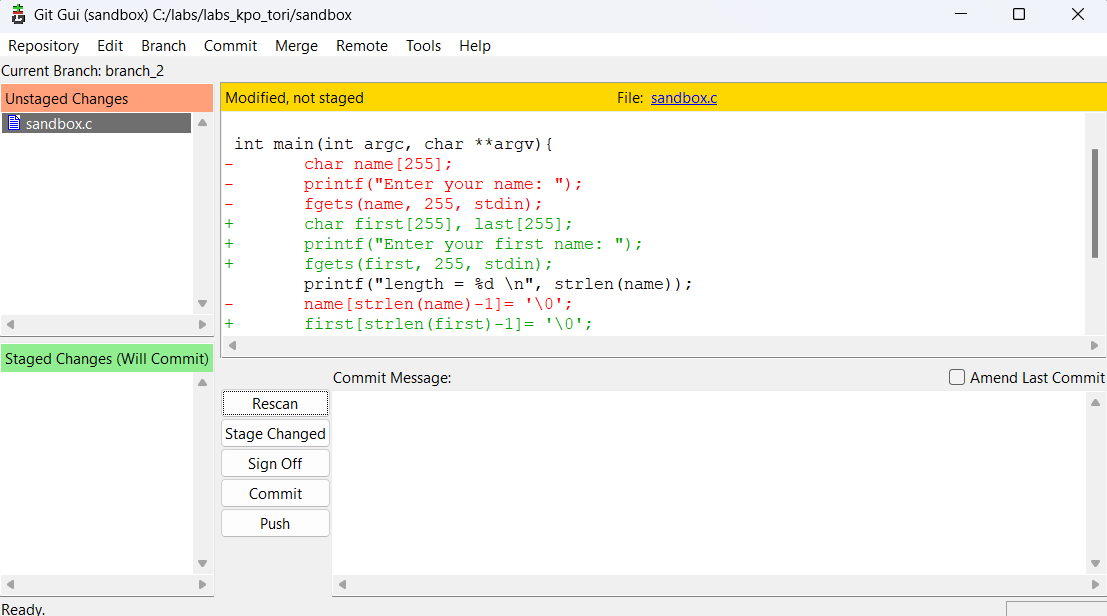


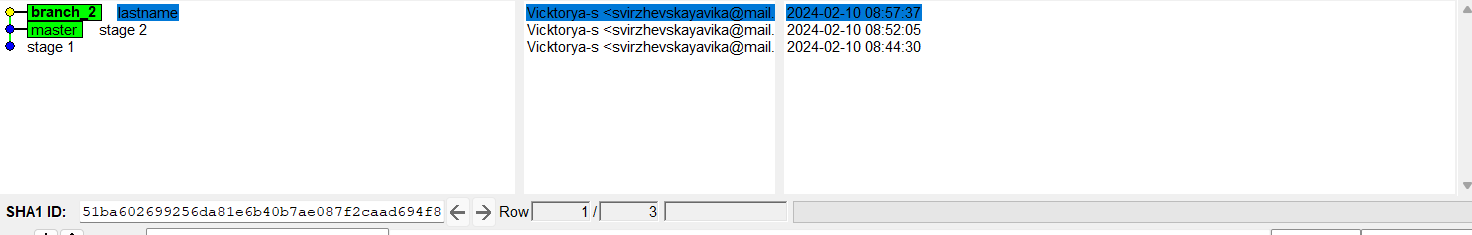
И введя текст коммита, нажмем на кнопку “Commit”.

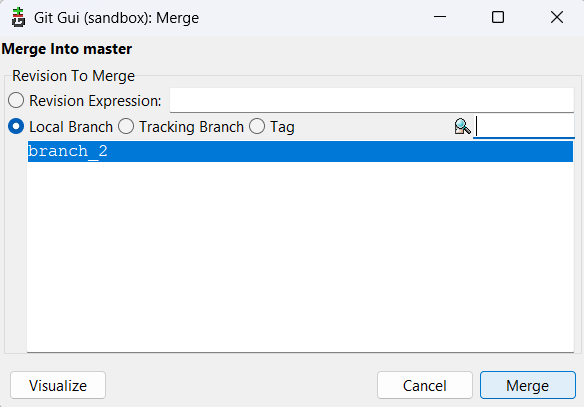
1. Ветвление

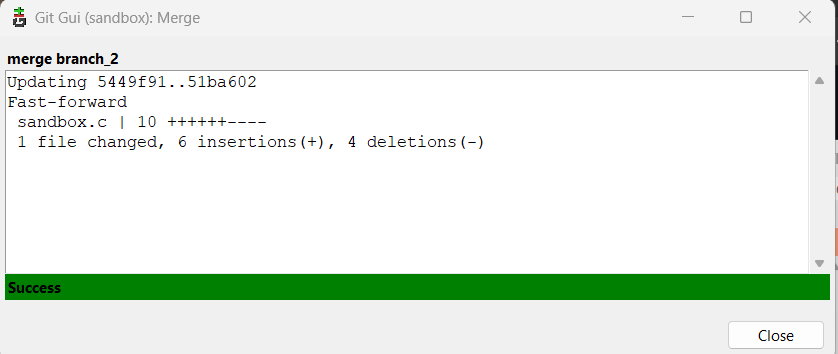




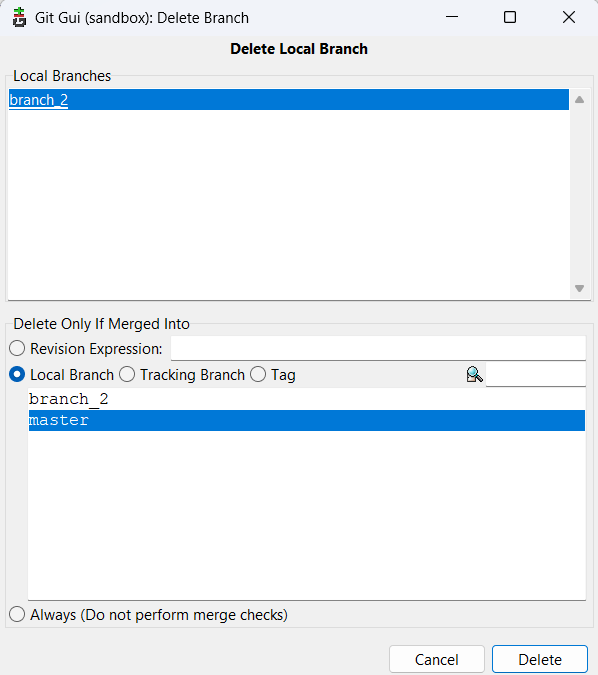




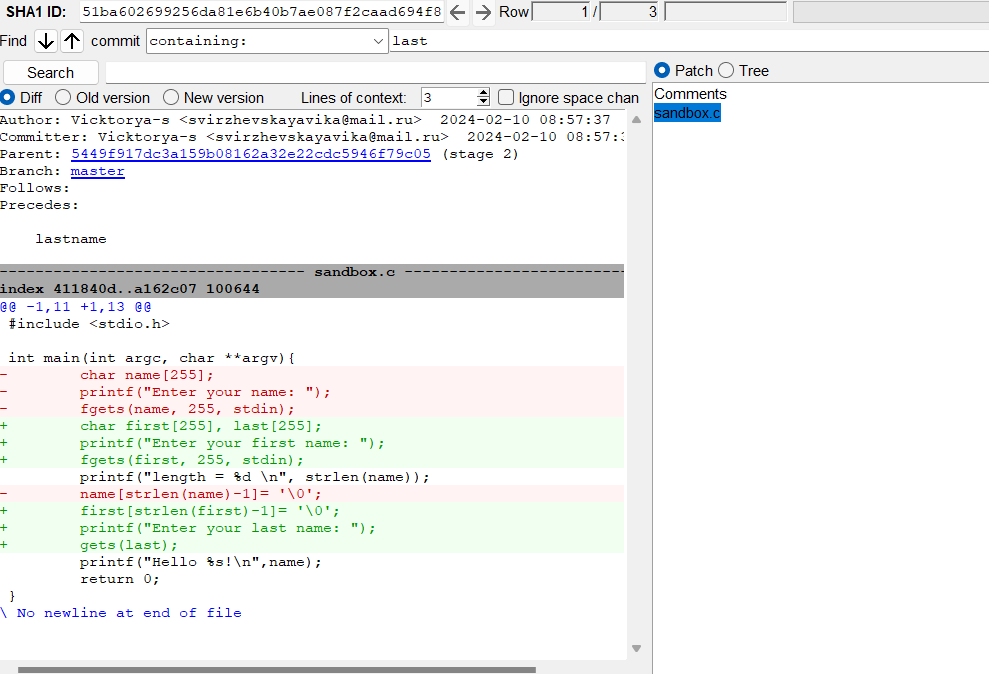












Контрольные вопросы:

1. Системы контроля версий (СКВ) - это системы, которые отслеживают изменения в файлах и помогают управлять версиями этих файлов. Они позволяют возвращаться к предыдущим версиям, сравнивать и анализировать их, а также совместно работать над проектами.

2. СКВ могут использоваться в любой сфере, где требуется отслеживание изменений. Это включает разработку программного обеспечения, дизайн, написание научных статей, создание законодательства и многое другое.

3. Наиболее популярные СКВ для разработки ПО включают Git, Mercurial, Subversion и Perforce.

4. Git является одной из самых популярных СКВ среди разработчиков из-за его гибкости, производительности и поддержки распределенной работы.

5. Существуют два основных типа СКВ: централизованные (CVCS) и распределенные (DVCS). CVCS, такие как Subversion, имеют один центральный репозиторий, в то время как DVCS, такие как Git, позволяют каждому разработчику иметь свой собственный репозиторий.

6. Git - это распределенная система контроля версий, созданная для быстрого и эффективного управления версиями в больших проектах.

7. Git был создан в 2005 году Линусом Торвальдсом для управления разработкой ядра Linux.

8. Создателем Git является Линус Торвальдс.

9. “Git” в британском сленге означает “неприятный человек”. Торвальдс выбрал это имя, потому что хотел, чтобы оно было самоироничным.

10. Основные команды Git включают git init (создание нового репозитория), git add (добавление файлов в индекс), git commit (сохранение изменений), git branch (создание новой ветки), git merge (слияние веток), git pull (получение изменений) и git push (отправка изменений).

11. Система управления версиями - это инструмент, который помогает отслеживать изменения в файлах и управлять версиями этих файлов.

12. Репозиторий можно создать с помощью команды git init.

13. Commit - это набор изменений в файлах. Git называют системой управления коммитами, потому что он отслеживает историю коммитов, а не файлов.

14. Ветку можно создать с помощью команды git branch.

15. Слияние проводится с помощью команды git merge. Конфликт происходит, когда две ветки вносят противоречивые изменения в один и тот же файл. Его можно разрешить вручную, редактируя файлы и выбирая нужные изменения.

16. Изменения можно зафиксировать с помощью команд git add и git commit.

17. Откат можно провести с помощью команд git reset (для отмены коммитов) и git revert (для создания нового коммита, который отменяет изменения). Мягкий reset (git reset --soft) отменяет коммит, но сохраняет изменения в файлах, в то время как жесткий reset (git reset --hard) полностью отменяет изменения.

18. При работе с локальным репозиторием вы обычно делаете изменения в файлах, добавляете их в индекс с помощью git add, делаете коммит с помощью git commit, и, возможно, переключаетесь между ветками с помощью git checkout.

19. При работе с удаленным репозиторием вы обычно клонируете репозиторий с помощью git clone, получаете изменения с помощью git pull и отправляете свои изменения с помощью git push.

20. При работе с удаленным репозиторием вы можете клонировать его с помощью git clone, получать изменения с помощью git pull и отправлять свои изменения с помощью git push.

21. Если вы измените название папки с рабочим проектом, Git все равно сможет отслеживать файлы внутри нее, пока вы не переместите или не удалите скрытую папку .git, которая содержит всю историю репозитория.

22. В Unix-подобных системах, таких как Linux и macOS, файлы и папки, которые начинаются с точки, считаются скрытыми. Это значит, что они не отображаются по умолчанию при просмотре содержимого директории. Папка .git - это скрытая папка, которая содержит всю информацию, необходимую Git для управления репозиторием. Это включает в себя историю коммитов, ссылки на ветки и другие метаданные.