**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ОБРАЗОВАНИЯ**

**«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ» ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

**Отчет о лабораторной работе №1.2 по дисциплине основы программной инженерии**

Выполнила:

Гробова Софья Кирилловна,

2 курс, группа ПИЖ-б-о-20-1,

Проверил:

Доцент кафедры прикладной математики и

компьютерной безопасности, Воронкин Р.А.

Отчет защищен с оценкой Дата защиты

Ставрополь, 2021 г.

ВЫПОЛНЕНИЕ:

Знакомство с возможностями просмотра журнала хранилища и отката изменений

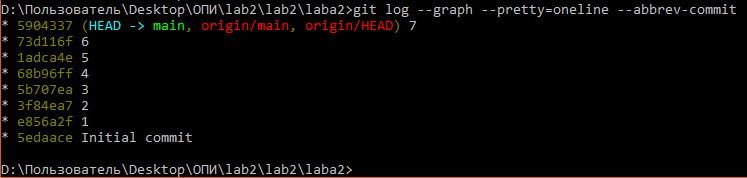


Рисунок 1.1 – Просмотр хранилища с помощью команды «git log».

Просмотр отдельных коммитов

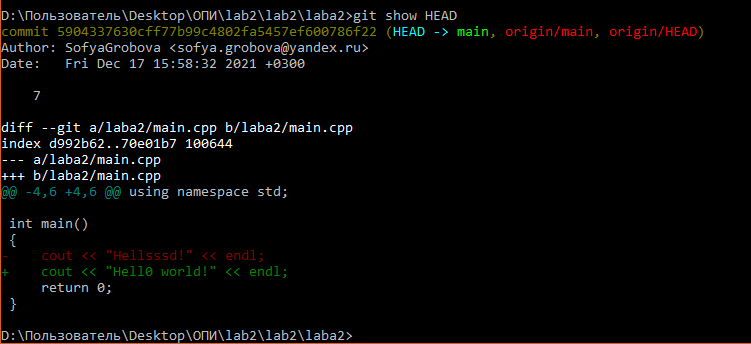


Рисунок 1.2 – Вывод последнего коммита с помощью команды «git show HEAD»

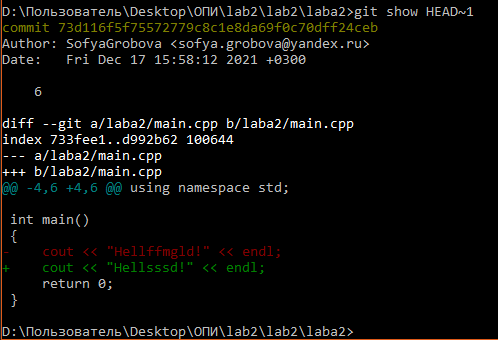


Рисунок 1.3 – Вывод предпоследнего коммита с помощью добавления флага «~1»



Рисунок 1.4 – Вывод коммита по значению хэша

Отмена изменений в локальном файле с помощью удаленного репозитория

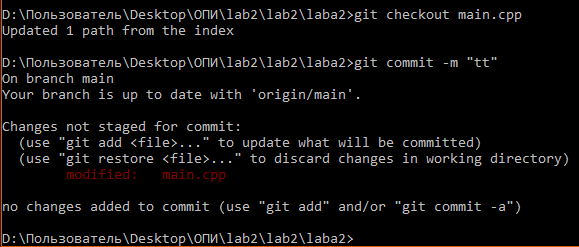


Рисунок 1.5 – Отмена изменений в файле с помощью команды «git checkout»

Отмена последнего коммита

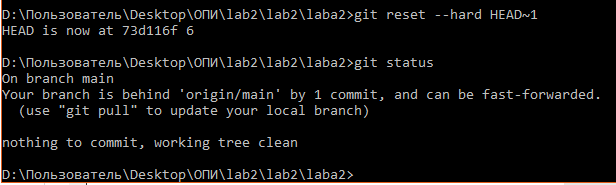


Рисунок 1.6 – Отмена последнего коммита с помощью «git reset —hard»

Благодаря команде «git checkout» можно восстановить данные, которые были удалены из файла. Для этого Git загружает данные в этот файл из

последнего отправленного в удаленный репозиторий коммита. Команда «git reset —hard» позволяет даже отменять коммиты сделанных изменений, но является довольно опасным инструментом, так как удаленный коммит не будет сохранен в истории.

Просмотр тэгов

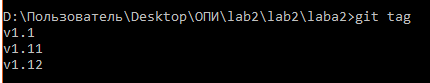


Рисунок 1.7 – Просмотр всех тэгов в репозитории

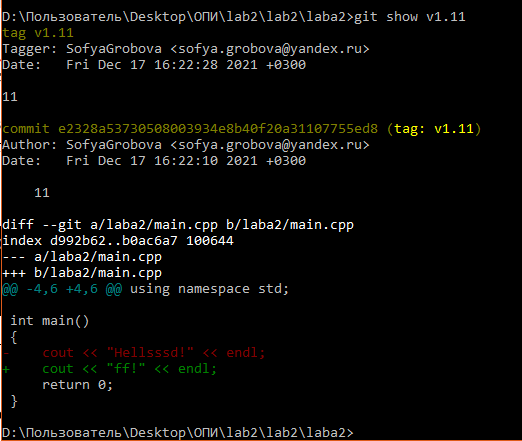


Рисунок 1.8 – Просмотр конкретного тэга с помощью команды «git show»

Ответы на вопросы

1. Как выполнить историю коммитов в Git? Какие существуют дополнительные опции для просмотра истории коммитов?

Одним из основных и наиболее мощных инструментов для этого является команда git log. По умолчанию (без аргументов) git log

перечисляет коммиты, сделанные в репозитории в обратном к

хронологическому порядке — последние коммиты находятся вверху. Данная команда перечисляет коммиты с их SHA-1 контрольными суммами, именем и электронной почтой автора, датой создания и сообщением коммита.

Команда git log имеет очень большое количество опций для поиска коммитов по разным критериям. Одним из самых полезных аргументов является -p или —patch, который показывает разницу (выводит патч), внесенную в каждый коммит. Если вы хотите увидеть сокращенную

статистику для каждого коммита, вы можете использовать опцию –stat. Опция –pretty меняет формат вывода. Существует несколько встроенных вариантов отображения. Опция oneline выводит каждый коммит в одну

строку, что может быть очень удобным если вы просматриваете большое количество коммитов. К тому же, опции short , full и fuller делают вывод

приблизительно в том же формате, но с меньшим или большим количеством информации соответственно. Наиболее интересной опцией является format , которая позволяет указать формат для вывода информации. Особенно это

может быть полезным, когда вы хотите сгенерировать вывод для

автоматического анализа — так как вы указываете формат явно, он не будет изменен даже после обновления Git.

1. Как ограничить вывод при просмотре истории коммитов?

Вы можете ограничить количество записей в выводе команды; используйте параметр -2 для вывода только двух записей, опции для

ограничения вывода по времени: —since и —until, следующим действительно полезным фильтром является опция -S, которая принимает аргумент в виде строки и показывает только те коммиты, в которых изменение в коде

повлекло за собой добавление или удаление этой строки. Последней

полезной опцией, которую принимает команда git log как фильтр, является путь. Если вы укажете каталог или имя файла, вы ограничите вывод только теми коммитами, в которых были изменения этих файлов.

1. Как внести изменения в уже сделанный коммит?

git commit –amend.

1. Как отменить индексацию файла в Git?

git reset HEAD <file>.

1. Как отменить изменения в файле?

git checkout — <file>.