Отчет к лабораторной работе №3

Дисциплина: операционные системы

Крупенникова Виктория Александровна

Содержание

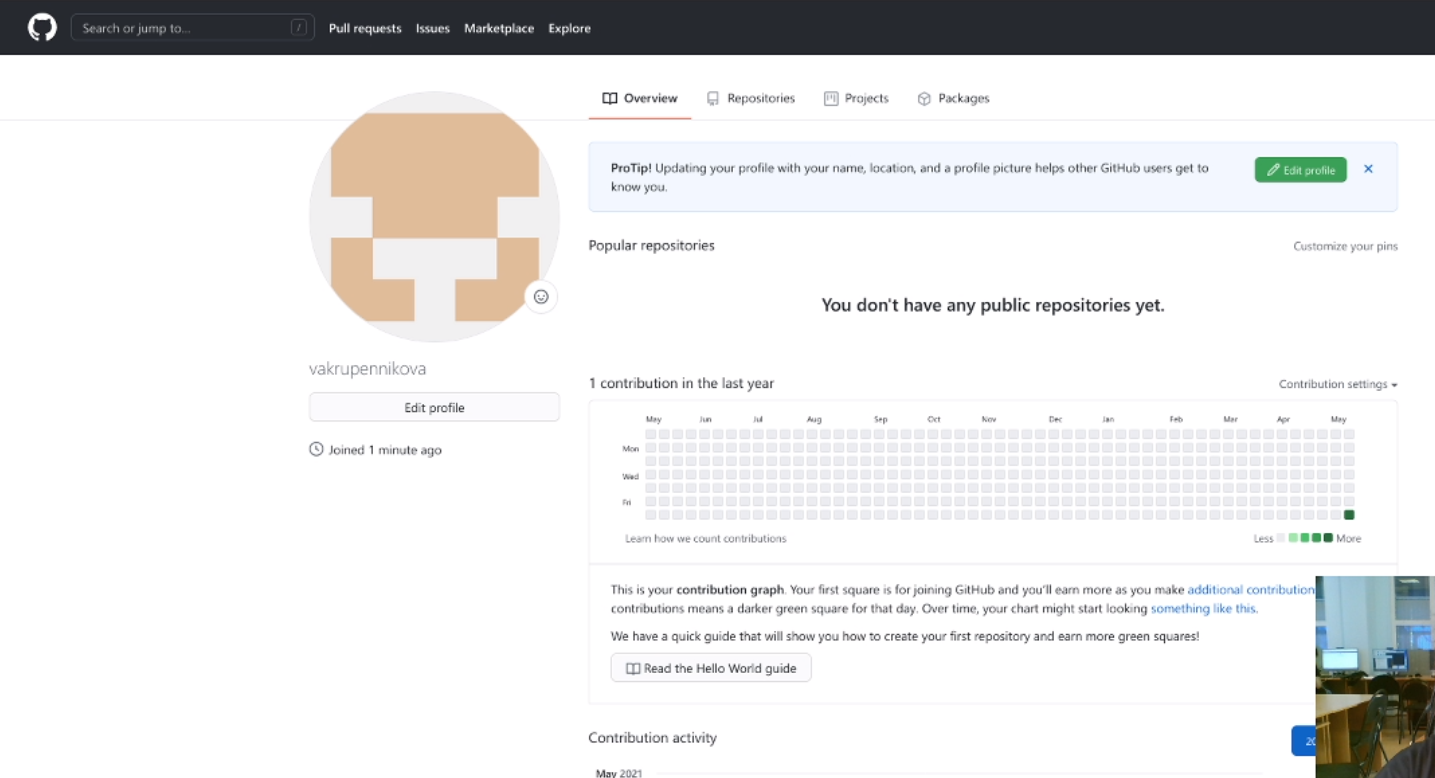
# Цель работы

Изучить идеологию и применение средств контроля версий

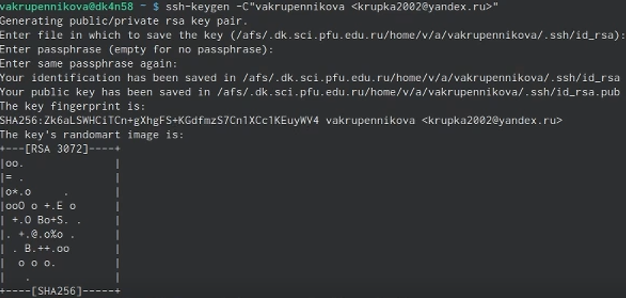
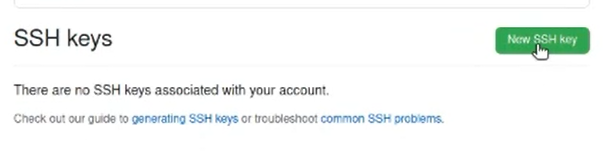
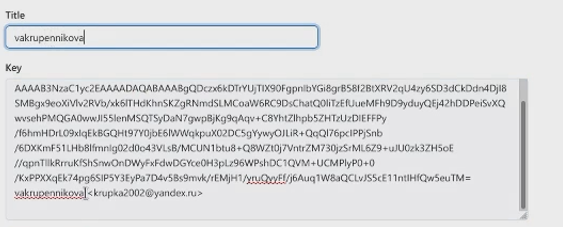
# Задание

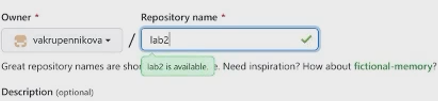
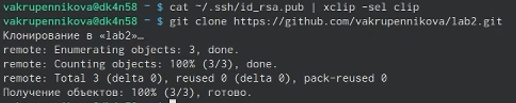
Сделайте отчёт по предыдущей лабораторной работе в формате Markdown.–В качестве отчёта просьба предоставить отчёты в 3 форматах:pdf,docxиmd(вархиве,поскольку он должен содержать скриншоты,Makefile ит.д.

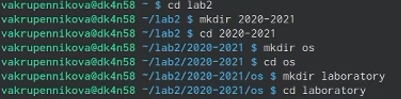
# Выполнение лабораторной работы

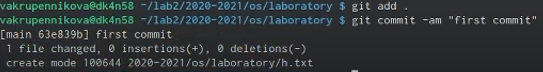
Создаю учетную запись на https://github.com. (рис. **¿fig:001?**) 

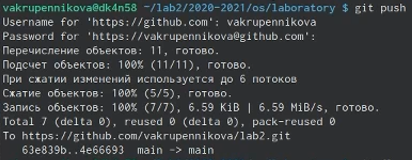
Настраиваю систему контроля версий git. Синхранизирую учётную запись github с компьютером: git config –global user.name “Имя Фамилия” git config –global user.email “work@mail” (рис. **¿fig:002?**) Настройка системы контроля версий

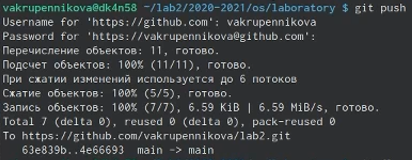
Создаю новый ключ на github (команда ssh-keygen -C “vakrupennikova [krupka2002@yandex.ru](mailto:krupka2002@yandex.ru)”) и привязываю его к копьютеру через консоль. (рис. **¿fig:003?**) (рис. **¿fig:004?**) (рис. **¿fig:005?**) (рис. **¿fig:006?**)    

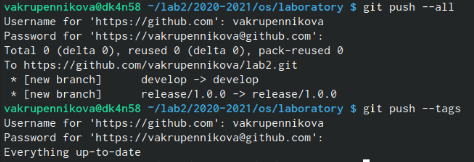
В githup захожу в «repositories» и создаю новый репозиторий. Копируем в консоль ссылку на репозиторий.(рис. **¿fig:007?**)(рис. **¿fig:008?**)  

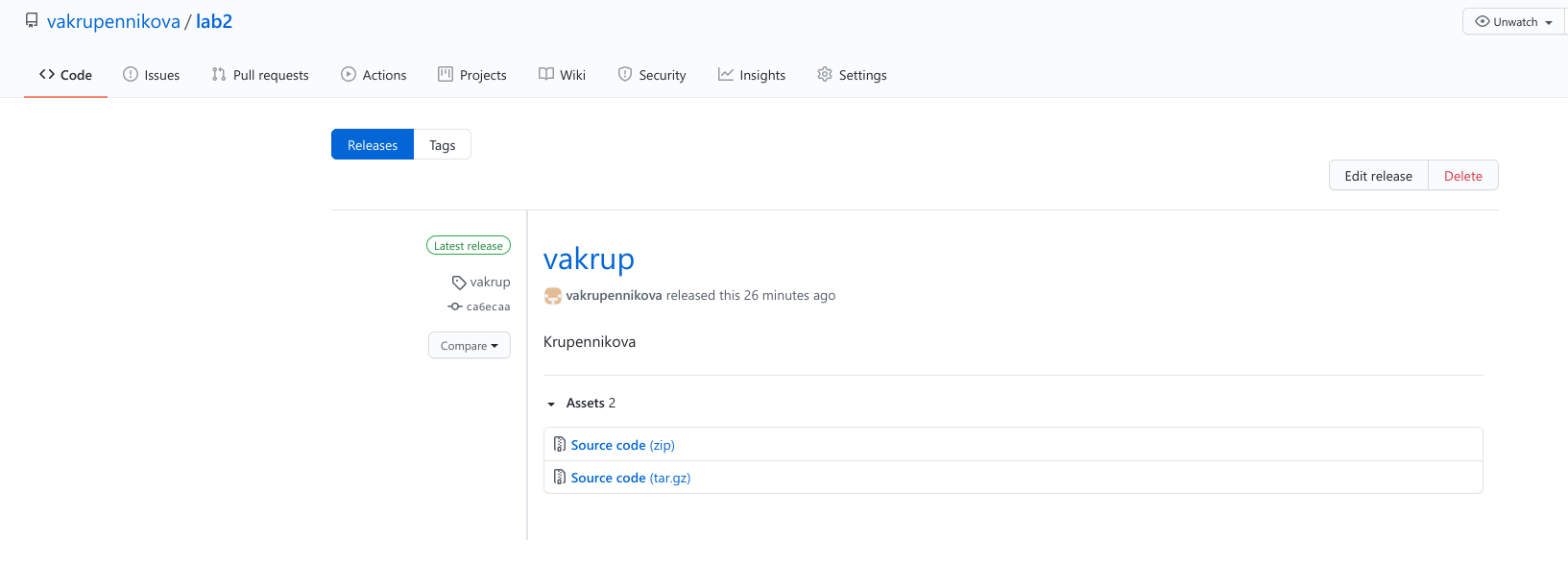
Работаю с каталогом и папками через консоль. Перед тем, как создавать файлы, захожу в репозиорий и создаю файлы: (рис. **¿fig:009?**)(рис. **¿fig:0010?**)  

Добавляю первый коммит и выкладываю на githup. Для того, чтобы правильно разместить первый коммит, необходимо добавить команду git add . , далее с помощью команды git commit -m “first commit” выкладываем коммит:(рис. **¿fig:011?**) 

Сохраняю первый коммит (git push):(рис. **¿fig:012?**) 

Добавляю файл лицензии (скринкаст оборвался, скриншоты тоже). Добавляю шаблон игнорируемых файлов. Получаю список имеющихся шаблонов. Скачиваю шаблон, например, для C. Также добавляю новые файлы и выполняю коммит. Отправляю на github (git push): (рис. **¿fig:013?**) 

Инициализирую git-flow, используя команду git flow init -f (префикс для ярлыков установлен в v): Проверяю, что нахожусь на ветке develop (git branch): Создаю релиз с версией 1.0.0: Записываю версию и добавляю в индекс: echo“1.0.0” >> VERSION git add . git commit -am ‘chore(main): add version’ Заливаю релизную ветку в основную ветку (команда git flow release finish1.0.0): Отправляю данные на github: git push - -all git push - -tags (рис. **¿fig:014?**) 

Создаю релиз на github. Заходим в «Releases», нажимаю «Создать новый релиз». Захожу в теги и заполняю все поля (теги для версии 1.0.0). После создания тега, автоматически сформируется релиз. (рис. **¿fig:015?**) 

# Выводы

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