## Лабораторная работа № 5.

## Управление версиями

### 1. Цель работы

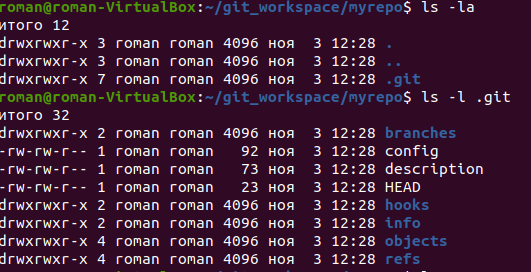
Изучить идеологию и применение средств контроля версий

Последовательность выполнения работы

**1 Настройте систему контроля версий git, как это описано выше c использованием сервера репозиториев** [**http://bitbucket.org/**](http://bitbucket.org/)**.**



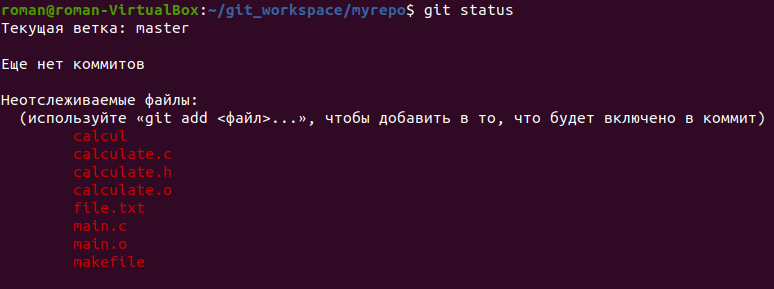


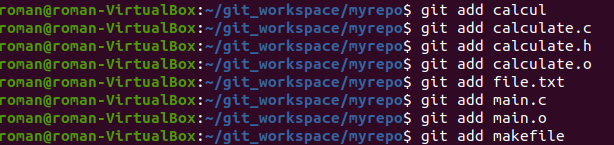
****

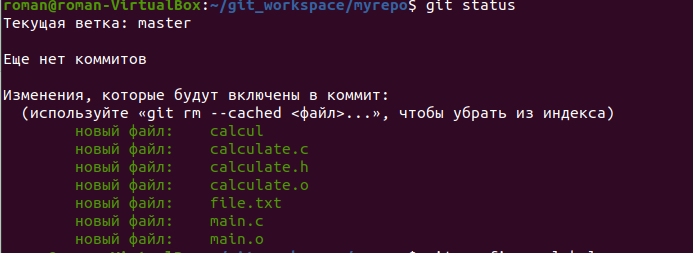
**2 Для создания репозитория используйте файлы из лабораторной работы №** [**11**](#_bookmark86)**: calculate.h, calculate.c, main.c.**

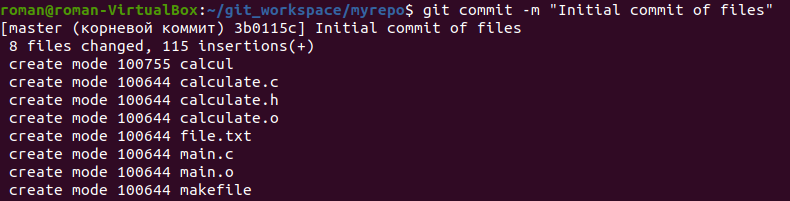
**Внесите какие-либо изменения в файлы проекта.**

****



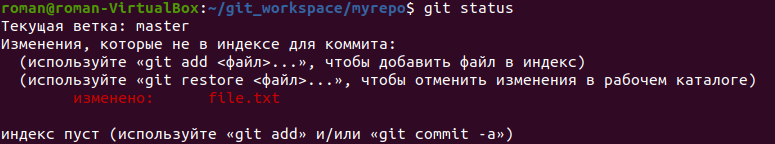
****

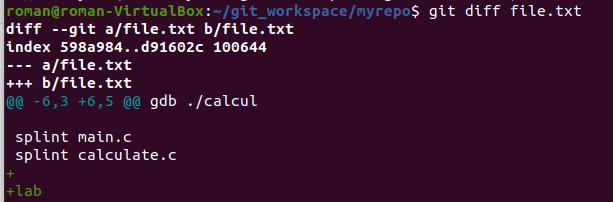
****

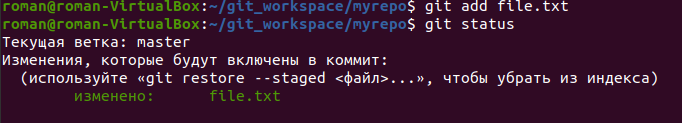
****

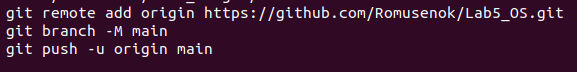
**3 Добавьте изменения в локальный и удалённый репозитории, используя соответствующие команды git.**

**New changes in the calculate.c file**

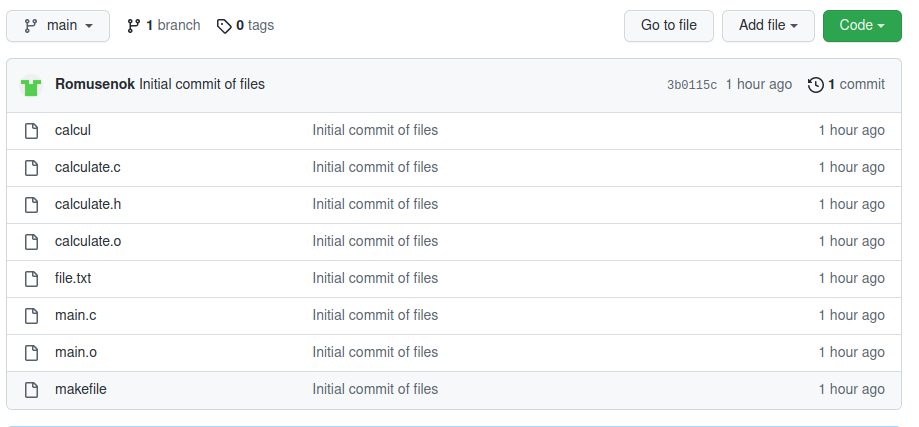
****

****



****

**4 Подтвердите появление изменений на сервере репозиториев, сделав соответствующий скриншот.**



**3 Добавьте изменения в локальный и удалённый репозитории, используя соответствующие команды git.**

**4 Подтвердите появление изменений на сервере репозиториев, сделав соответствующий скриншот.**

1. **Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются?**

Version Control System, VCS или Revision Control System) — программное обеспечение для облегчения работы с изменяющейся информацией

1. **Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.**
2. **Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.**

В случае с централизованной VCS репозиторий хранится на одном сервере, и все разработчики работают с ним.

Как следует из названия, у децентрализованных систем нет единого центрального владельца. Вместо этого они используют нескольких центральных владельцев, каждый из которых обычно хранит копию ресурсов, к которым пользователи могут получить доступ.

### [Git-ssb](https://habr.com/ru/post/537678/) — децинтрализованный

### Git bucket — централизованный

1. **Опишите действия с VCS при единоличной работе с хранилищем.**
2. **Опишите порядок работы с общим хранилищем VCS.**
3. **Каковы основные задачи, решаемые инструментальным средством git?**
4. **Назовите и дайте краткую характеристику командам git.**
5. **Приведите примеры использования при работе с локальным и удалённым репозиториями.**
6. **Что такое и зачем могут быть нужны ветви (branches)**

Для того, чтобы разделять код. Например одна ветка у нас может быть основная для разработки. Если мы делаем новый функционал, то мы создаем новую ветку под него, а после окончания работы сливаем то, что мы сделали в основную ветку.

1. **Как и зачем можно игнорировать некоторые файлы при commit?**

Чтобы игнорировать файл, для которого ранее был сделан коммит, необходимо удалить этот файл из репозитория, а затем добавить для него правило в .gitignore. Используйте команду git rm с параметром --cached, чтобы удалить этот файл из репозитория, но оставить его в рабочем каталоге как игнорируемый файл. Игнорируемые файлы — это, как правило, артефакты сборки и файлы, генерируемые машиной из исходных файлов в вашем репозитории, либо файлы, которые по какой-либо иной причине не должны попадать в коммиты.