

Лабораторная работа 1

Работа с git

Кудряшов А. Н.

Российский университет дружбы народов, Москва, Россия

Информация

- Кудряшов Артём Николаевич
- студент
- Российский университет дружбы народов
- 1132226433@pfur.ru
- <https://ankudryashovNKA.github.io/ru/>

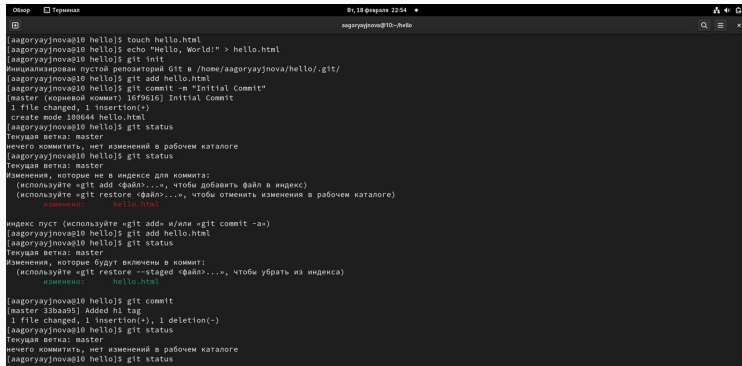
.....
.....



Приобрести практические навыки работы с системой управления версиями Git.

1. Создать репозиторий
2. Изменить его содержимое, изучить работу с коммитами, версиями и тегами
3. Поработать с объектами Git
4. Создать дополнительную ветку, провести слияние, разрешение конфликтов
5. Создать клон репозитория и слить его с оригинальным
6. Залить и извлечь изменения

Выполнение лабораторной работы



```
Вит, 18 февраля 22:54
aagoryayjnova@10 ~ - hello

[aagoryayjnova@10 hello]$ touch hello.html
[aagoryayjnova@10 hello]$ echo "Hello, World!" > hello.html
[aagoryayjnova@10 hello]$ git init
Инициализирован пустой репозиторий Git в /home/aagoryayjnova/hello/.git/
[aagoryayjnova@10 hello]$ git add hello.html
[aagoryayjnova@10 hello]$ git commit -m "Initial Commit"
[master (корневой коммит) 16f9616] Initial Commit
1 file changed, 1 insertion(+)
create mode 100644 hello.html
[aagoryayjnova@10 hello]$ git status
Текущая ветка: master
ничего коммитить, нет изменений в рабочем каталоге
[aagoryayjnova@10 hello]$ git status
Текущая ветка: master
Изменения, которые не в индексе для коммита:
  (используйте «git add <файл>...», чтобы добавить файл в индекс)
  (используйте «git restore <файл>...», чтобы отменить изменения в рабочем каталоге)
изменено:   hello.html

индекс пуст (используйте «git add» и/или «git commit -а»)
[aagoryayjnova@10 hello]$ git add hello.html
[aagoryayjnova@10 hello]$ git status
Текущая ветка: master
Изменения, которые будут включены в коммит:
  (используйте «git restore --staged <файл>...», чтобы убрать из индекса)
изменено:   hello.html

[aagoryayjnova@10 hello]$ git commit
[master 33baa95] Added h1 tag
1 file changed, 1 insertion(+), 1 deletion(-)
[aagoryayjnova@10 hello]$ git status
Текущая ветка: master
ничего коммитить, нет изменений в рабочем каталоге
[aagoryayjnova@10 hello]$ git status
```

Рис. 1: Создание репозитория

Изменим содержимое файла hello.html на:

```
<h1>Hello, World!</h1>
```


Внесение изменений

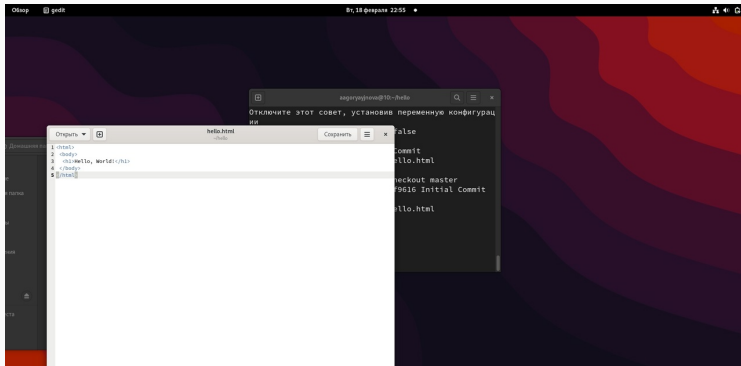
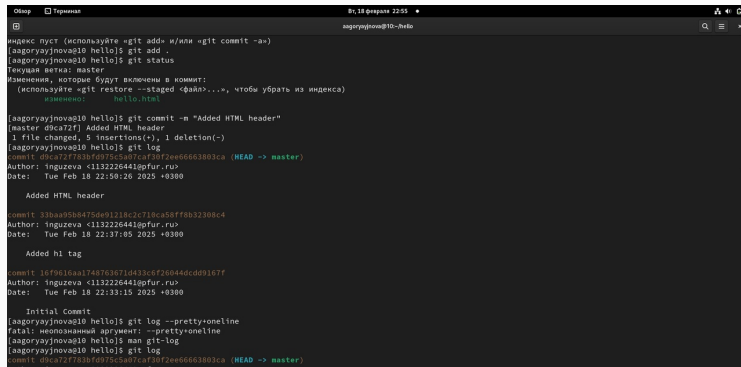


Рис. 2: Внесение нескольких изменений в файл

Получим список произведённых изменений в стандартном виде, затем в однострочном, а также с указанием времени и количества.

Получение старых версий



```
индекс пуст (используйте «git add» и/или «git commit -a»)
[aagoryayjnovae@10 hello]$ git add .
[aagoryayjnovae@10 hello]$ git status
Текущая ветка: master
Изменения, которые будут включены в коммит:
  (используйте «git restore --staged <файл>...», чтобы убрать из индекса)
    изменено:   hello.html

[aagoryayjnovae@10 hello]$ git commit -m "Added HTML header"
[master d9ca72f] Added HTML header
1 file changed, 5 insertions(+), 1 deletion(-)
[aagoryayjnovae@10 hello]$ git log
commit d9ca72f783b9d975c5a07caf30f2ee66663803ca (HEAD -> master)
Author: inguzeva <1132226441@pfur.ru>
Date:   Tue Feb 18 22:50:26 2025 +0300

    Added HTML header

commit 33baa95b8475de91218c2c710ca58ff8b32308c4
Author: inguzeva <1132226441@pfur.ru>
Date:   Tue Feb 18 22:37:05 2025 +0300

    Added hi tag

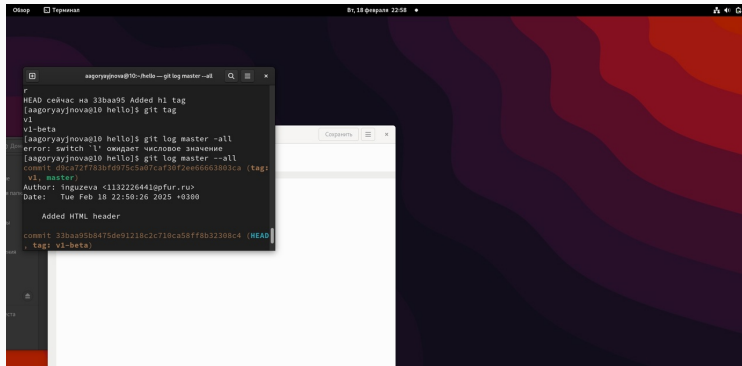
commit 16f9616aa1748763671d433c6f26044dcdd9167f
Author: inguzeva <1132226441@pfur.ru>
Date:   Tue Feb 18 22:33:15 2025 +0300

    Initial Commit
[aagoryayjnovae@10 hello]$ git log --pretty=oneline
fatal: неопознанный аргумент: --pretty=oneline
[aagoryayjnovae@10 hello]$ man git-log
[aagoryayjnovae@10 hello]$ git log
commit d9ca72f783b9d975c5a07caf30f2ee66663803ca (HEAD -> master)
Author: inguzeva <1132226441@pfur.ru>
Date:   Tue Feb 18 22:50:26 2025 +0300

    Added HTML header
```

Рис. 3: Просмотр разных версий репозитория

Создание тегов версий



```
aaagoryaynova@10:~/hello -- git log master --all
HEAD сейчас на 33baa95 Added h1 tag
[aaagoryaynova@10 hello]$ git tag
v1
v1-beta
[aaagoryaynova@10 hello]$ git log master --all
error: switch 'l' expects числовое значение
[aaagoryaynova@10 hello]$ git log master --all
commit d9ca72f783bf0975c5a07caf30f2ee6663803ca (tag:
v1, master)
Author: inguzeva <1132226441@pfur.ru>
Date: Tue Feb 18 22:58:26 2025 +0300

    Added HTML header

commit 33baa95b8475de91218c2c710ca58ff8b32308c4 (HEAD
, tag: v1-beta)
```

Рис. 4: Переключение по имени тега и просмотр доступных тегов

Добавления нежелательного комментария

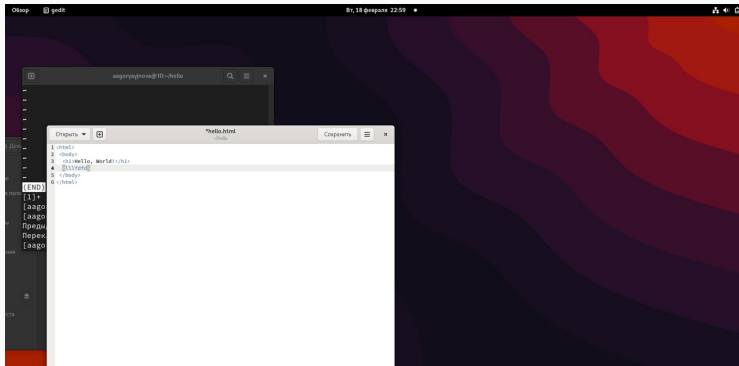


Рис. 5: Добавления нежелательного комментария

Отмена проиндексированных изменений (перед коммитом)

Используем команду `git reset`, чтобы сбросить буферную зону к HEAD. Это очищает буферную зону от изменений, которые мы только что проиндексировали.

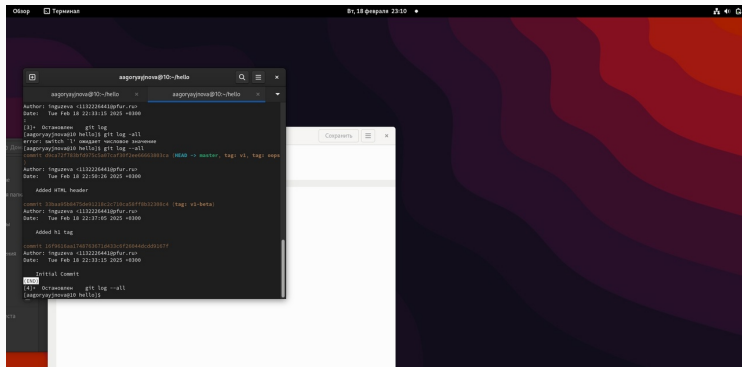
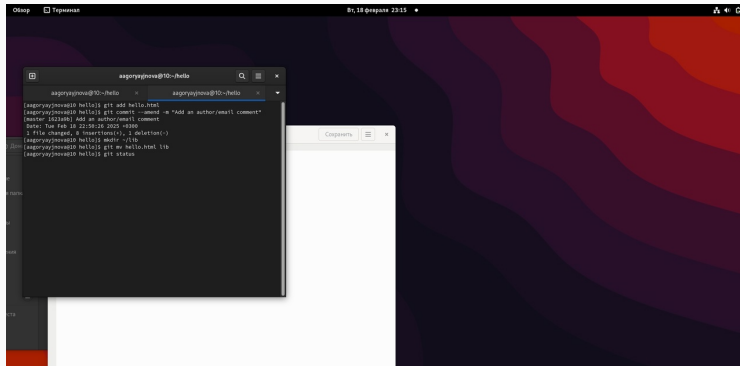


Рис. 6: Отмена коммитов

Удалим последние два коммита с помощью сброса, сначала отметим последний коммит тегом, чтобы его можно было потом найти. Используем команду `git reset`, чтобы вернуться к версии до ЭТИХ КОММИТОВ.

Удалим тег oops и коммиты, на которые он ссылался, сборщиком мусора. Теперь этот тег не отображается в репозитории.

Изменение предыдущего коммита



The screenshot shows a terminal window titled "Терминал" with the prompt "aagoryajnova@10:/hello". The user has executed the following commands:

```
[aagoryajnova@10:/hello]$ git add hello.html
[aagoryajnova@10:/hello]$ git commit --amend -m "Add an author/email comment"
[master 1621a8b] Add an author/email comment
Date: Tue Feb 18 22:28:26 2025 +0300
1 file changed, 8 insertions(+), 1 deletion(-)
```

The terminal output shows the commit message "Add an author/email comment" and the date "Tue Feb 18 22:28:26 2025 +0300". The commit is now on the master branch with hash 1621a8b. The user then executes the following commands:

```
[aagoryajnova@10:/hello]$ mkdir -p lib
[aagoryajnova@10:/hello]$ git mv hello.html lib
[aagoryajnova@10:/hello]$ git status
```

The terminal output shows the creation of the lib directory and the moving of the hello.html file to lib. The user then executes the following command:

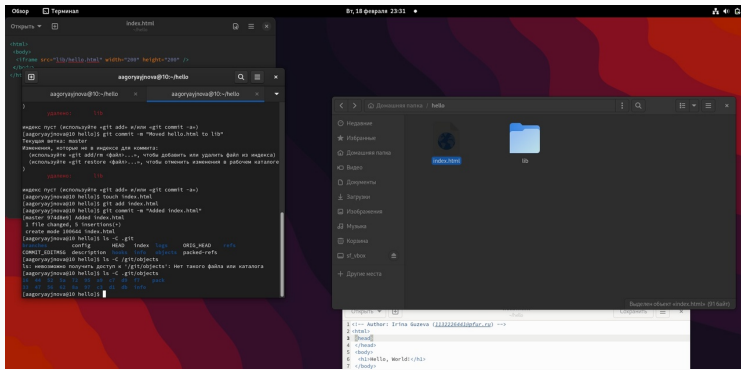
```
[aagoryajnova@10:/hello]$ git status
```

Рис. 7: Изменение предыдущего коммита

Переместим наш файл в каталог `lib`. Для этого создадим его и используем команду `git mv`, сделаем коммит этого перемещения.

Добавим файл index.html в наш репозиторий

```
<html>
  <body>
    <iframe src="lib/hello.html" width="200" height="200" />
  </body>
</html>
```



Git внутри: Каталог .git

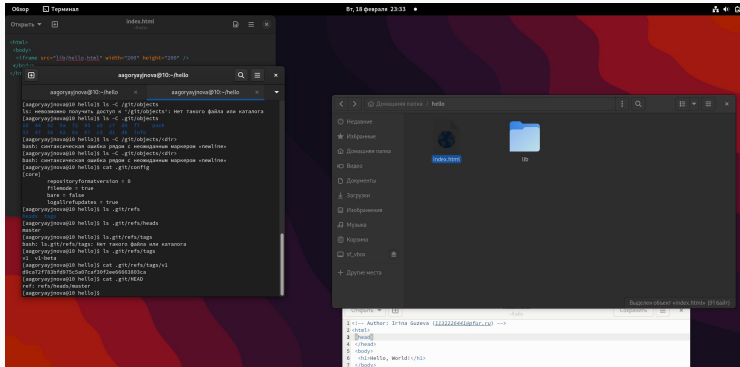


Рис. 9: Каталог .git

Работа непосредственно с объектами git

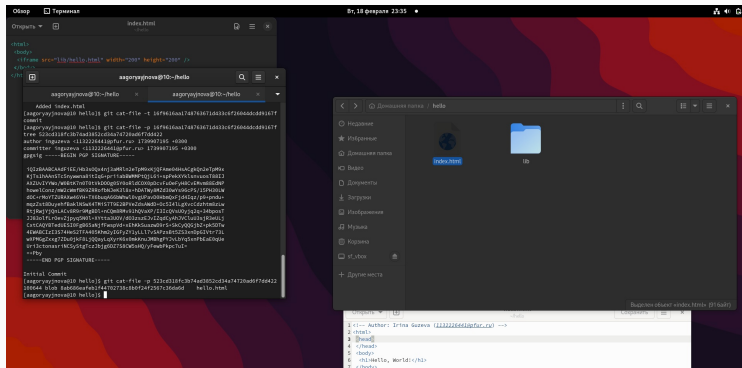


Рис. 10: Работа непосредственно с объектами git

Создание ветки

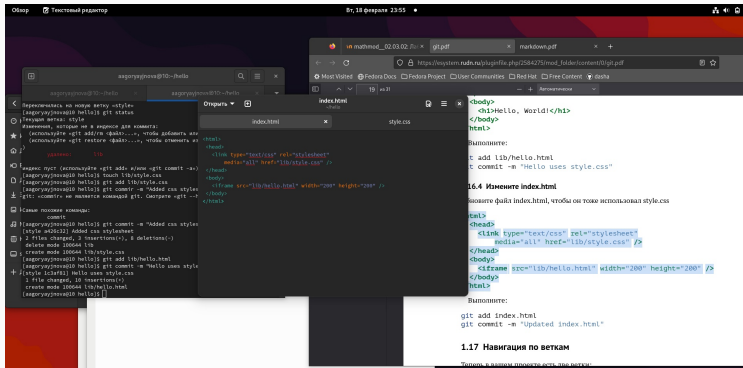


Рис. 11: Редактирование файла

Посмотрим все логи. Переключимся обратно на основную ветку и посмотрим содержимое файла `ib/hello.html`, заметим, что он не использует стили, также посмотрим содержимое этого файла в новой ветке.

Вернемся в основную ветку и добавим файл README.md. Просмотрим ветки и их различия.

Слияние переносит изменения из двух веток в одну. Вернемся к ветке `style` и сольем `master` с `style`.

После коммита «Added README» ветка master была объединена с веткой style, но в настоящее время в master есть дополнительный коммит, который не был слит с style. Последнее изменение в master конфликтует с некоторыми изменениями в style.

Вернемся к ветке `style` и попытаемся объединить ее с новой веткой `master`. Внесем изменения в `lib/hello.html`, оставив только необходимую нам запись и добавим этот файл в репозиторий, чтобы вручную разрешить конфликт.

Вернемся на ветке style к точке перед тем, как мы слили ее с веткой master. Мы видим, что коммит «Updated index.html» был последним на ветке style перед слиянием. Сбросим ветку style к этому коммиту.

Коммит «Added README» идет непосредственно перед коммитом конфликтующего интерактивного режима. Мы сбросим ветку master к коммиту «Added README».

Вернемся в ветку master и сольем ветку style в неё с помощью команды `git merge`.

Перейдем в наш рабочий каталог и сделаем клон репозитория `hello`, затем создадим клон репозитория. Просмотрев его увидим список всех файлов на верхнем уровне оригинального репозитория `README.md`, `index.html` и `lib`.

Что такое origin?

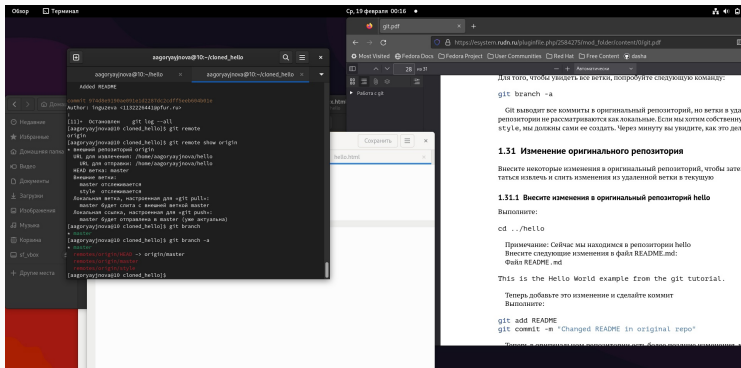


Рис. 12: Просмотр имени по умолчанию удаленного репозитория

Посмотрим на ветки, доступные в нашем клонированном репозитории. Можно увидеть, что в списке только ветка master.

Перейдём в клон репозитория и используем команду `git fetch`, которая будет извлекать новые коммиты из удаленного репозитория, но не будет сливать их с наработками в локальных ветках.

Сольем внесённые изменения в главную ветку. Также можно было бы использовать команду `git pull`, которая является объединением `fetch` и `merge` в одну команду.

Добавим локальную ветку, которая отслеживает удаленную ветку, теперь мы можем видеть ветку `style` в списке веток и логе.

Создание чистого репозитория

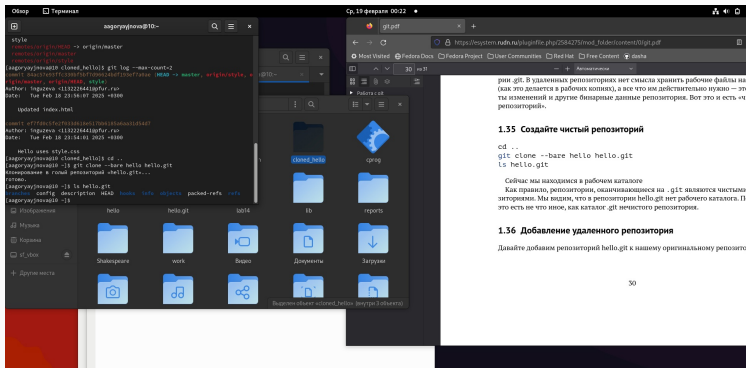


Рис. 13: Создание чистого репозитория

Отправка и извлечение изменений

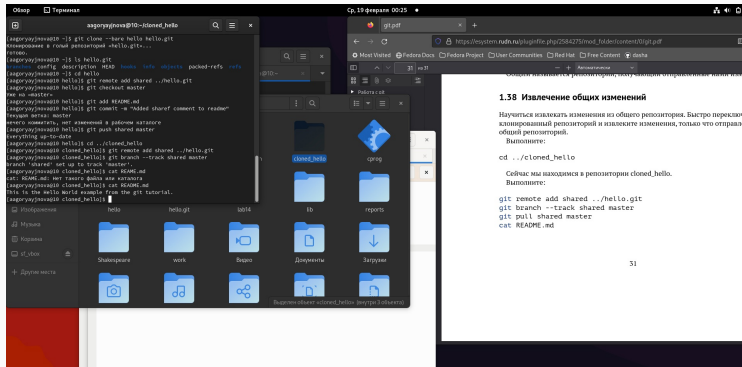


Рис. 14: Извлечение изменений

Выводы

В процессе выполнения данной лабораторной работы я приобрел практические навыки работы с Git.