Лабораторная работа №3

Дисциплина: Операционные системы

Кочуров Арсений Владимирович

Содержание

# Цель работы

Цель данной лабораторной работы — Научиться оформлять отчёты с помощью легковесного языка разметки Markdown.

# Задание

1.Сделайте отчёт по предыдущей лабораторной работе в формате Markdown. 2.В качестве отчёта предоставить отчёты в 3 форматах: pdf,docx и md

# Выполнение лабораторной работы

1). Создаем учетную запись на https://github.com. (иллюстрация на рис. 1)

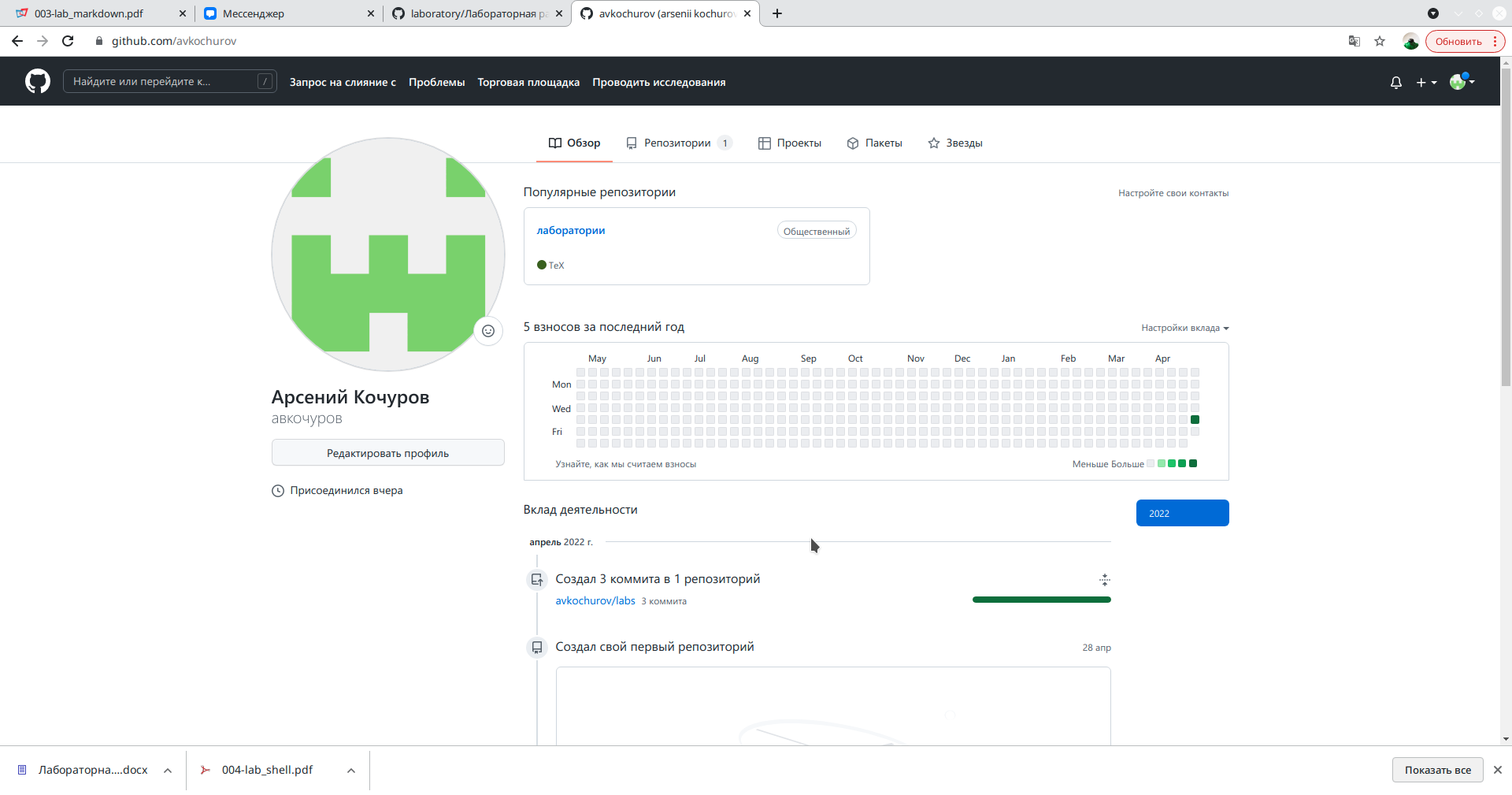


Figure 1: Создание учётной записи

2). Настраиваем систему контроля версий git. Синхранизируем учётную запись github с компьютером: git config –global user.name“Имя Фамилия” git config –global user.email“work@mail” После этого создаём новый ключ на github (команда ssh-keygen -C“TanyaKonovalova [tanya.konovalova.02@mail](mailto:tanya.konovalova.02@mail)”) и привязываем его к копьютеру через консоль (алгоритм действий представлен на рис.2).

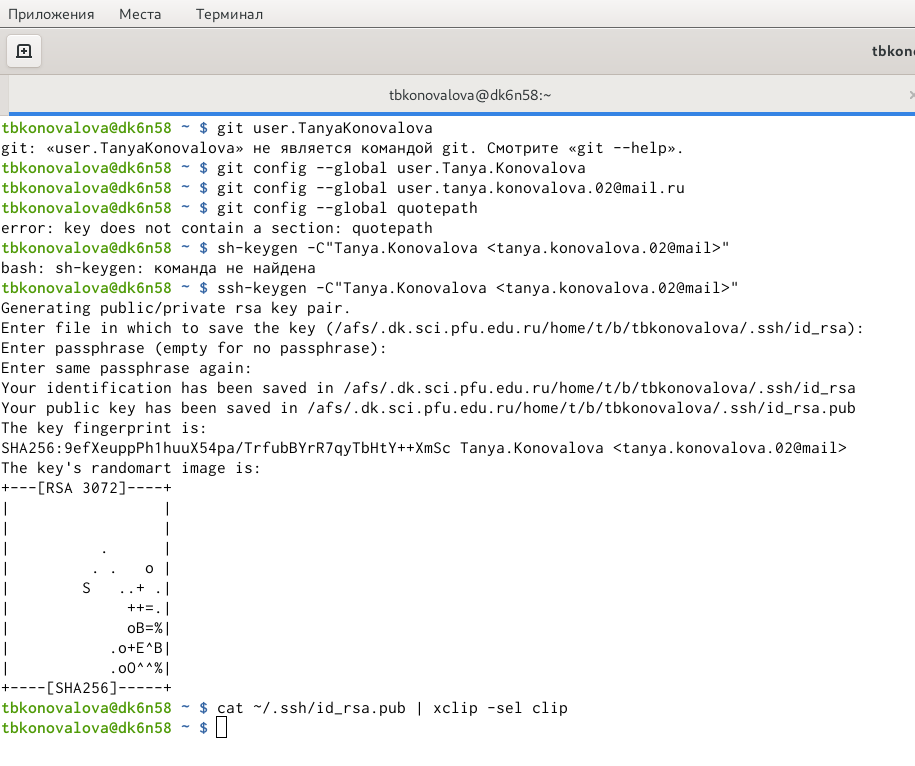
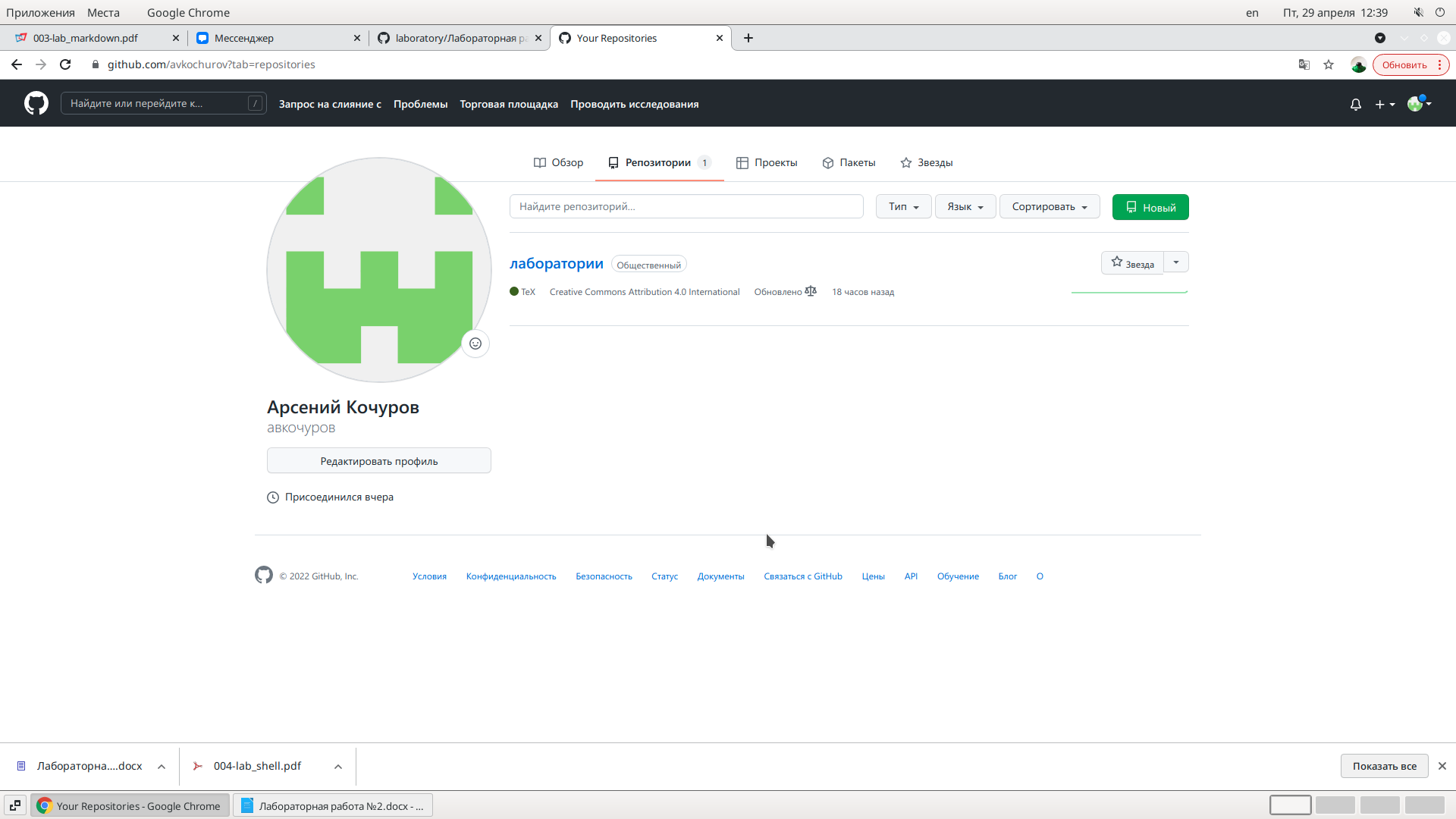
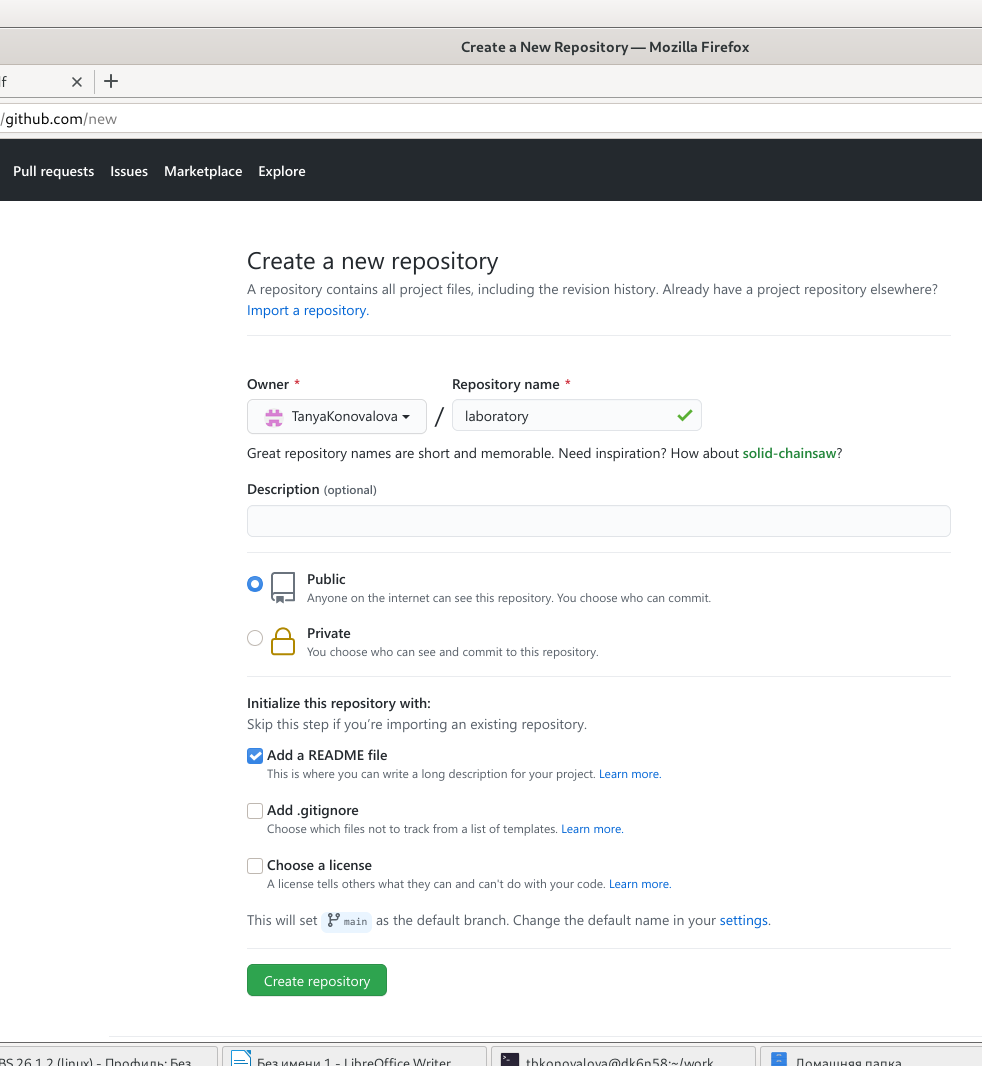
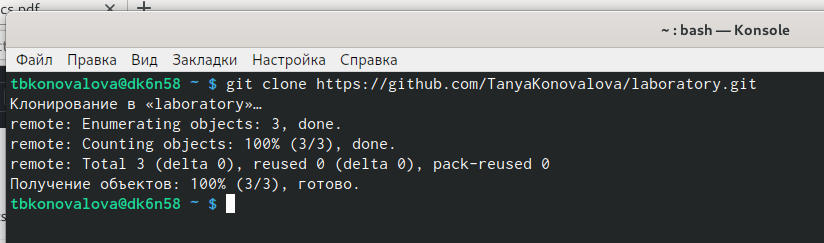


Figure 2: Создание нового ключа

3). Следующим шагом будет создание и подключение репозитория к github. В gethup заходим в «repository» и создаём новый репозиторий (имя «laboratory», а зоголовок для файла README) (иллюстрация представлена на рис. 3.1, 3.2). Копируем в консоль ссылку на репозиторий (для дальнейшей работы с файлами) (алгоритм действий см. на рис.3.3):

4). В лабораторной работе описан алгоритм создания структуры каталога через консоль. Но легче будет создать репозиторий в github и после этого работать с каталогом и папками через консоль (перед этим необходимо скопировать ссылку на репозиторий в консоль, в формате https или ssh). Перед тем, как создавать файлы, заходим в наш репозиторий (алгоритм действий представлен на рис. 4.1):

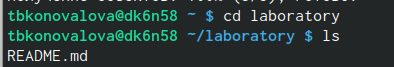


Figure 3: Создание репозитория

После этого можем уже создавать наши файлы:

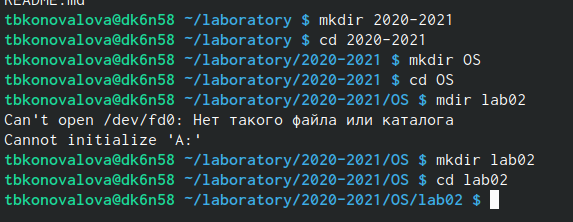


Figure 4: Создание файлов

5). Добавляем первый коммит и выкладываем на github. Для того, чтобы правильно разместить первый коммит, необходимо добавить команду git add ., после этого с помощью команды git commit -am “first commit” выкладываем коммит:

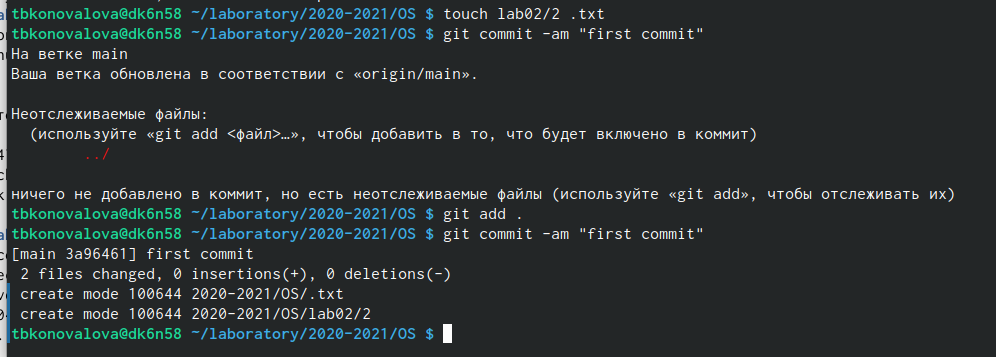


Figure 5: Добавление первого коммита

6). Сохраняем первый коммит, используя команду git push:

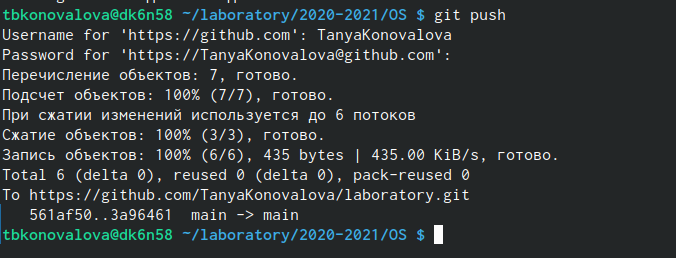


Figure 6: Сохранение первого коммита

7). Первичная конфигурация (Пояснения к каждому действию алгоритма представлены на рис. 71-74): a. Добавляем файл лицензии;

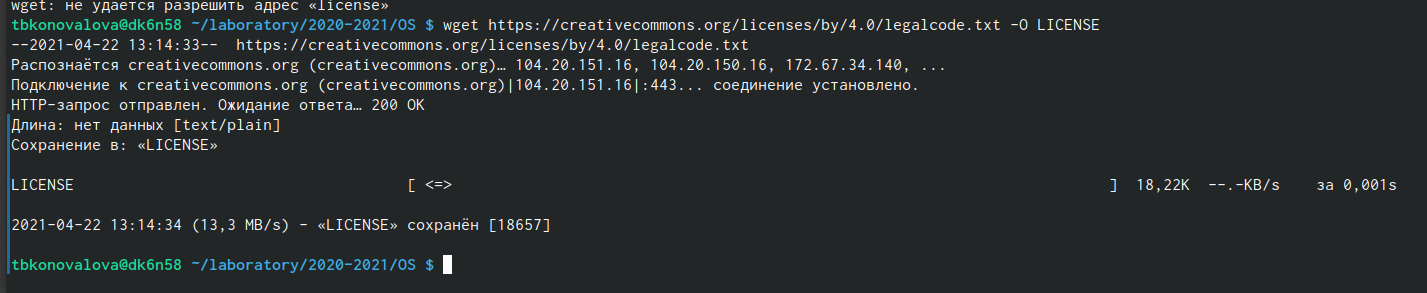


Figure 7: Добавление файла лицензии

1. Добавим шаблон игнорируемых файлов. Просмотрим список имеющихся шаблонов (на скриншоте список шаблонов представлен не в целом виде):

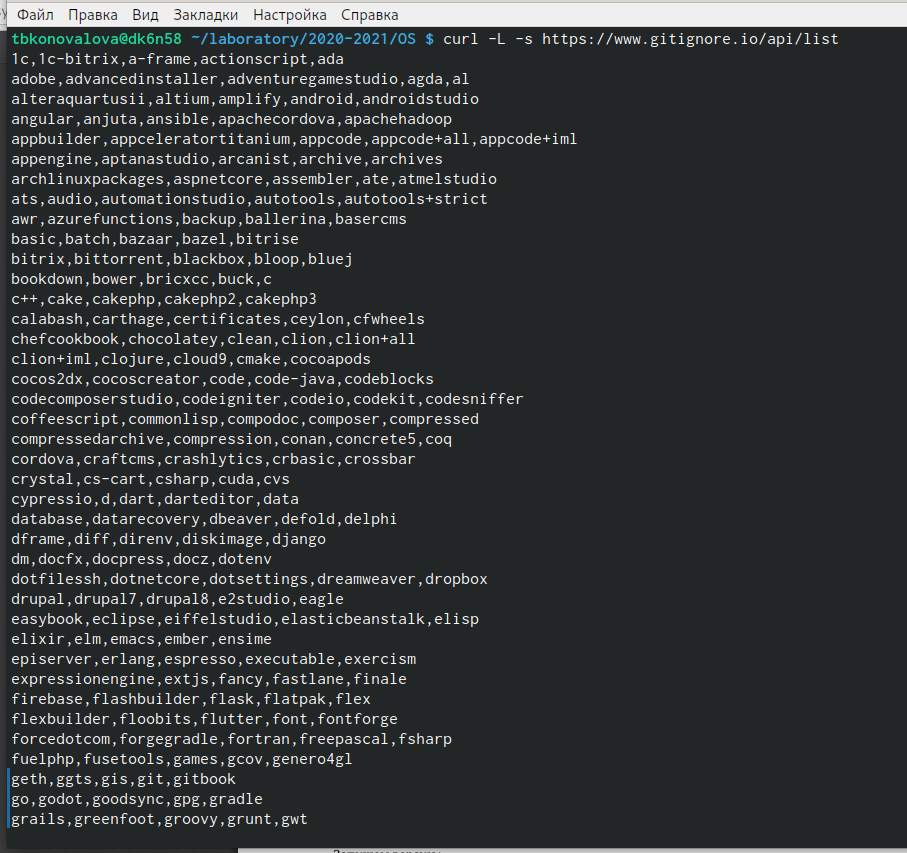


Figure 8: Добавление шаблона игнорируемых файлов

1. Скачиваем шаблон, например, для C. Также добавляем новые файлы и выполняем коммит:

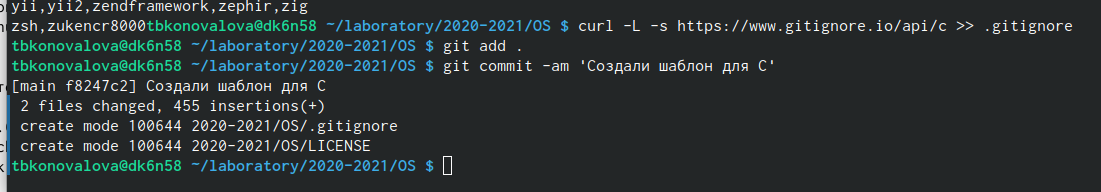


Figure 9: Скачивание шаблона на C

1. Отправим на github (для этого сохраним все созданные шаблоны и файлы, используя команду git push):

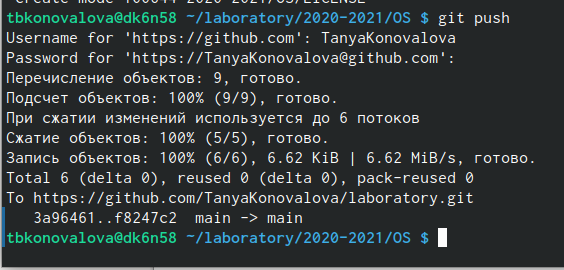


Figure 10: Отправляем на GitHub созданные шаблоны

8). Работаем с конфигурацией git-flow (алгоритм к каждому действию представлен на рис. 81-87): a. Инициализируем git-flow, используя команду git flow init -f (префикс для ярлыков установлен в v):

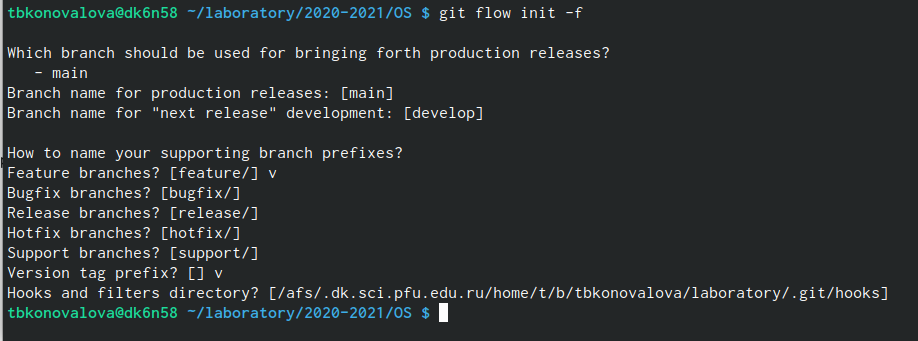
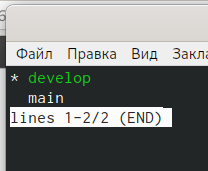


Figure 11: Инициализация git-flow

1. Проверяем, что мы находимся на ветке develop (используем команду git branch):

Проверка 

1. Создаём релиз с версией 1.0.0:



Figure 12: Создание релиза

1. Запишем версию и добавим в индекс: echo’hello world’> hello.txt git add hello.txt git commit -am’Новый файл

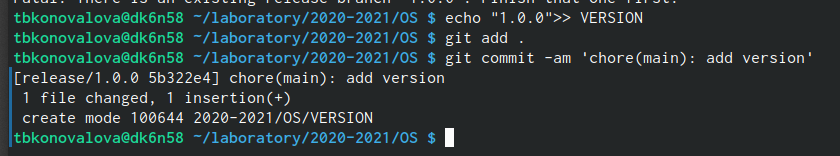


Figure 13: Записывание версии и добавление в индекс

1. Заливаем релизнутую ветку в основную ветку (используем команду git flow release finish1.0.0):

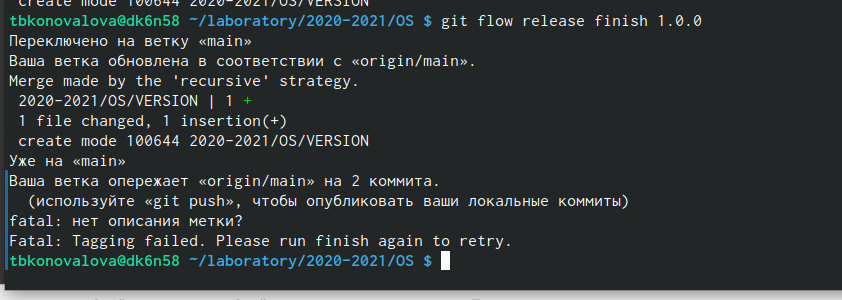


Figure 14: Заливаем ветку

1. Отправляем данные на github: git push - -all git push - -tags

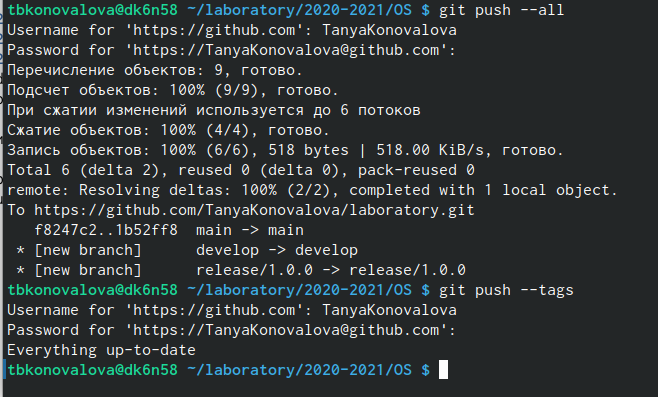
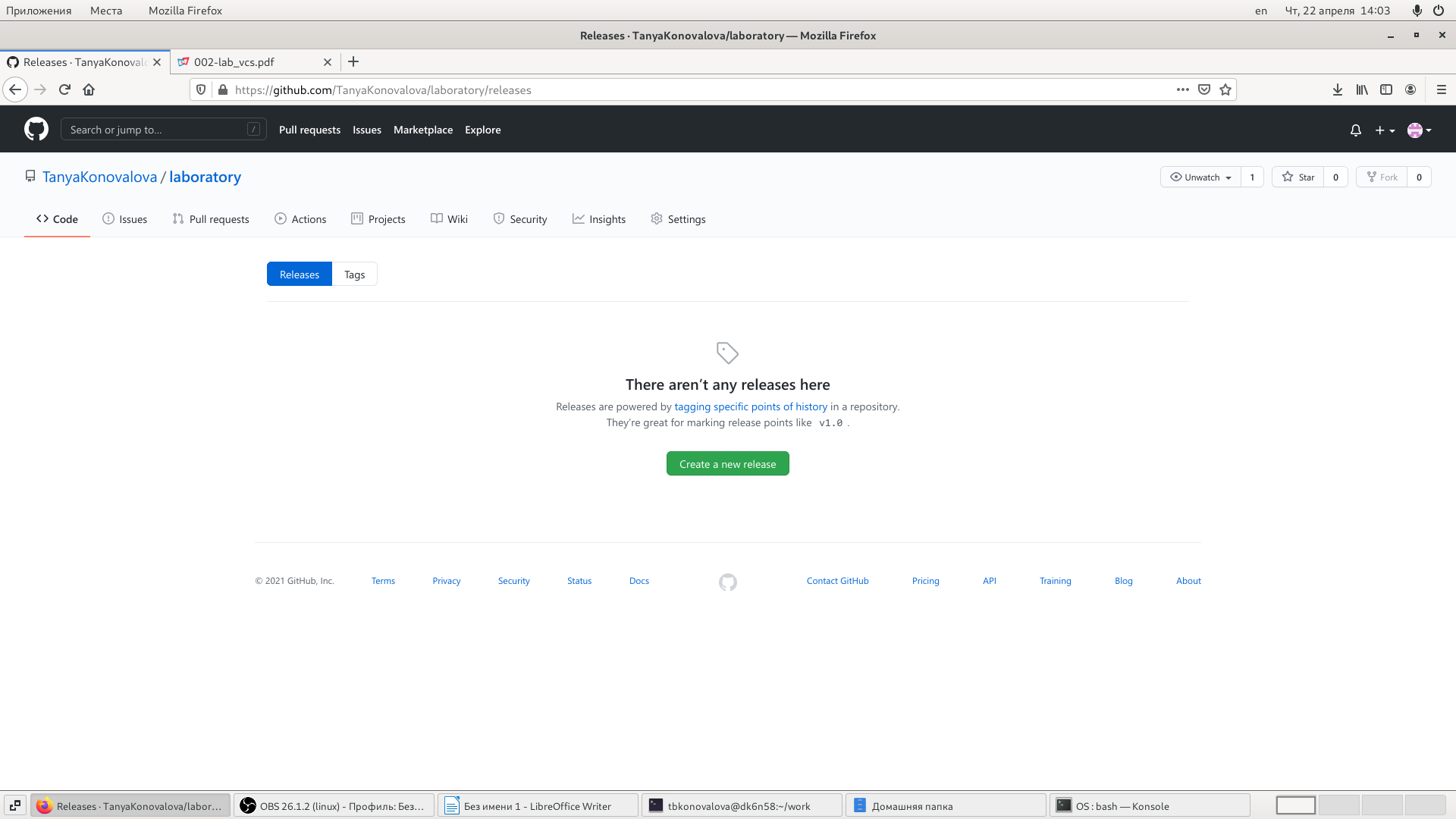
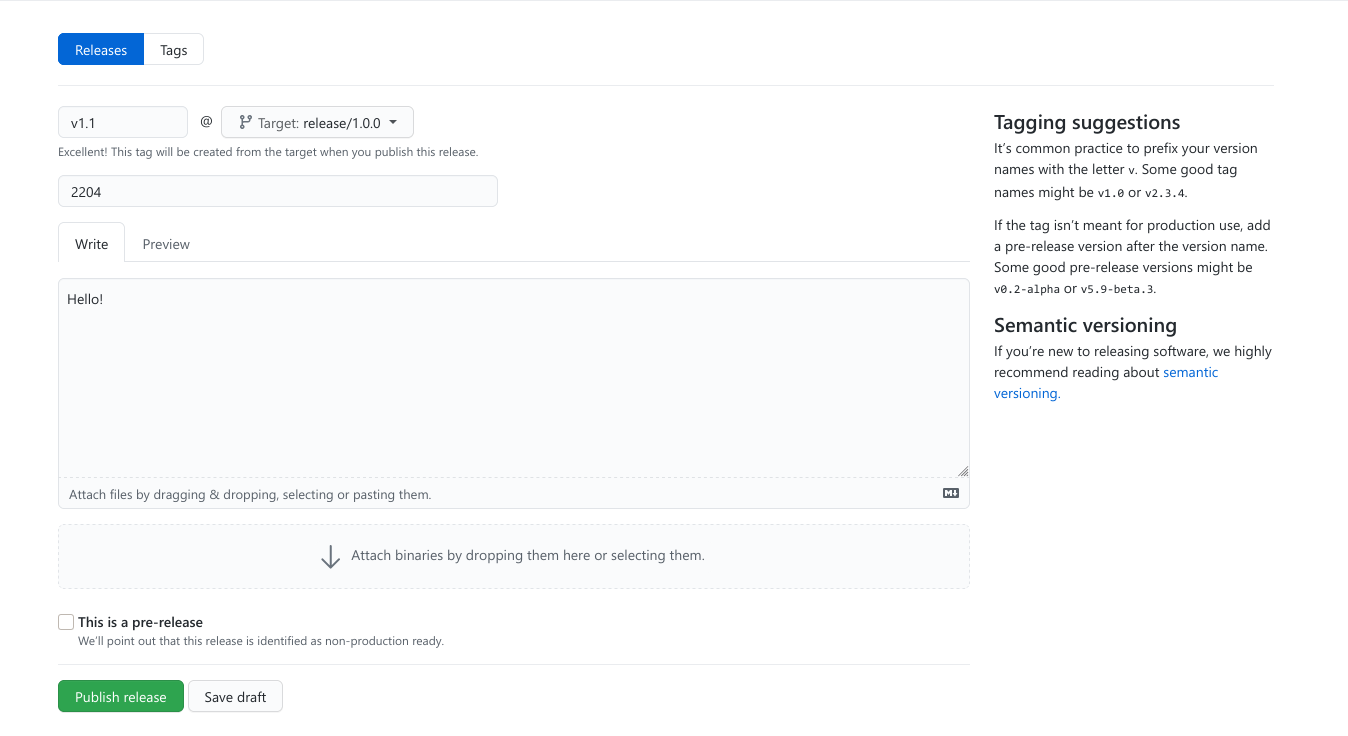
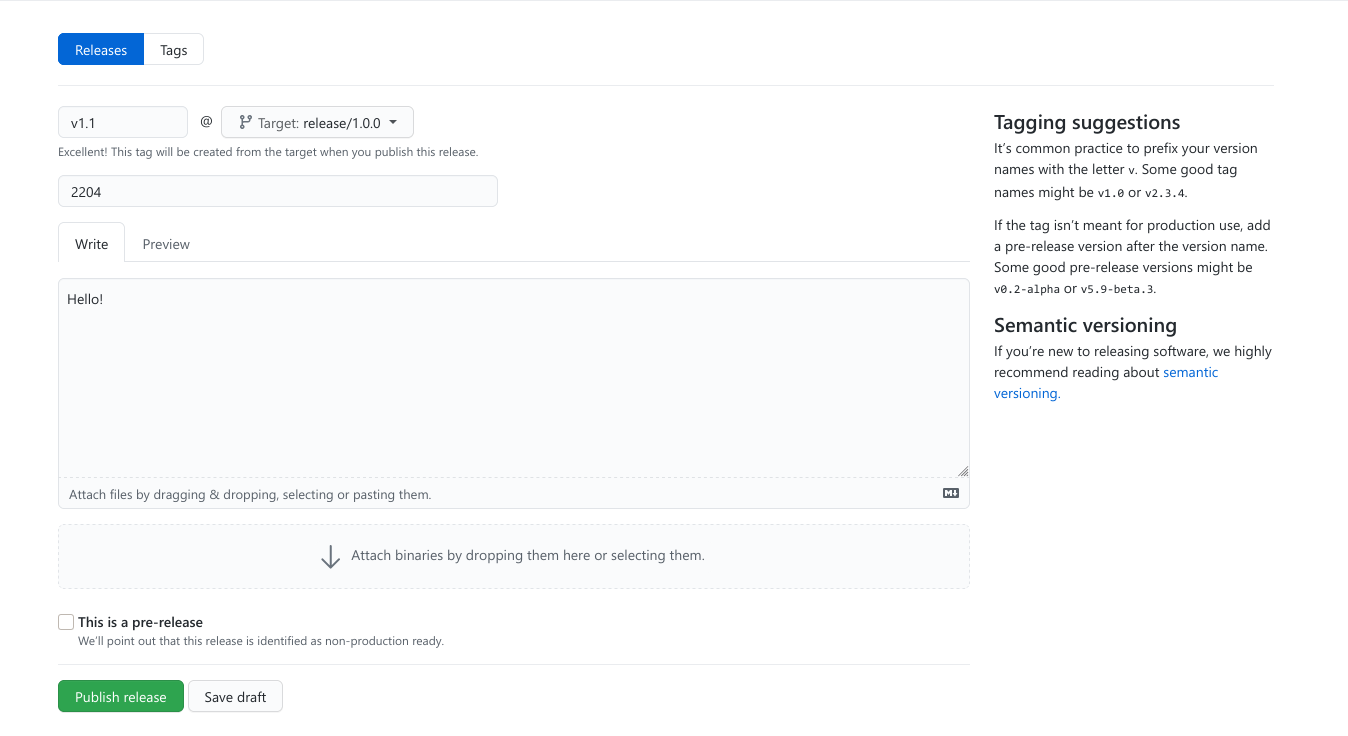


Figure 15: Отправляем данные на GitHub

9). Создаем релиз на github. Для этого заходим в «Releases», нажимаем «Создать новый релиз». Заходим в теги и заполняем все поля (создаём теги для версии 1.0.0). После создания тега, автоматически сформируется релиз (подробное изложение действий представлено на рис. 91-93).

# Выводы

Изучил идеологию и научилась применять средства контроля версий.