Лабораторная работа №3

Дисциплина: Операционные системы

Егорова Александра

Содержание

# Цель работы

Изучить идеологию и применение средств контроля версий.

# Задание

Сделайте отчёт по предыдущей лабораторной работе в формате Markdown.

# Выполнение лабораторной работы

Создаем учётную запись на https://github.com. (рис. -fig. 1)

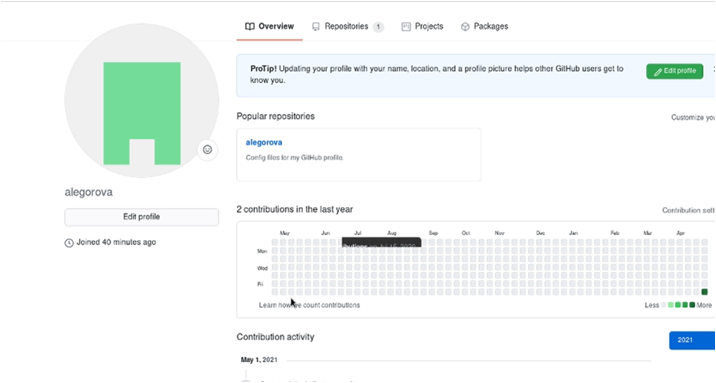


Figure 1: Рис.1

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” (рис. -fig. 2)

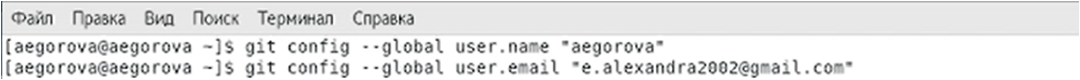


Figure 2: Рис.2

Создаем новый ключ на github и привязываю его к компьютеру через консоль. (рис. -fig. 3) (рис. -fig. 4) (рис. -fig. 5) (рис. -fig. 6)

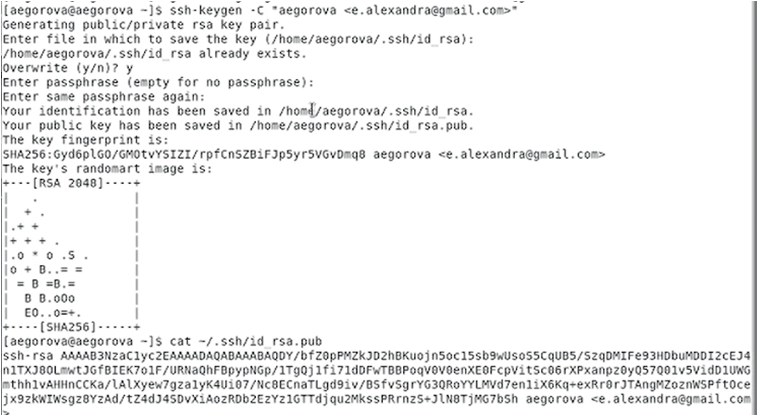


Figure 3: Рис.3

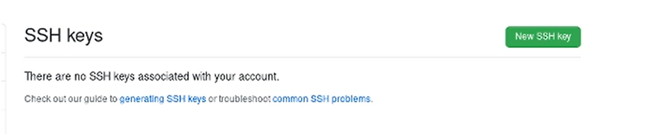


Figure 4: Рис.4

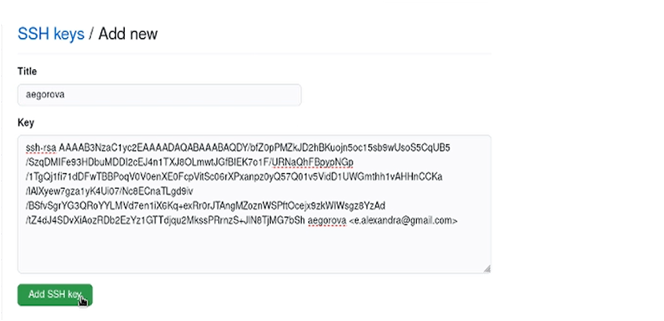


Figure 5: Рис.5



Figure 6: Рис.6

Подключение репозитория к github.Заходим в «repositories» и создаем новый репозиторий (имя «labor2»). Копируем в консоль ссылку на репозиторий. (рис. -fig. 7) (рис. -fig. 8)



Figure 7: Рис.7

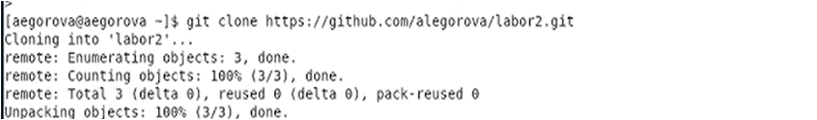


Figure 8: Рис.8

Работаем с каталогом и папками через консоль. Создаем файлы. (рис. -fig. 9) (рис. -fig. 10)



Figure 9: Рис.9

Figure 10: Рис.10

Figure 10: Рис.10

Добавляем первый коммит и выкладываем на githup. Чтобы правильно разместить первый коммит, необходимо добавить команду git add . , далее с помощью команды git commit -am “first commit” выкладываем коммит. Сохраняем первый коммит (git push). (рис. -fig. 11) (рис. -fig. 12)

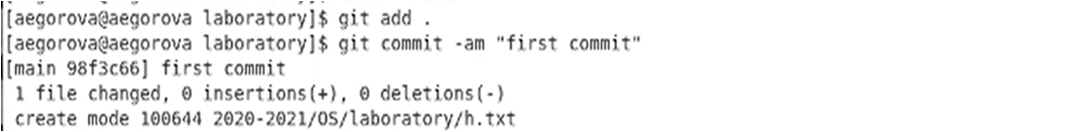


Figure 11: Рис.11

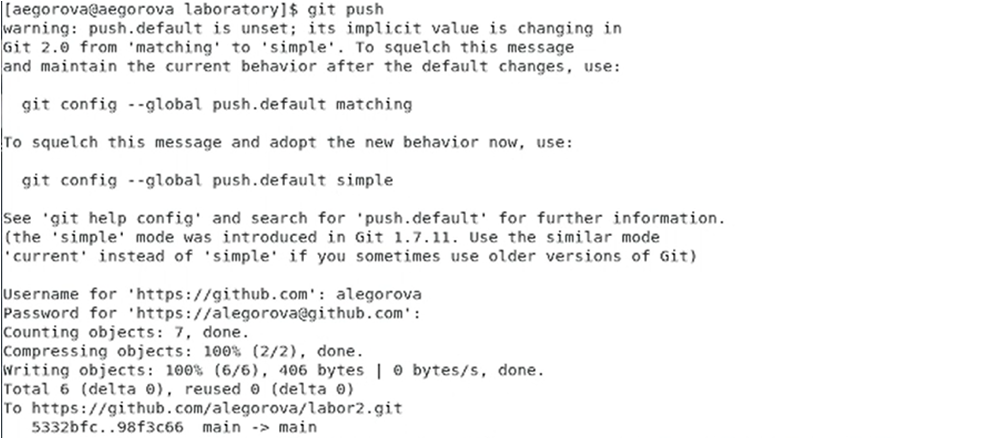


Figure 12: Рис.12

Первичная конфигурация. Добавим файл лицензии. (рис. -fig. 13)



Figure 13: Рис.13

Добавим шаблон игнорируемых файлов. Получим список имеющихся шаблонов (на скрине представлены не все шаблоны). (рис. -fig. 14)



Figure 14: Рис.14

Затем скачаем шаблон, например, для C. Также добавим новые файлы и выполним коммит. Отправляем на github (git push). (рис. -fig. 15) (рис. -fig. 16)



Figure 15: Рис.15

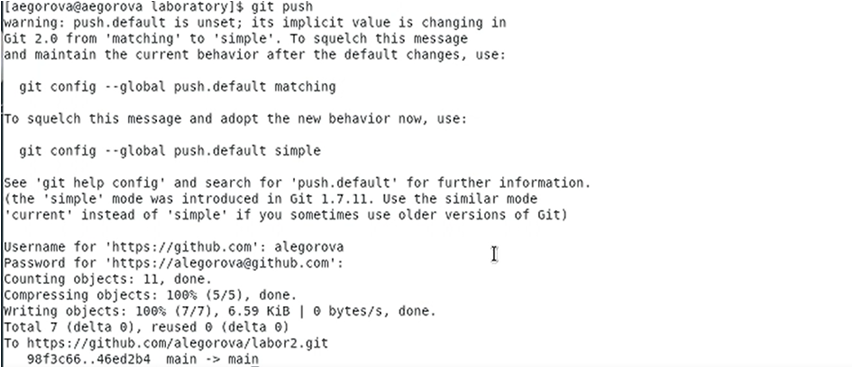


Figure 16: Рис.16

Конфигурация git-flow. Инициализируем git-flow с помощью команды git flow init -f. Префикс для ярлыков установим в v. (рис. -fig. 17)

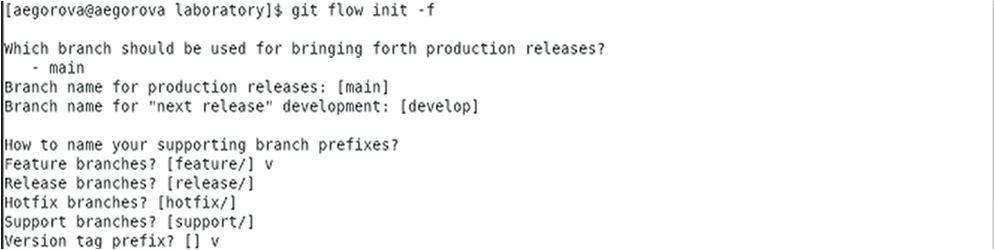


Figure 17: Рис.17

Проверяем, что находимся на ветке develop (git branch). (рис. -fig. 18)

Figure 18: Рис.18

Figure 18: Рис.18

Создадим релиз с версией 1.0.0. Запишем версию и добавим в индекс. (рис. -fig. 19) (рис. -fig. 20)



Figure 19: Рис.19

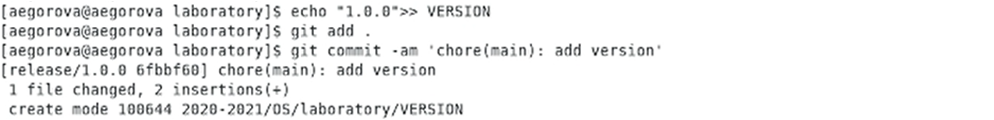


Figure 20: Рис.20

Зальём релизную ветку в основную ветку. Отправим данные на github. (рис. -fig. 21) (рис. -fig. 22)

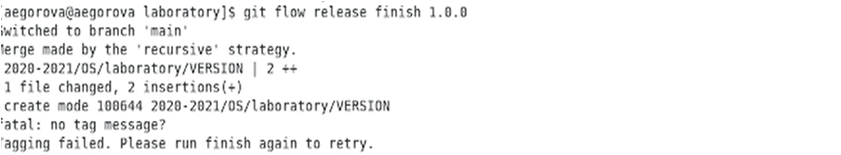


Figure 21: Рис.21



Figure 22: Рис.22

Создаем релиз на github. Заходим в «Releases», нажимаем «Создать новый релиз». Заходим в теги и заполняем все поля. После создания тега, автоматически сформируется релиз. (рис. -fig. 23) (рис. -fig. 24)



Figure 23: Рис.23



Figure 24: Рис.24

# Выводы

В ходе выполнения лабораторной работы я изучила идеологию и применение средств контроля версий.