Лабораторная работа №3

Егорова Александра

# Цель работы

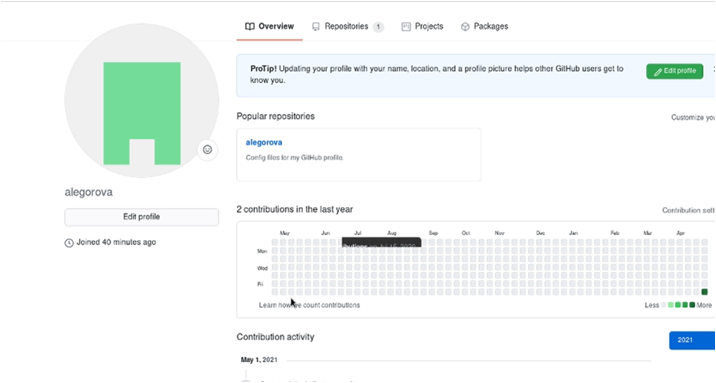
Изучить идеологию и применение средств контроля версий.

# Задание

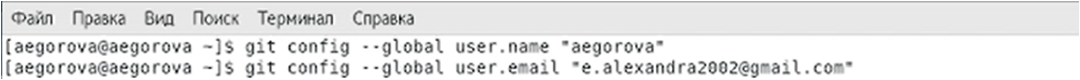
Сделайте отчёт по предыдущей лабораторной работе в формате Markdown.

# Выполнение лабораторной работы

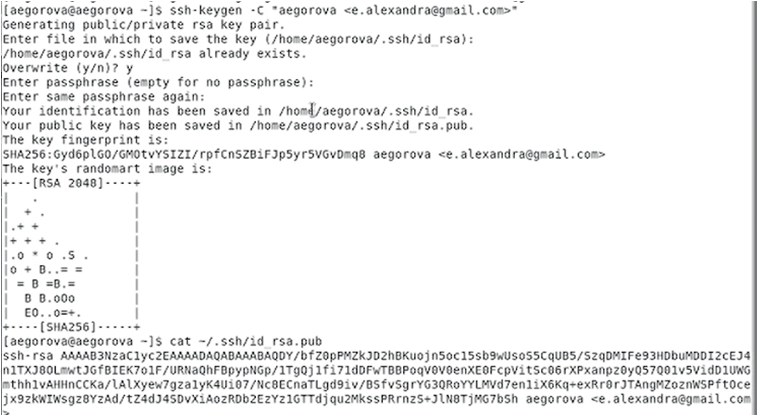
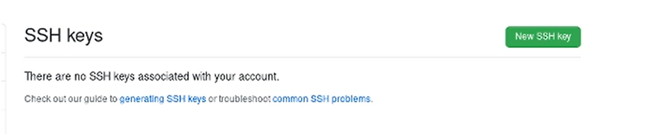
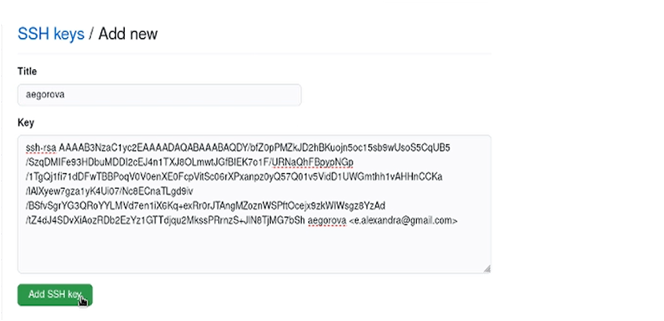
Создаем учётную запись на https://github.com. (рис. @fig:001)

{ #fig:001 width=70% }

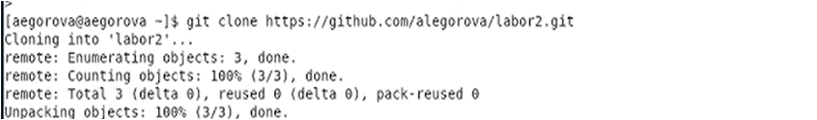
Настраиваем систему контроля версий git. Синхранизируем учётную запись github с компьютером: git config --global user.name "Имя Фамилия" git config --global user.email "work@mail" (рис. @fig:002)

{ #fig:002 width=70% }

Создаем новый ключ на github и привязываю его к компьютеру через консоль. (рис. @fig:003) (рис. @fig:004) (рис. @fig:005) (рис. @fig:006)

{ #fig:003 width=70% } { #fig:004 width=70% } { #fig:005 width=70% } { #fig:006 width=70% }

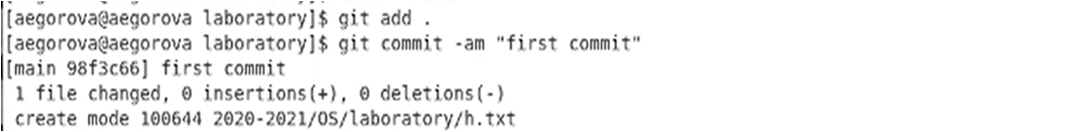
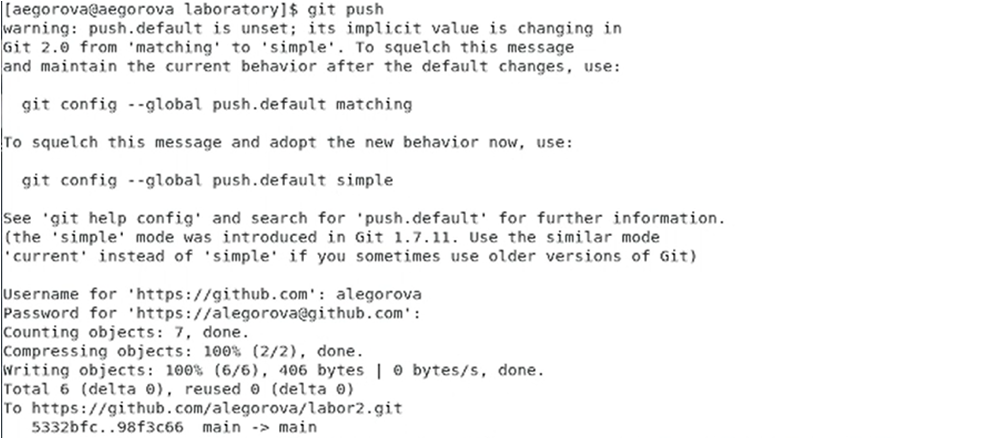
Подключение репозитория к github.Заходим в «repositories» и создаем новый репозиторий (имя «labor2»). (рис. @fig:007) Копируем в консоль ссылку на репозиторий.(рис. @fig:008)

{ #fig:007 width=70% } { #fig:008 width=70% }

Работаем с каталогом и папками через консоль.(рис. @fig:009) Создаем файлы. (рис. @fig:010)

{ #fig:009 width=70% } { #fig:010 width=70% }

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. (рис. @fig:011) Сохраняем первый коммит (git push). (рис. @fig:012)

{ #fig:011 width=70% } { #fig:012 width=70% }

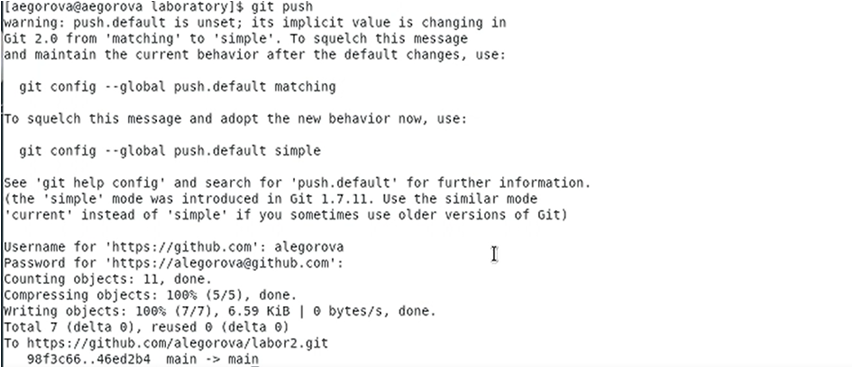
Первичная конфигурация. Добавим файл лицензии. (рис. @fig:013)

{ #fig:013 width=70% }

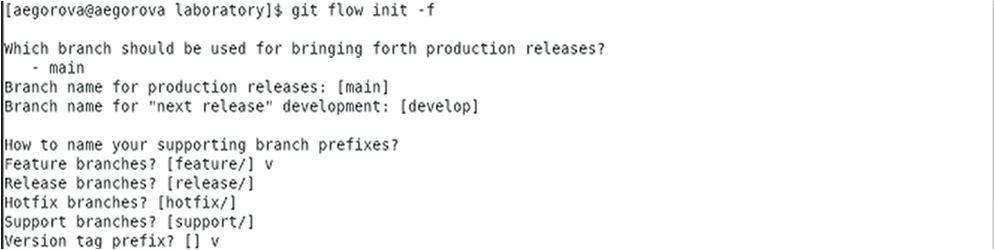
Добавим шаблон игнорируемых файлов. Получим список имеющихся шаблонов (на скрине представлены не все шаблоны). (рис. @fig:014)

{ #fig:014 width=70% }

Затем скачаем шаблон, например, для C. Также добавим новые файлы и выполним коммит. (рис. @fig:015) Отправляем на github (git push). (рис. @fig:016)

{ #fig:015 width=70% } { #fig:016 width=70% }

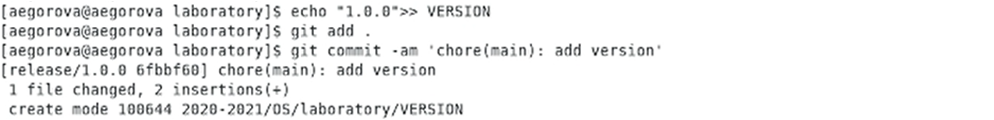
Конфигурация git-flow. Инициализируем git-flow с помощью команды git flow init -f. Префикс для ярлыков установим в v. (рис. @fig:017)

{ #fig:017 width=70% }

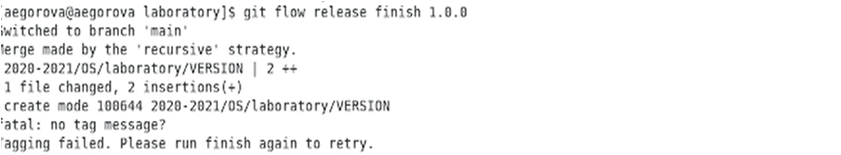
Проверяем, что находимся на ветке develop (git branch). (рис. @fig:018)

{ #fig:018 width=70% }

Создадим релиз с версией 1.0.0. (рис. @fig:019) Запишем версию и добавим в индекс. (рис. @fig:020)

{ #fig:019 width=70% } { #fig:020 width=70% }

Зальём релизную ветку в основную ветку. (рис. @fig:021) Отправим данные на github. (рис. @fig:022)

{ #fig:021 width=70% } { #fig:022 width=70% }

Создаем релиз на github. Заходим в «Releases», нажимаем «Создать новый релиз». Заходим в теги и заполняем все поля. (рис. @fig:023) После создания тега, автоматически сформируется релиз. (рис. @fig:024)

{ #fig:023 width=70% } { #fig:024 width=70% }

# Выводы

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