Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Северо - Восточный федеральный университет имени М.К.Аммосова»

Колледж инфраструктурных технологий

Кафедра эксплуатации и обслуживания информационных систем

Лабораторная работа № 2

Тема лабораторной работы: «Настройка работы системы контроля версий (типов импортируемых файлов, путей, фильтров и др. параметров импорта в репозиторий)», 2 часа

Выполнил студент гр.ИСИП-22-4

Томтосов Л.Н.

Проверил (а):Алексеева И.А.

Якутск, 2024 г.

**Цель работы:** изучить лсновы коллективной разработки ПО с использованием системы контроля версий, научиться настраивать СКС и формировать следующие компетенции: ОК01-ОК03, ПК2.2.

**Ход работы**

**Первоначальная настройка Git**

Теперь, когда Git установлен в вашей системе, самое время настроить среду для работы с Git’ом под себя. Это нужно сделать только один раз — при обновлении версии. Git’а настройки сохранятся. Но, при необходимости, вы можете поменять их в любой момент, выполнив те же команды снова.

В состав Git’а входит утилита git config, которая позволяет просматривать и настраивать параметры, контролирующие все аспекты работы Git’а, а также его внешний вид. Эти параметры могут быть сохранены в трёх местах:

1. Файл /etc/gitconfig содержит значения, общие для всех пользователей

системы и для всех их репозиториев. Если при запуске git config указать параметр --system, то параметры будут читаться и сохраняться именно в этот файл.

2. Файл ~/.gitconfig или ~/.config/git/config хранит настройки конкретного пользователя. Этот файл используется при указании параметра --global.

3. Файл config в каталоге Git’а (т.е. .git/config) в том репозитории, который вы используете в данный момент, хранит настройки конкретного репозитория.

Настройки на каждом следующем уровне подменяют настройки из предыдущих уровней, то есть значения в .git/config перекрывают соответствующие значения в /etc/gitconfig.

В системах семейства Windows Git ищет файл .gitconfig в каталоге $HOME (C:\Users\$USERдля большинства пользователей). Кроме того, Git ищет файл /etc/gitconfig, но уже относительно корневого каталога MSys, который находится там, куда вы решили установить Git, когда запускали инсталлятор.

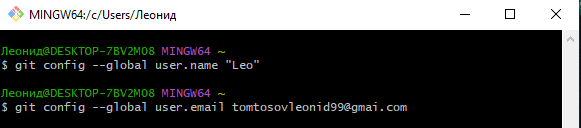
**Имя пользователя**

Первое, что вам следует сделать после установки Git’а, — указать ваше имя и адрес электронной почты. Это важно, потому что каждый коммит в Git’е содержит эту информацию, и она включена в коммиты, передаваемые вами, и не может быть далее изменена:

$ git config --global user.name "John Doe"

$ git config --global user.email johndoe@example.com

Опять же, если указана опция --global, то эти настройки достаточно сделать только один раз, поскольку в этом случае Git будет использовать эти данные для всего, что вы делаете в этой системе. Если для каких-то отдельных проектов вы хотите указать другое имя или электронную почту, можно выполнить эту же команду без параметра --global в каталоге с нужным проектом.



Многие GUI-инструменты предлагают сделать это при первом запуске.

**Выбор редактора**

Теперь, когда вы указали своё имя, самое время выбрать текстовый редактор, который будет использоваться, если будет нужно набрать сообщение в Git’е. По умолчанию Git использует стандартный редактор вашей системы, которым обычно является Vim. Если вы хотите использовать другой текстовый редактор, например, Emacs, можно проделать следующее:

$ git config --global core.editor emacs

**git Проверка настроек**

Если вы хотите проверить используемую конфигурацию, можете использовать команду git config --list, чтобы показать все настройки, которые Git найдёт:

$ git config --list

user.name=John Doe

user.email=johndoe@example.com

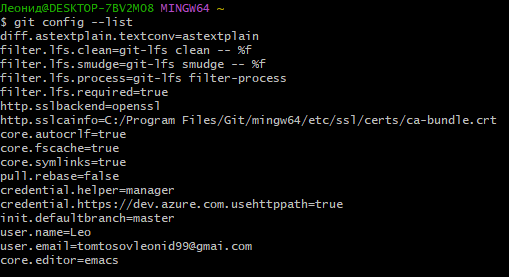
color.status=auto

color.branch=auto

color.interactive=auto

color.diff=auto

...



Некоторые ключи (названия) настроек могут появиться несколько раз, потому что Git читает один и тот же ключ из разных файлов (например, из /etc/gitconfig и ~/.gitconfig). В этом случае Git использует последнее значение для каждого ключа. Также вы можете проверить значение конкретного ключа, выполнив git config <key>:

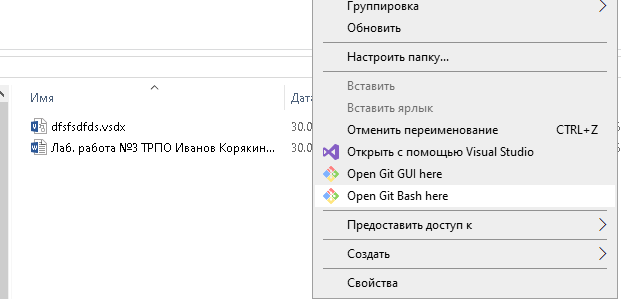
$ git config user.name

John Doe

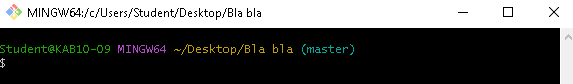
****

**Подключение к удаленному репозиторию**

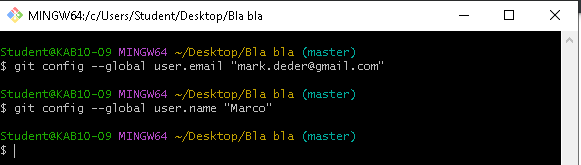
Откройте каталог с файлами, которые необходимо отслеживать в системе контроля версий и выложить на GitHub. В пустую часть каталога нажмите правой кнопкой мыши и выберете Git Bash Here.

****

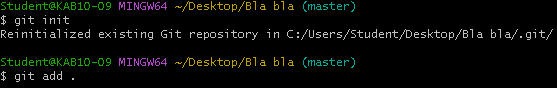
Перед вами откроется приглашение командной строки в зависимости от настроек



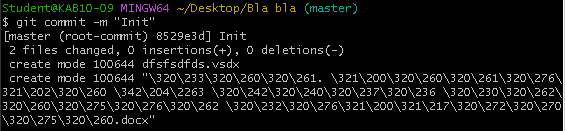
Для настройки необходимо указать ваше имя и электронную почту: git config --global user.email "you@example.com" git config --global user.name "Ваше имя"



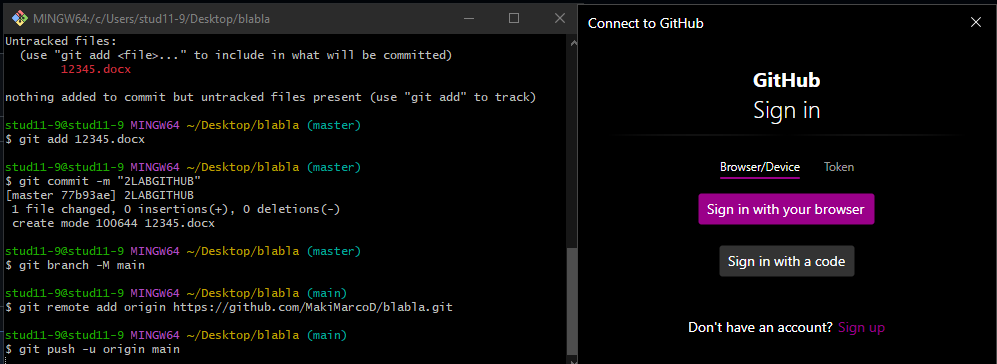
Для того чтобы начать отслеживать содержимое папки в системе, выполните команды: git init git add

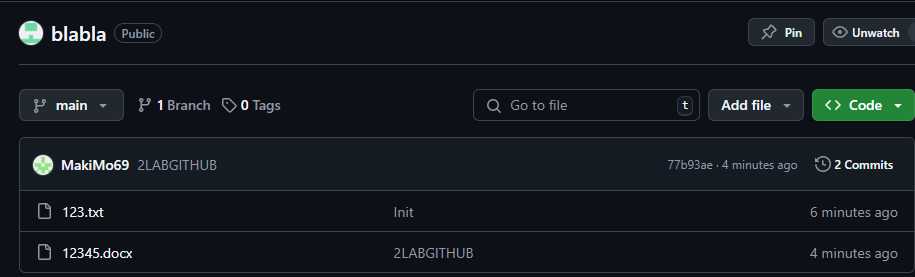


Выполните первый коммит: git commit -m "Init"



Чтобы добавить изменения, например, на github выполните действие: git remote add origin https://github.com/пользователь/репозиторий.git git push -u origin master





**Аутентификация по SSH-ключам**

Услуги 1cloud

Виртуальный сервер на базе Windows

• Лицензия включена в стоимость

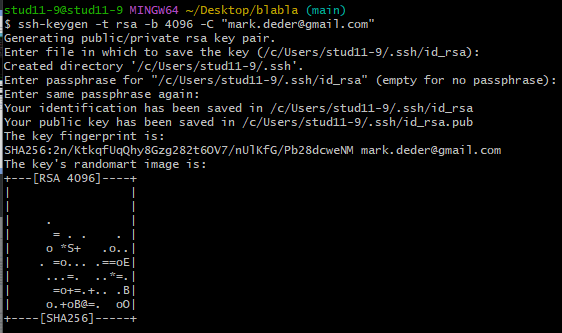
• Тестирование 3-5 дней

• Безлимитный трафик

Выполнить подключение к удаленному репозиторию можно по протоколу SSH с помощью приватного и публичного ключа. Чтобы сгенерировать пару, выполните команду в консоли GitBash, указав ваш реальный почтовый адрес:

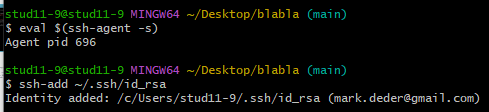
ssh-keygen -t rsa -b 4096 -C [email@example.com](mailto:email@example.com)

В процессе генерации вам будет предложено указать директорию и имя файла для ключа, примите значение по умолчанию, нажав Enter. Далее, при желании можно указать ключевую фразу в качестве дополнительной защиты, при ее отсутствии нажмите Enter. В результате будут сгенерированы ключи.



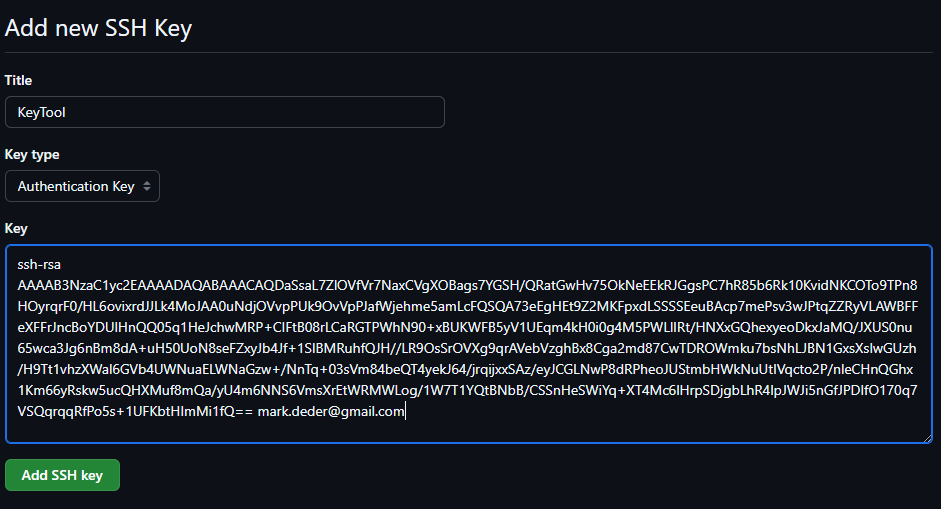
Запустите ssh-агент: eval $(ssh-agent -s)

Вы увидите идентификатор процесса. Добавьте ключи к ssh-агенту: ssh-add ~/.ssh/id\_rsa

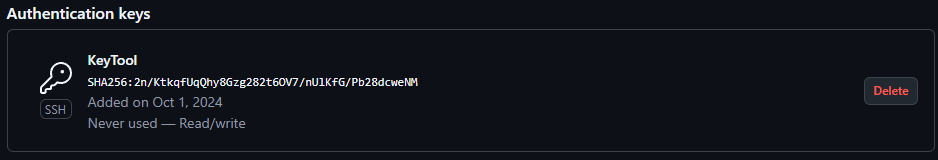


Теперь нужно добавить ключ к удаленному серверу Git. Заходим на сервер Git и в настройках добавляем SSH-ключ.

Например, на сервере GitHub это можно сделать следующим образом: откройте Settings -> SSH and GPG keys,нажмите New SSH key. В поле Title введите понятное название, в поле Key вставьте публичный ключ (содержимое файла ~/.ssh/id\_rsa.pub). Нажмите Add SSH key.



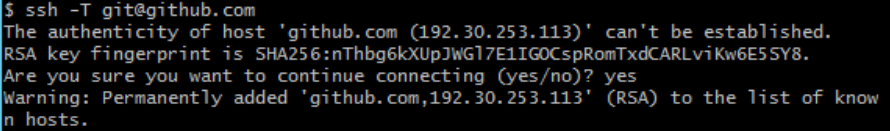
В результате у вас появится новый ключ



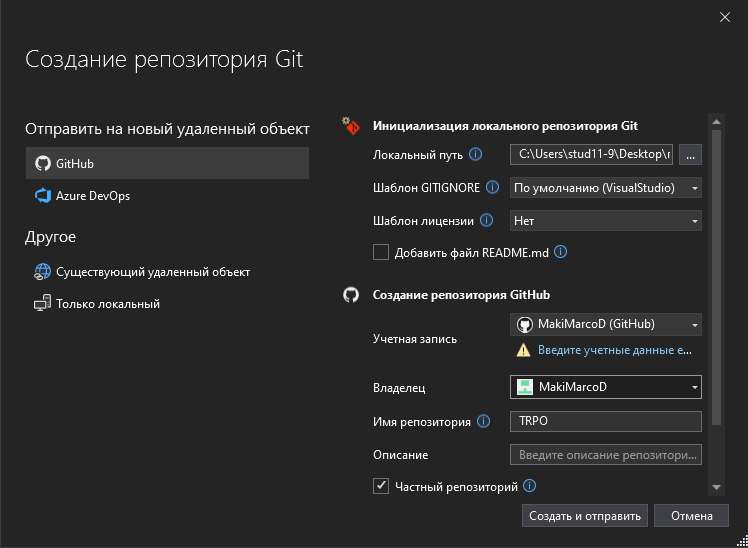
Для проверки настроек выполните команду: ssh -T git@

Например: ssh -T [git@github.com](mailto:git@github.com)

Если все настроено верно, то вы увидите следующее сообщение.



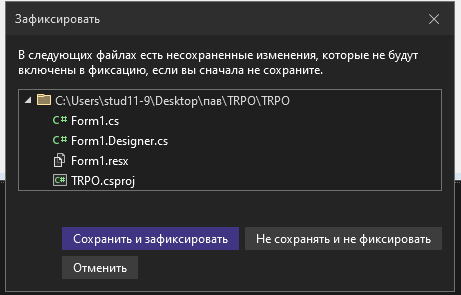
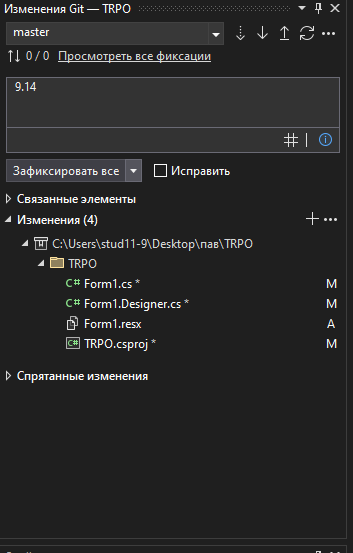
Чтобы открыть репозиторий GIT создадим репозиторий git выбираем GitHub, добовляем уч. запись, владельца и имя

****

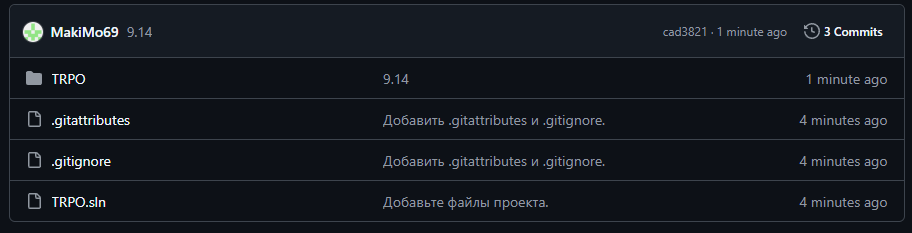
Открываем GitHub и проверяем.

****

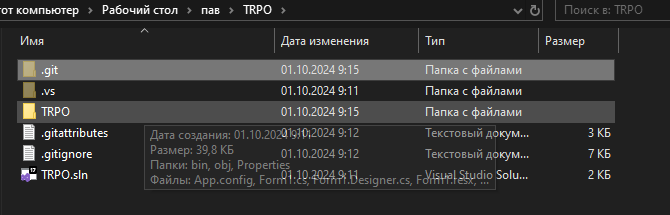
Чтобы сохранить изменение открываем коммит и сохраняем.

****

Открываем GitHub и проверяем

****

Чтобы выйти удалим скрытый git

****

**Вывод**

На данной лабораторной работе мы изучили основы коллективной разработки ПО с использованием системы контроля версий, научились настраивать СКС.