**ОТЧЕТ**

**о выполнении лабораторной работы №4 (часть 2)**

**Дисциплина «МДК 03.01 Моделирование и анализ программного обеспечения»**

**Тема: «Создание и изучение возможностей репозитория проекта»**

**Специальность: 09.02.07 Информационные системы и программирование**

**Квалификация: Специалист по информационным системам**

Выполнил: Арутюнян Э.А.

Студент группы: ИС50-11-21

Проверил:

Преподаватель: Кретова Е.М.

Дата: «\_\_\_» \_\_\_\_\_\_\_\_\_ 2022 г.

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Теоретические сведения**

**Контроль версий**, также известный как управление исходным кодом, — это практика отслеживания изменений программного кода и управления ими.

**Системы контроля версий** — это программные инструменты, помогающие командам разработчиков управлять изменениями в исходном коде с течением времени.

**Git** — система управления версиями с распределенной архитектурой. В отличие от некогда популярных систем вроде CVS и Subversion (SVN), где полная история версий проекта доступна лишь в одном месте, в Git каждая рабочая копия кода сама по себе является репозиторием. Это позволяет всем разработчикам хранить историю изменений в полном объеме.

Преимущества:

* Git показывает очень высокую производительность в сравнении со множеством альтернатив. Это возможно благодаря оптимизации процедур фиксации коммитов, создания веток, слияния и сравнения предыдущих версий.
* При разработке в Git прежде всего обеспечивается целостность исходного кода под управлением системы. Содержимое файлов, а также объекты репозитория, фиксирующие взаимосвязи между файлами, каталогами, версиями, тегами и коммитами, защищены при помощи криптографически стойкого алгоритма хеширования SHA1.
* Гибкость — одна из основных характеристик Git. Она проявляется в поддержке различных нелинейных циклов разработки, эффективности использования с малыми и крупными проектами, а также совместимости со многими системами и протоколами.

Команда **git config** позволяет настроить инсталляцию Git (или отдельный репозиторий) из командной строки. С помощью этой команды можно установить любые настройки: от информации о пользователе до его предпочтений и характеристик репозитория.

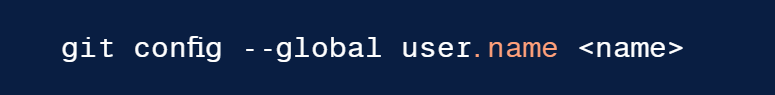


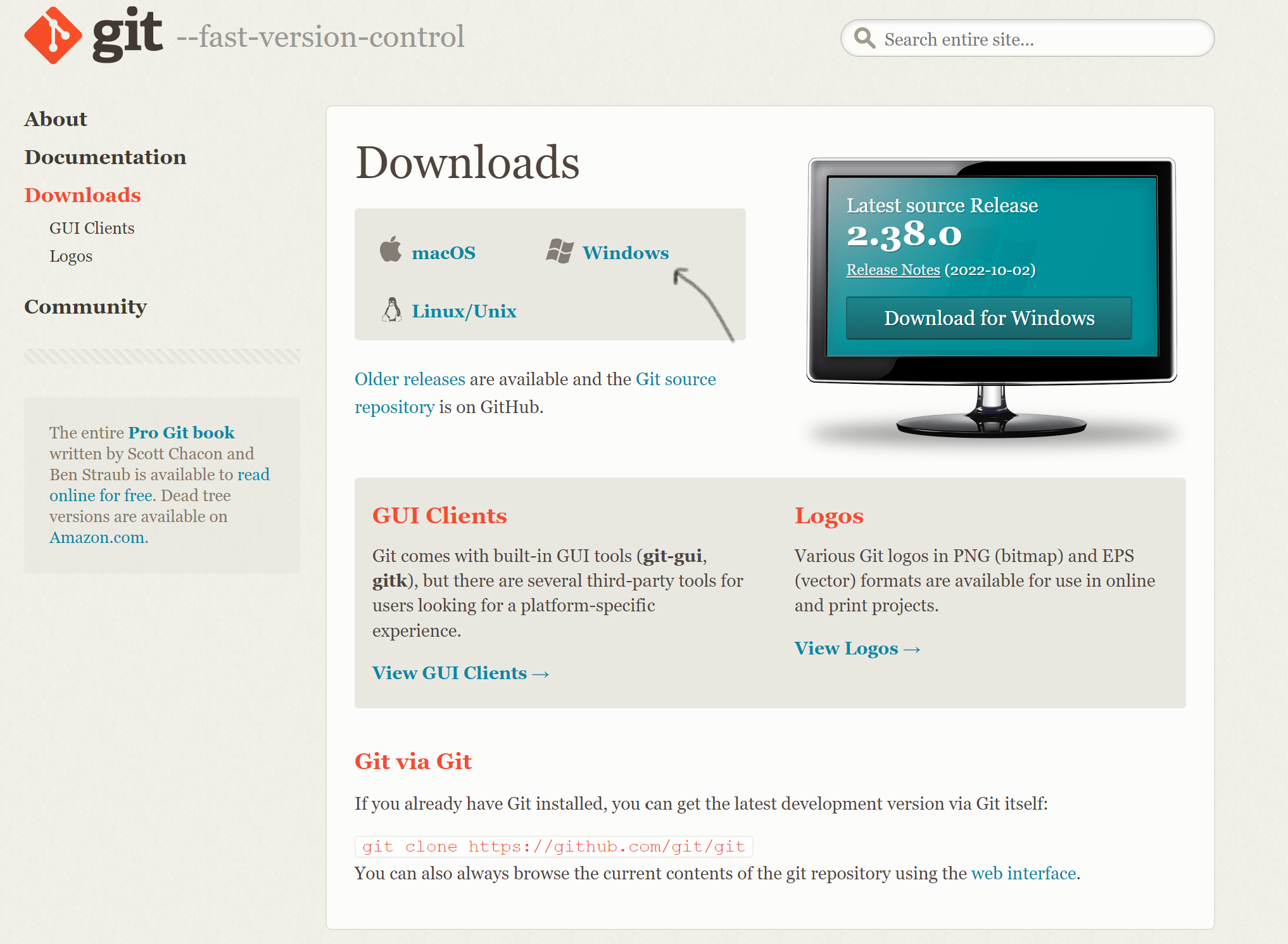
Рисунок 1 – Эта команда задает имя автора, которое будет использоваться для всех коммитов, выполненных текущим пользователем

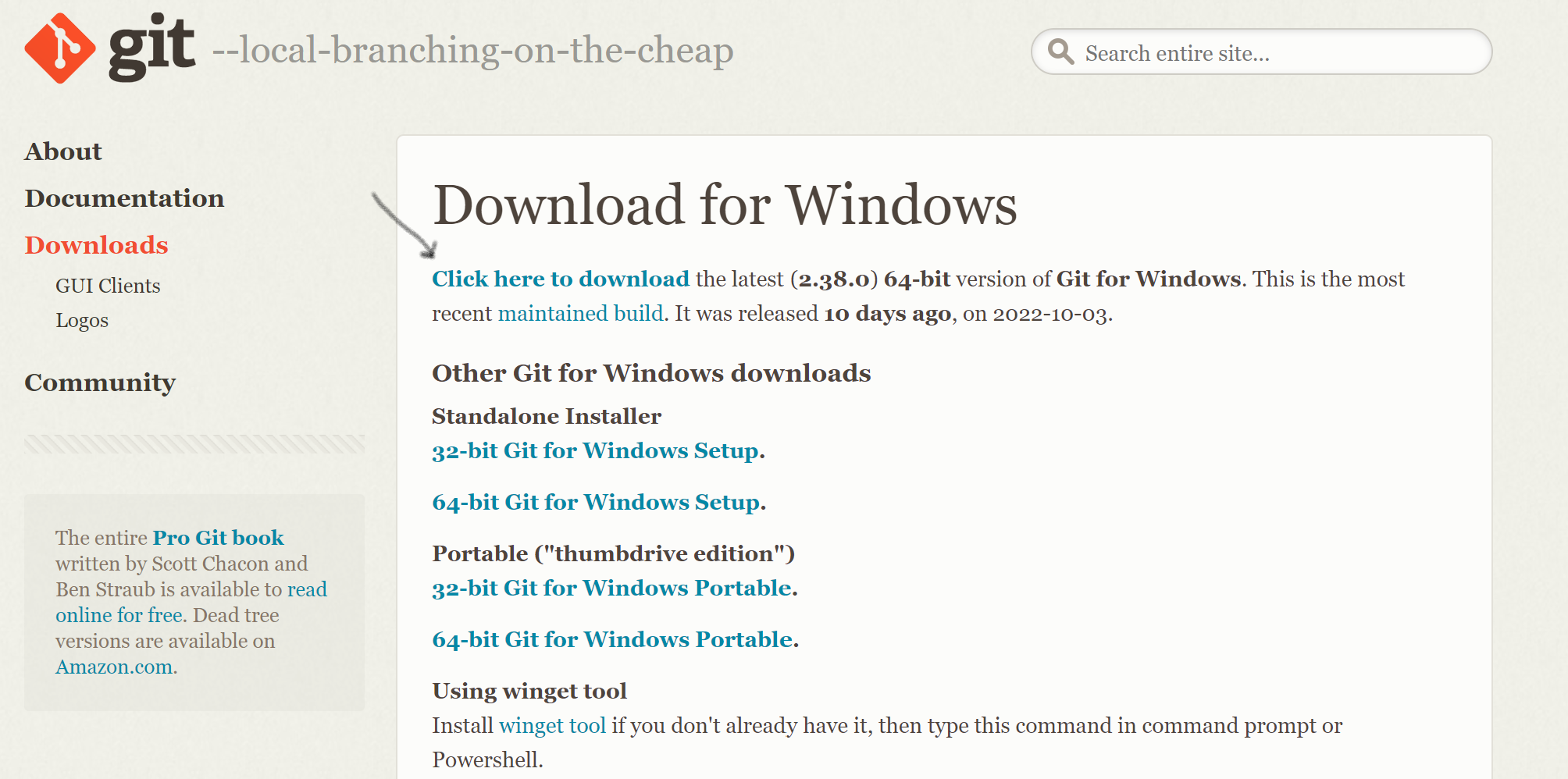


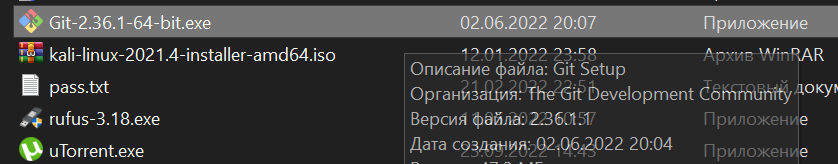
Рисунок 2 – Эта команда задает адрес электронной почты автора, который будет использоваться для всех коммитов, выполненных текущим пользователем

**Порядок выполнения работы**

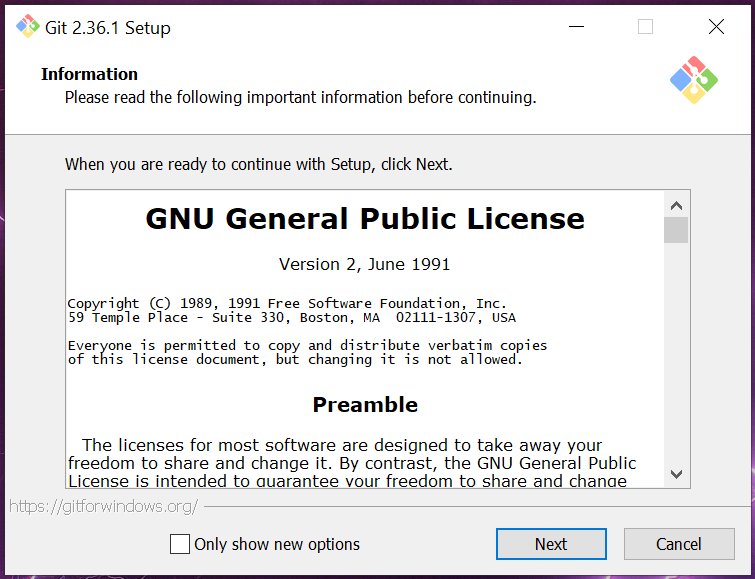
1. Переходим на сайт <https://git-scm.com/downloads>
2. Скачиваем установочник git

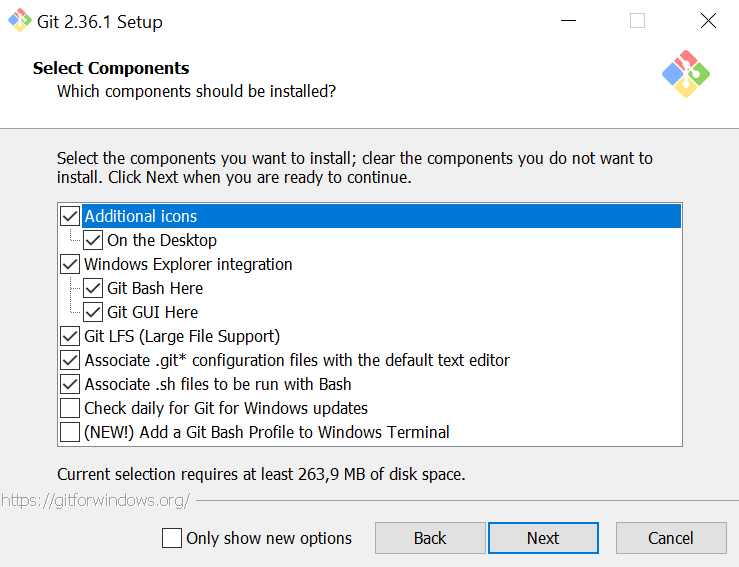


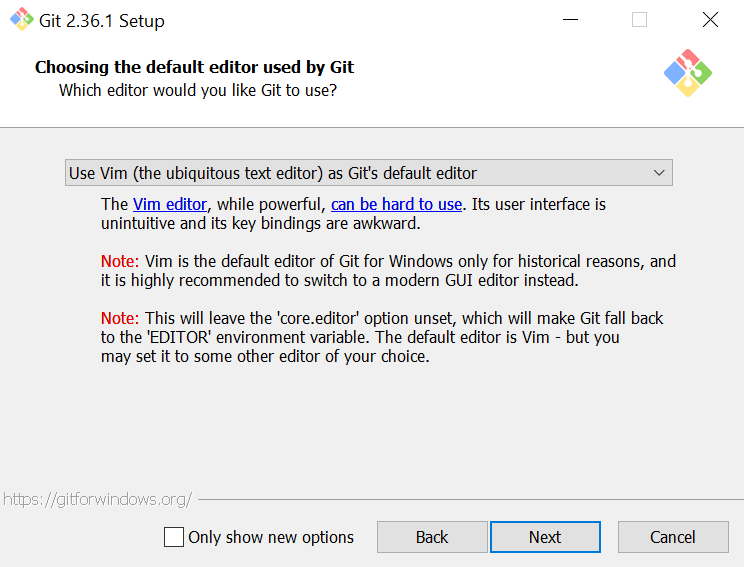


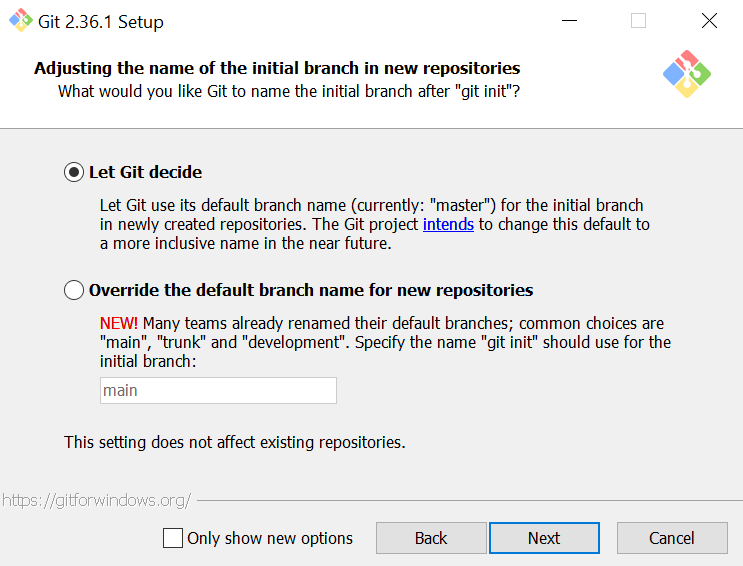


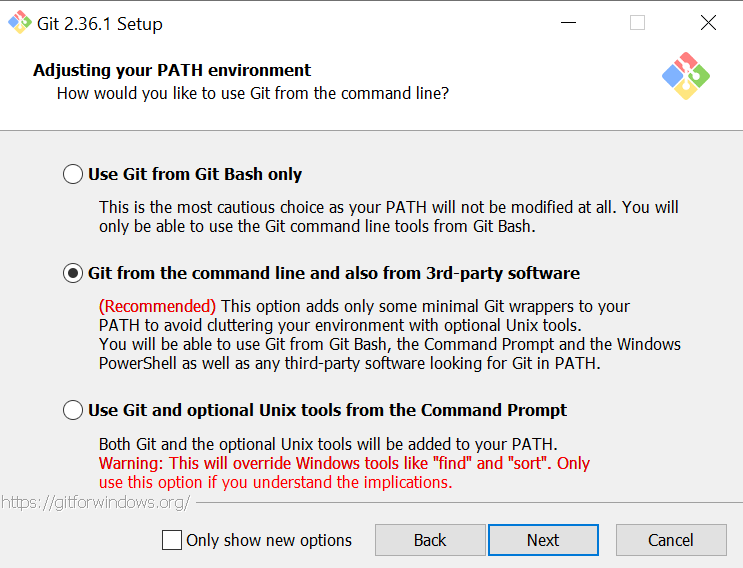
1. После успешного запуска программы установки вы увидите экран мастера установки Git. Для завершения установки следуйте инструкциям, нажимая кнопки Next (Далее) и Finish (Готово).

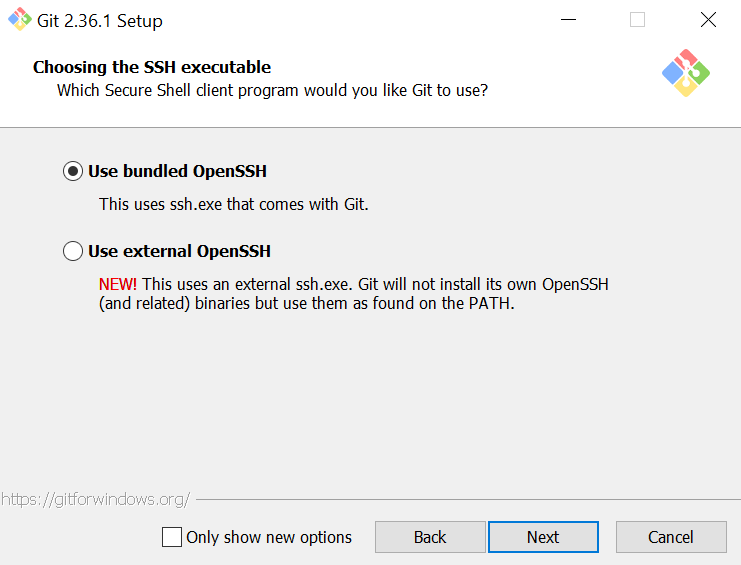


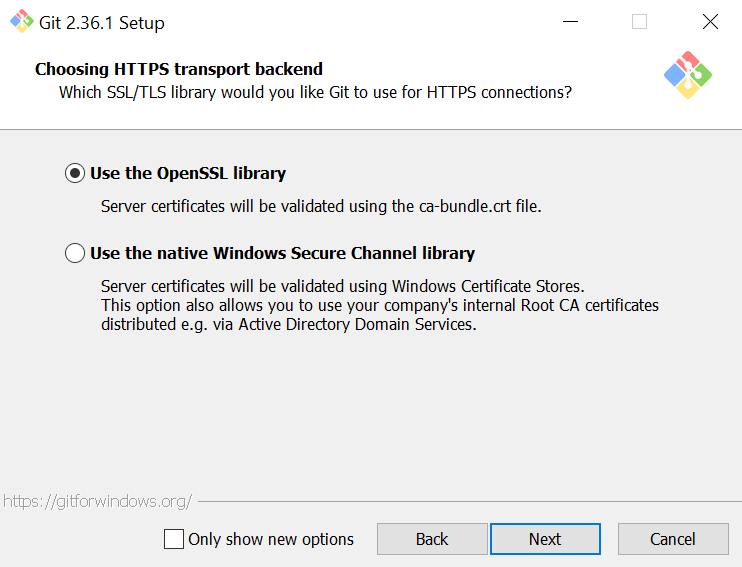


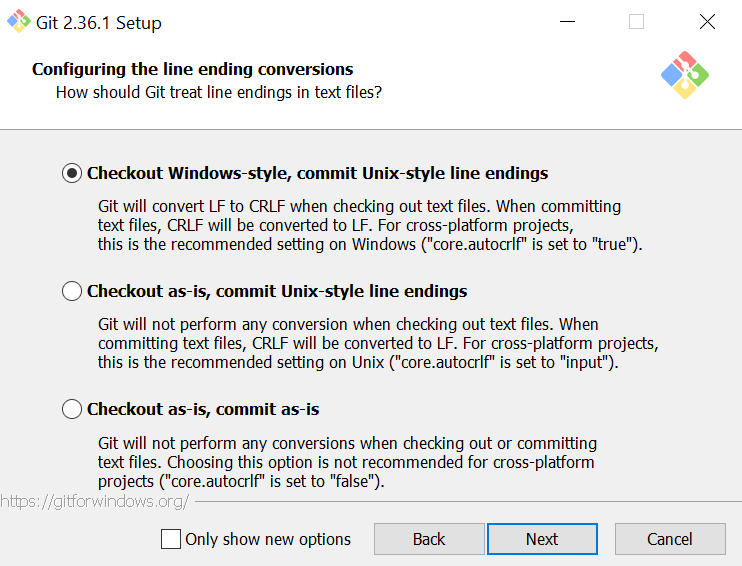


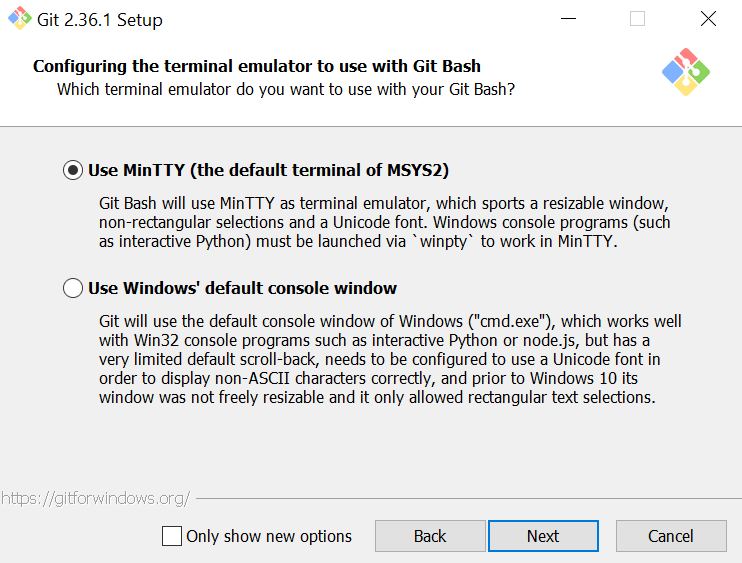


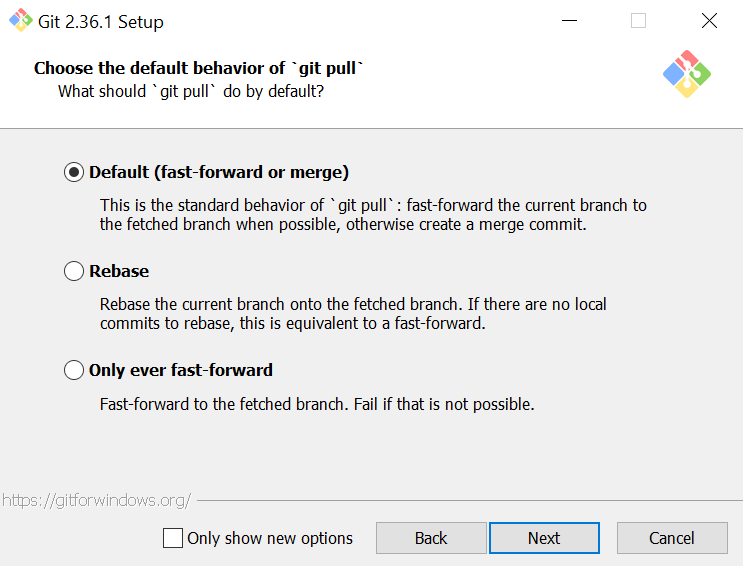


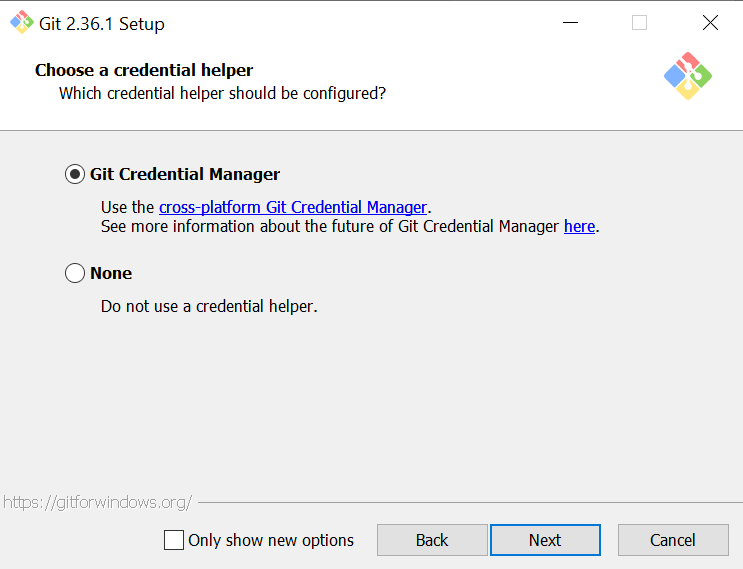


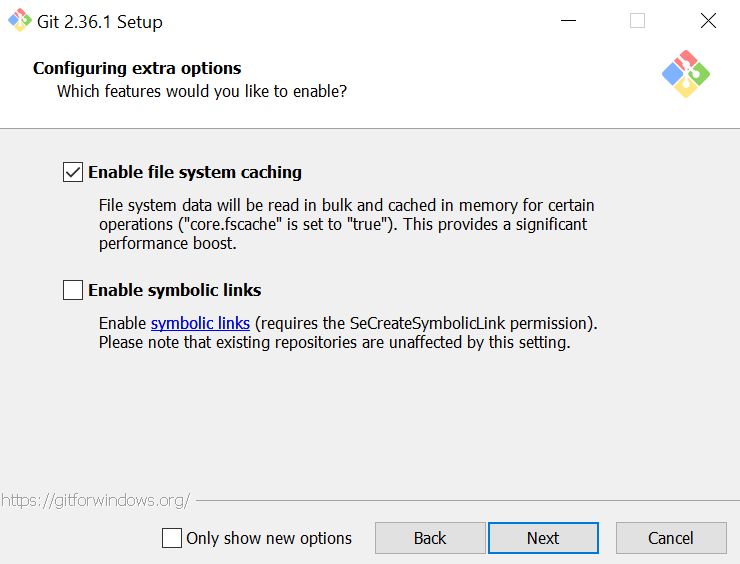


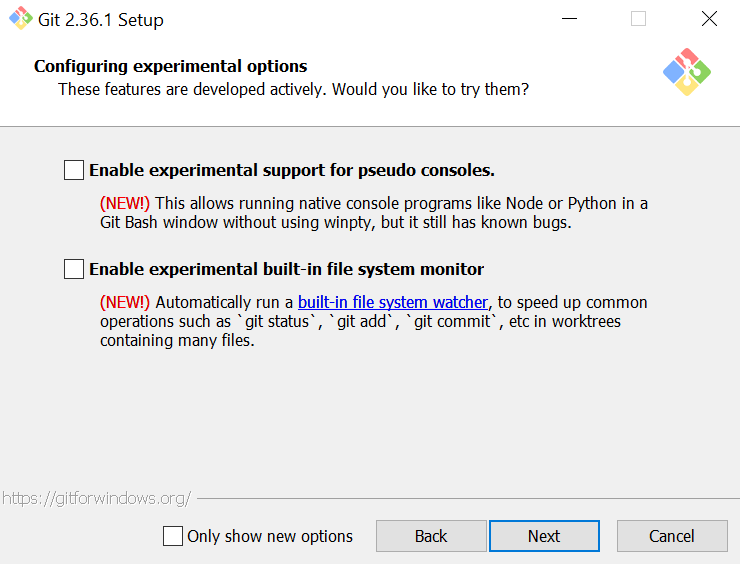


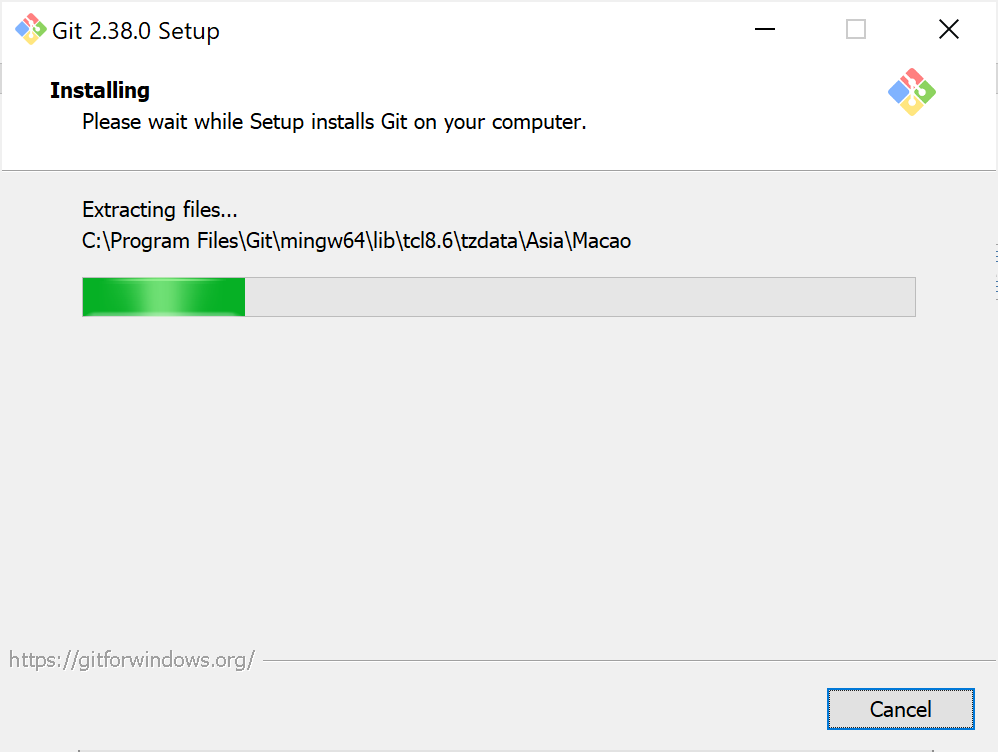


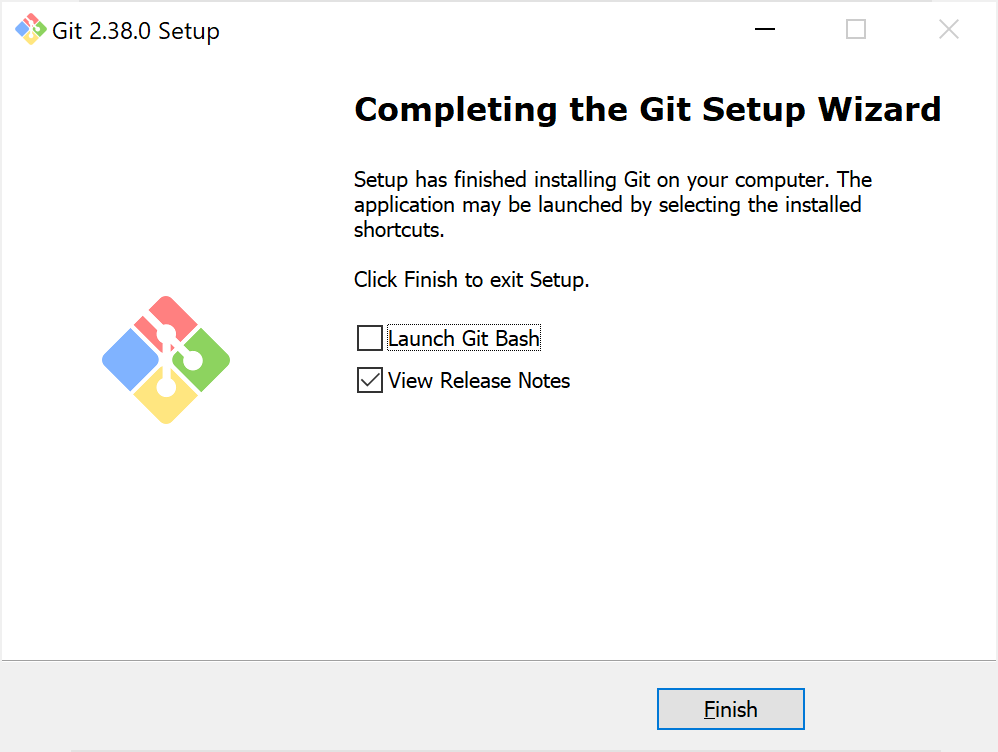




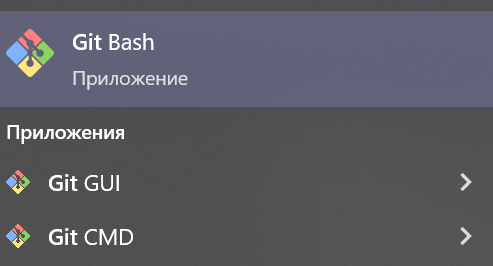


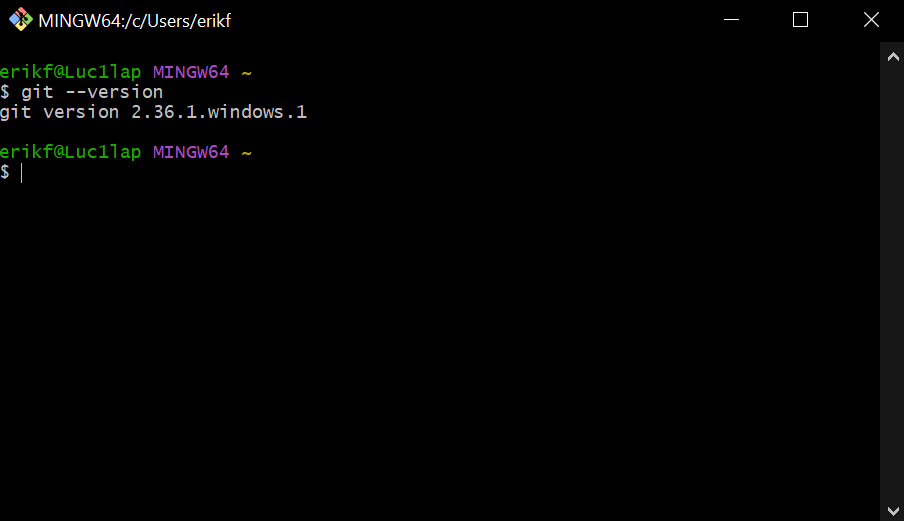




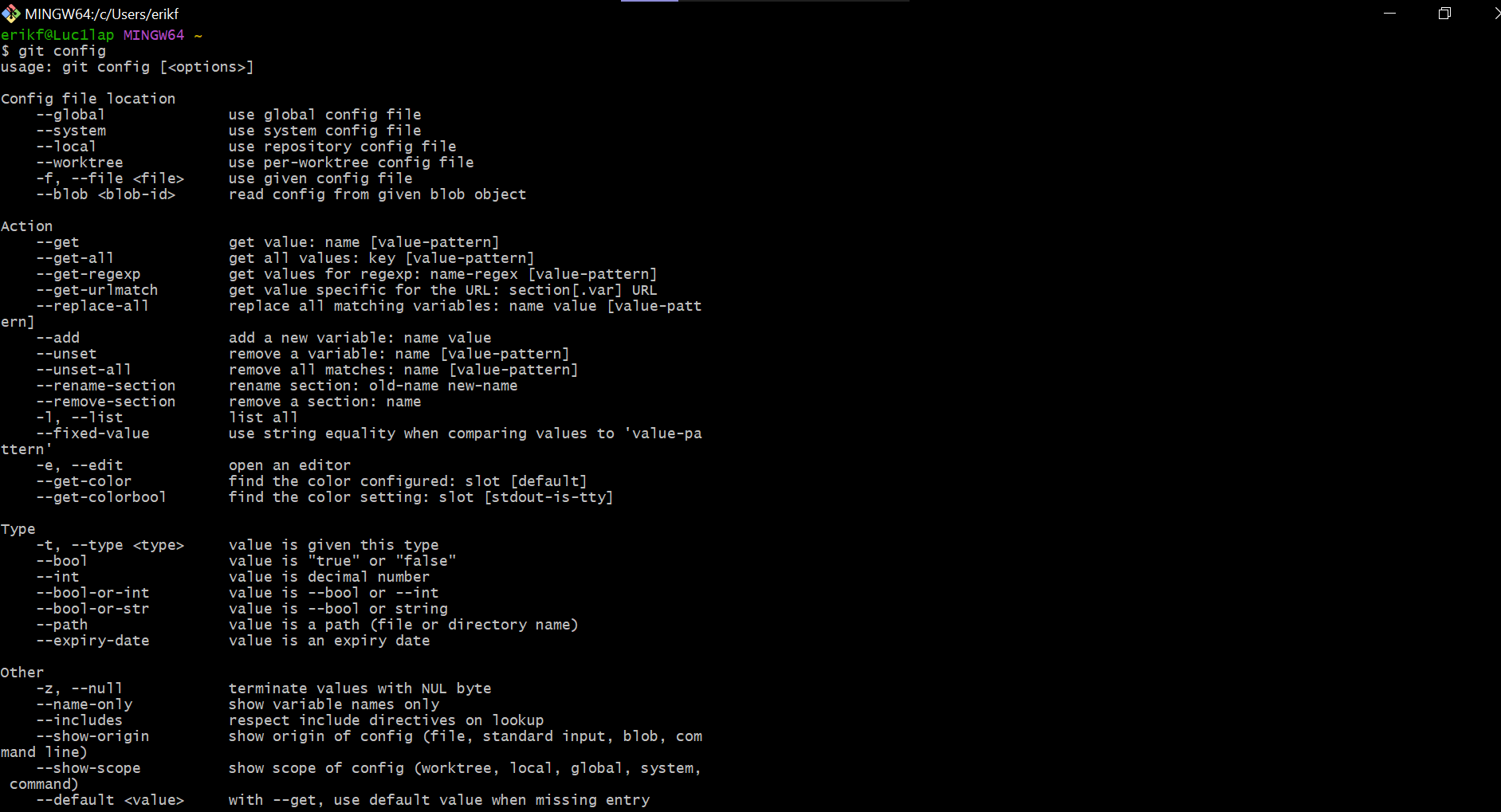


1. Запускаем Git Bash и производим проверку версии при помощи команды git –version

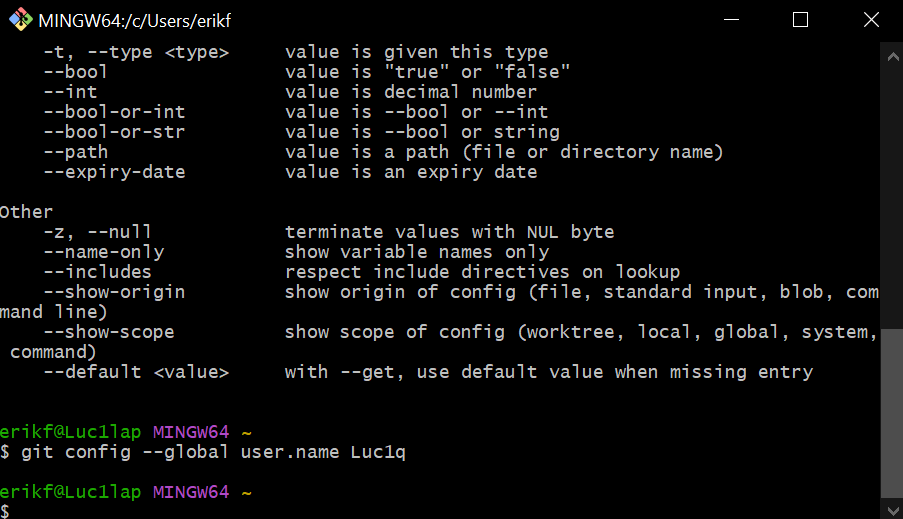




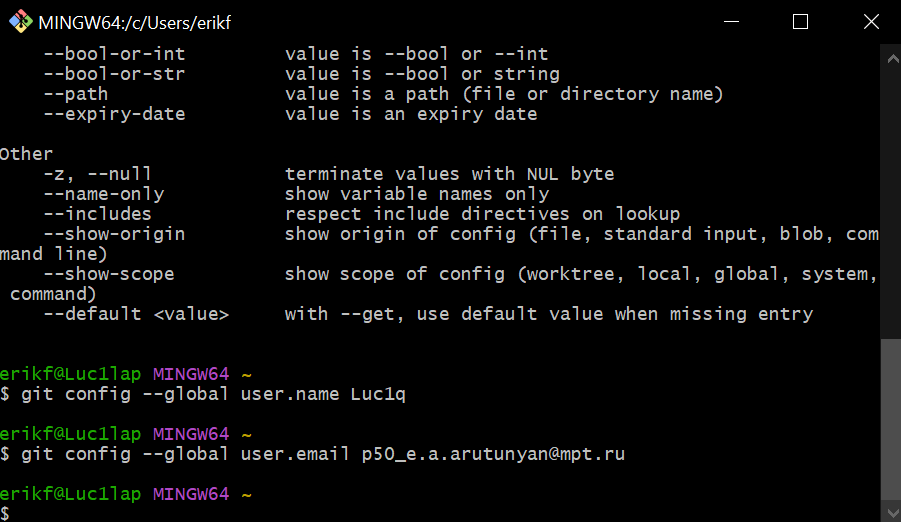
1. Производим первоначальную настройку среды для работы системы контроля версий в консольном режиме с использованием команды git config



1. Добавляем в Git контактные данные (это необходимо для идентификации изменений. Эти данные должны совпадать с аккаунтом, зарегистрированным на одном из сервисов (GitHub)
   * 1. Ввести имя пользователя при помощи настройки git config --global user.name;



* + 1. Указать адрес электронной почти при помощи настройки git config --global user.email;



* + 1. Получаем все заданные параметры при помощи команды git config --list.

