*Алиева Хадижат ПИ20-1*

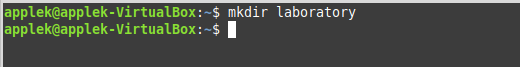
**2. Работа с Git в терминале**

**Цель работы**

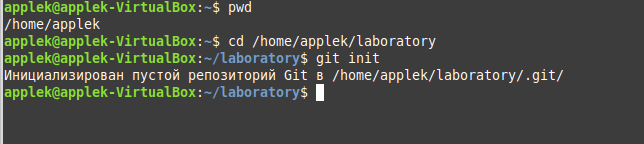
На основе уже полученных знаний о принципах работы СКВ получить более глубокое представление о работе Git при помощи команд терминала.

**Задания для выполнения**

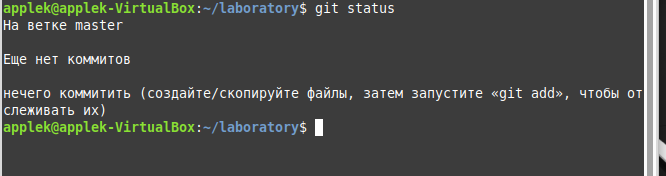
1. Выберите тематику программы, которую собираетесь написать. Создайте для нее рабочую директорию



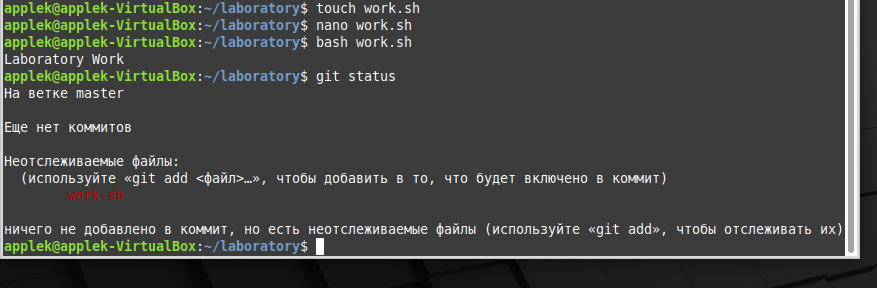
1. Инициализируйте в рабочей директории репозиторий при помощи команды git init.



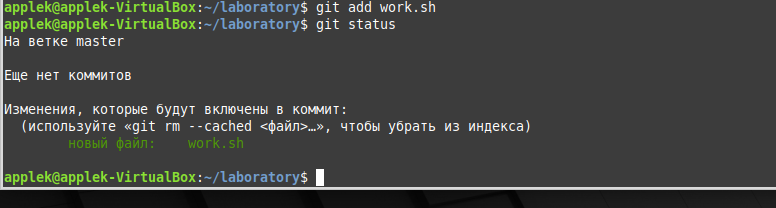
1. Выполните в репозитории команду git status. Проинтерпретируйте полученное сообщение.



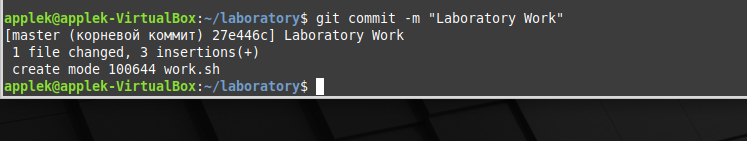
1. Создайте файл для исходного текста программы. Выполните команду git status.



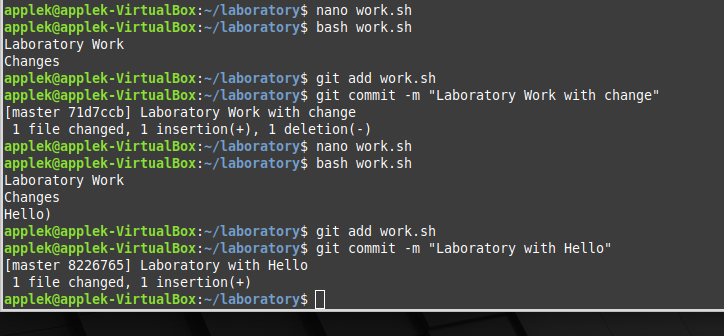
1. Добавьте созданный файл под версионный контроль при помощи команды git add. Еще раз выполните git status.

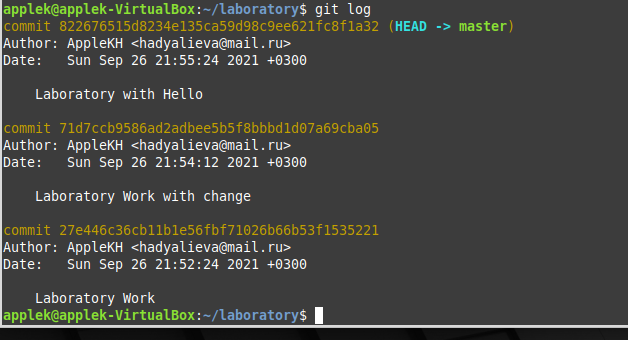


1. Сделайте начальный коммит при помощи команды git commit с опцией -m.

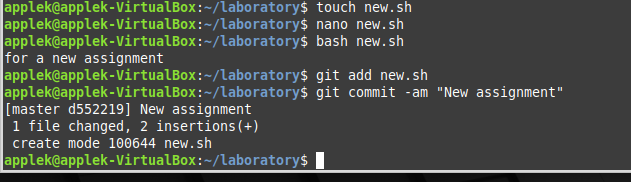


1. Сделайте еще несколько коммитов. Выполните команду git log для просмотра истории коммитов.

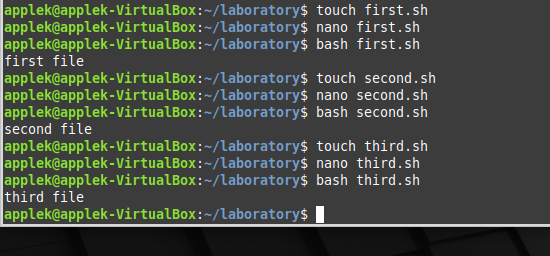




1. Сделайте так, чтобы при коммите измененные файлы автоматически добавлялись в коммит.



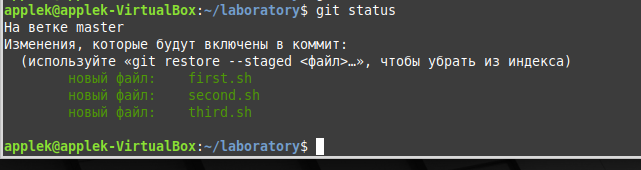
1. Добавьте еще несколько файлов с исходным текстом программы.



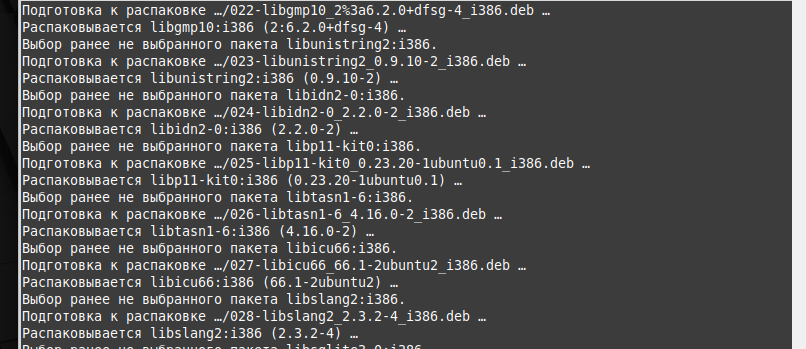
1. Добавьте все новые файлы под версионный контроль одной командой.

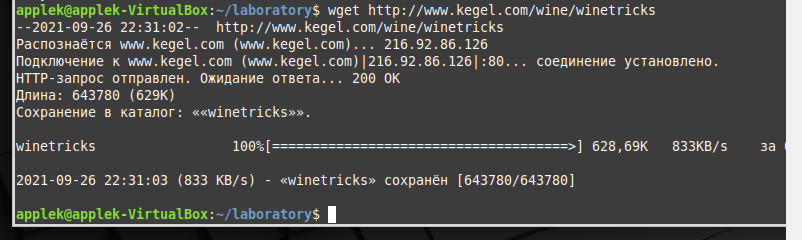


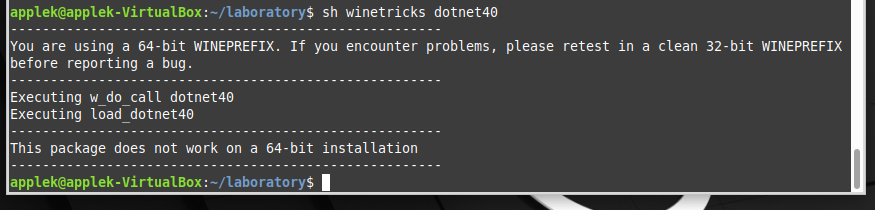
1. На всех стадиях работы пользуйтесь командой git status.



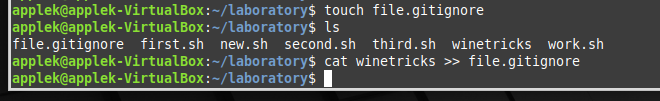
1. Инициализируйте в рабочей директории виртуальное окружение (Если вы пишите не на Python, то можете инициализировать какой-либо программный фреймворк, либо начать работать в IDE, которая создает скрытую папку с настройками в рабочем каталоге).

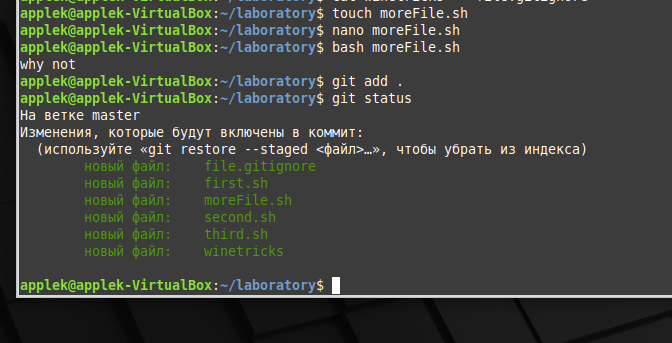


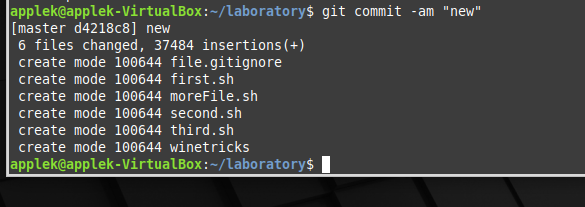




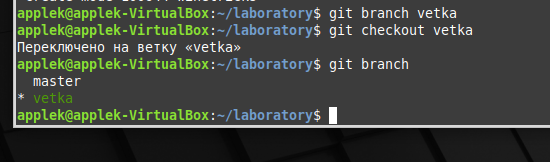
1. Добавьте созданную служебную папку в файл .gitignore. Проверьте, что они не добавляются в репозитории при добавлении новых файлов с исходным кодом.



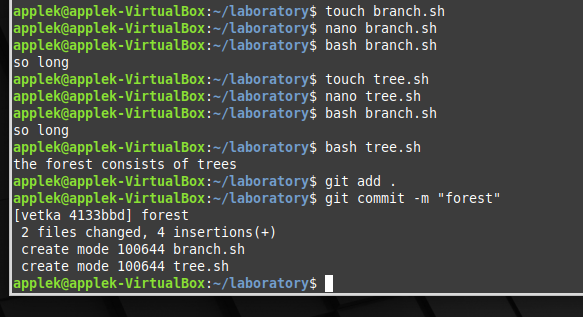


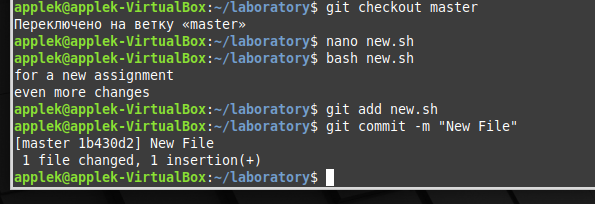


1. Создайте новую тематическую ветку git branch. Перейдите в нее с помощью git checkout. Выведите на экран список всех веток.

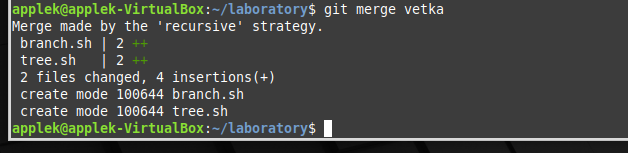


1. Сделайте несколько коммитов в основную и тематическую ветки.

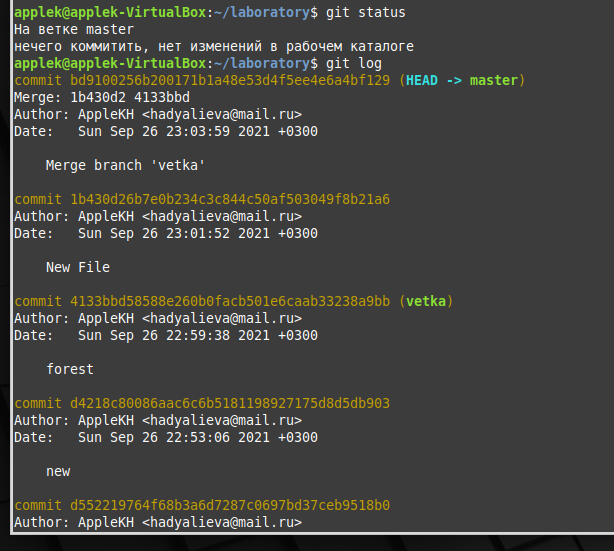




1. Слейте изменения в основную ветку с помощью git merge. Если произошел конфликт слияния, разрешите его и завершите слияние с помощью git commit.



1. При получении в процессе разработки программы в стабильно работающем состоянии, слейте это состояние в основную ветку и добавьте к коммиту слияния пометку с номером релиза.



**Контрольные вопросы**

1. Что такое удаленный репозиторий?

Это преобразование проекта, которое хранится в интернете или ином месте в сети.

1. Где нужно вводить команды git?

Команды git нужно вводить в терминале в репозитории.

1. Для чего нужны ветки в системах контроля версий?

Оно необходимо, когда пользователь производит два разных изменения, основываясь на одном и том же состоянии.

1. Как возникают конфликты слияния?

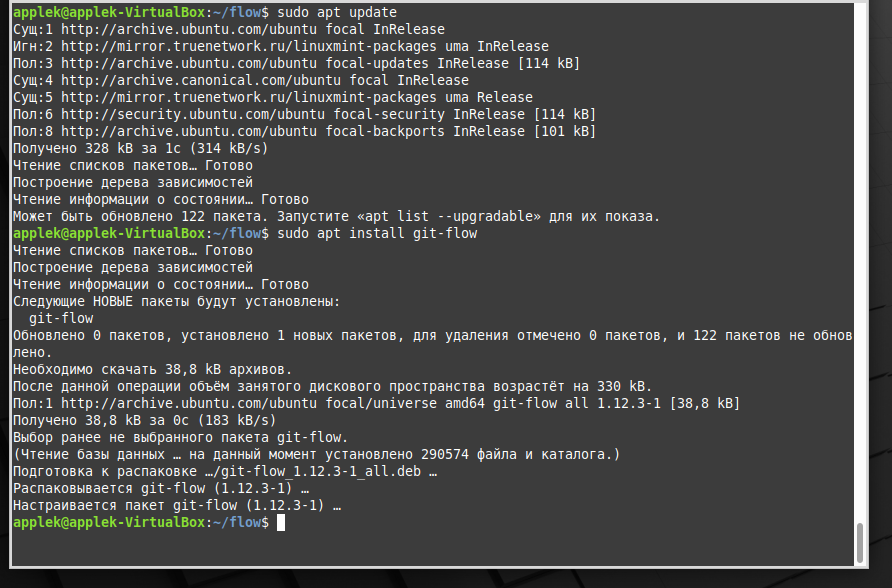
Конфликты слияния возникают, когда пользователь изменил одну и ту же часть файла по-разному в двух ветках, которые собирается слить.

1. Как разрешать конфликты слияния?

Необходимо выбрать одну из этих частей или как-то объединить содержимое по своему усмотрению. После нужно выполнить добавление этих файлов к индексу и коммит для завершения слияния веток.

**Дополнительные задания**

1. Ознакомьтесь с методологией разработки GitFlow.
2. Установите на свой компьютер инструментальное средство для работы с GitFlow.



1. Выполните основные задания лабораторной работы с использованием команд git-flow.

