Отчет по лабораторной работе №3

Дисциплина: Операционные системы

Морозова Анастасия Владимировна

Содержание

# Цель работы

Изучить идеологию и применение средств контроля версий

# Задание

Сделать отчёт по предыдущей лабораторной работе в формате Markdown.В качестве отчёта просьба предоставить отчёты в 3 форматах:pdf,docx и md(в архиве,поскольку он должен содержать скриншоты,Makefile ит.д.)

# Выполнение лабораторной работы

1. **Настройка git**

* Создаю учетную запись на https://github.com. (рис. -fig. 1)

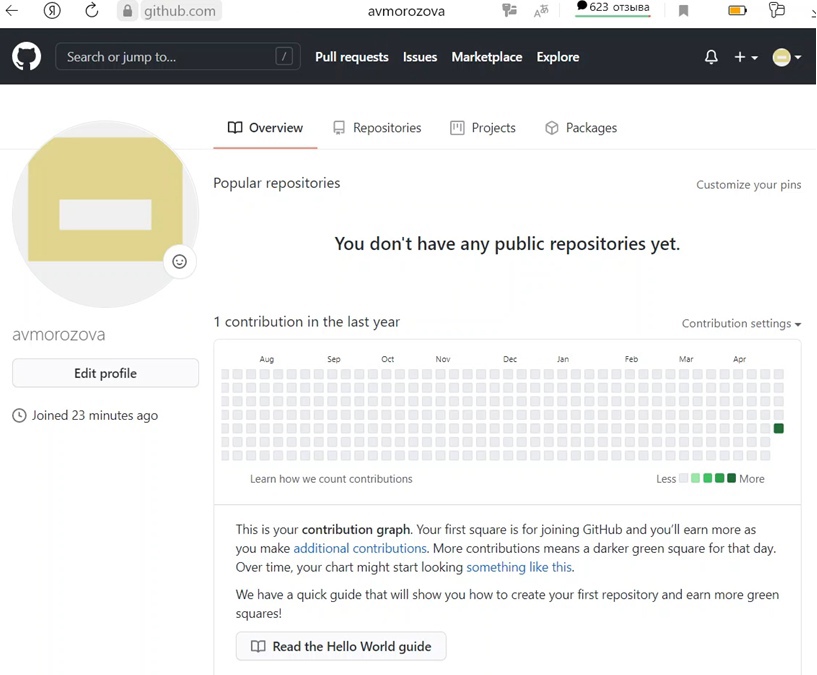


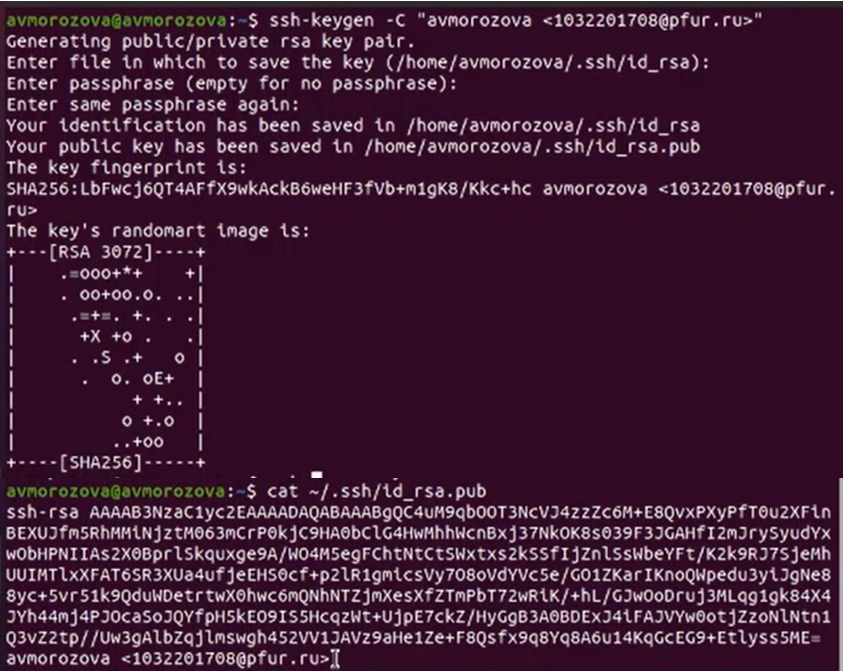
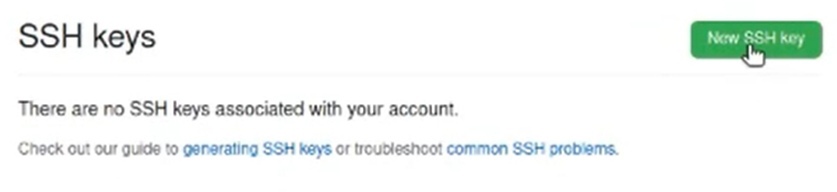
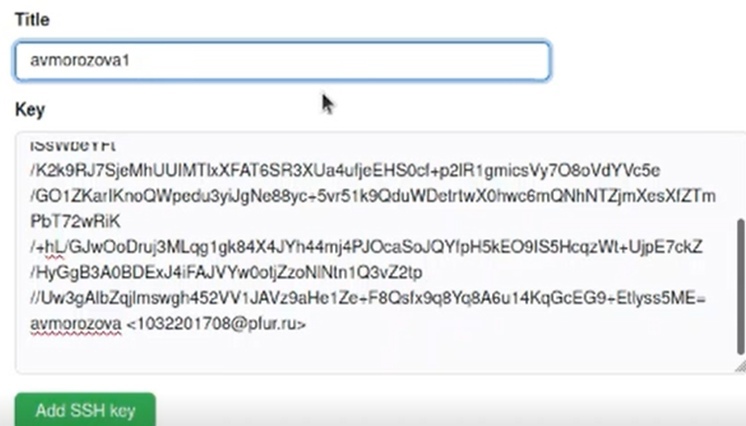
Figure 1: Учетная запись

* Настраиваю систему контроля версий git. Синхранизирую учётную запись github с компьютером: (рис. -fig. 2)

Figure 2: Настройка git

Figure 2: Настройка git

* Создаю новый ключ на github (команда ssh-keygen -C “avmorozova [1032201708@pfur.ru](mailto:1032201708@pfur.ru)”) и привязываю его к копьютеру через консоль. (рис. -fig. **¿fig:003?**) (рис. -fig. **¿fig:004?**) (рис. -fig. **¿fig:005?**) (рис. -fig. **¿fig:006?**)

1. **Подключение репозитория к github**

* В githup захожу в «repositories» и создаю новый репозиторий (имя «lab\_OS», заголовок для файла README). Копируем в консоль ссылку на репозиторий. (рис. -fig. 3)

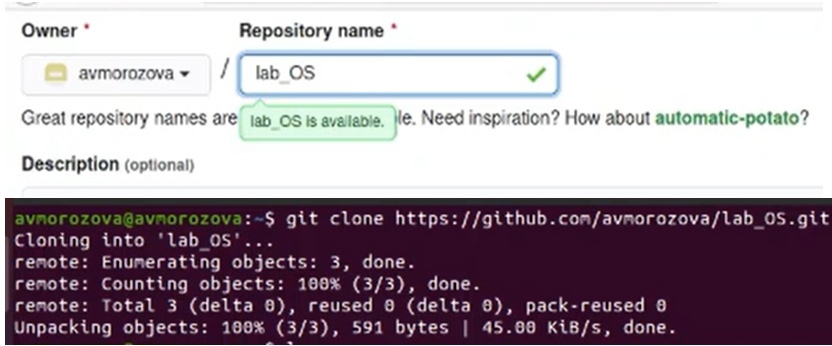


Figure 3: Новый репозиторий

* Работаю с каталогом и папками через консоль. Перед тем, как создавать файлы, захожу в репозиорий: (рис. -fig. 4)

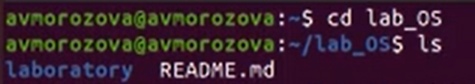


Figure 4: Работа с каталогом

* Создаю файлы: (рис. -fig. 5)

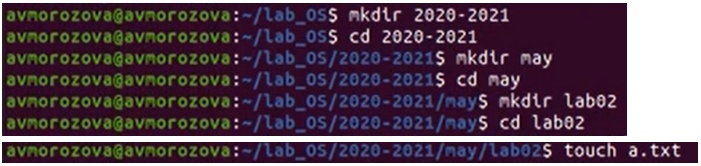


Figure 5: Каталоги

* 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. 6)

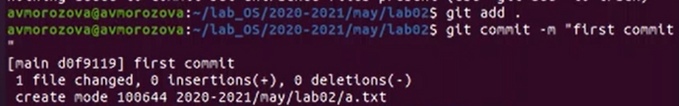


Figure 6: Первый коммит

* Сохраняю первый коммит (git push): (рис. -fig. 7)

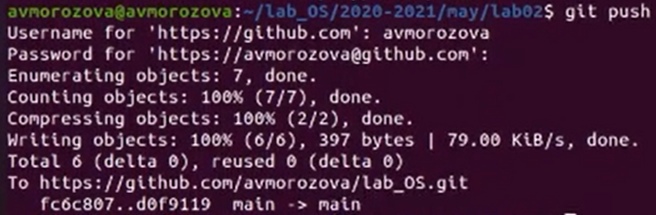


Figure 7: Сохранение первого коммита

1. **Первичная конфигурация**

* Добавляю файл лицензии: (рис. -fig. 8)



Figure 8: Файл лицензии

* Добавляю шаблон игнорируемых файлов. Получаю список имеющихся шаблонов (на скрине представлены не все шаблоны): (рис. -fig. 9)

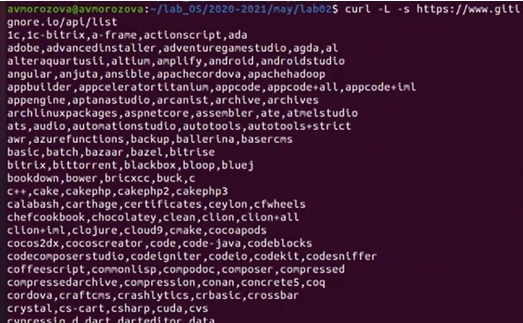
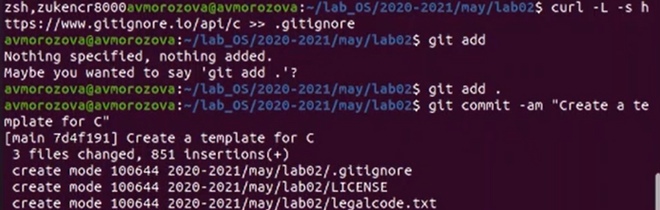
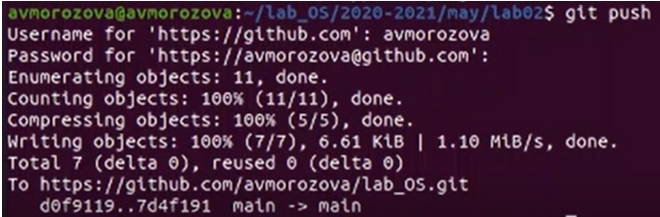
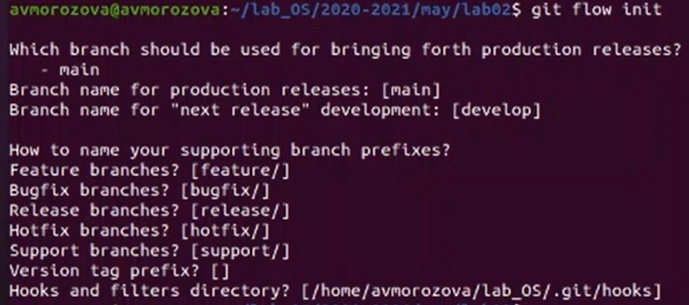
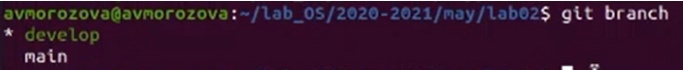
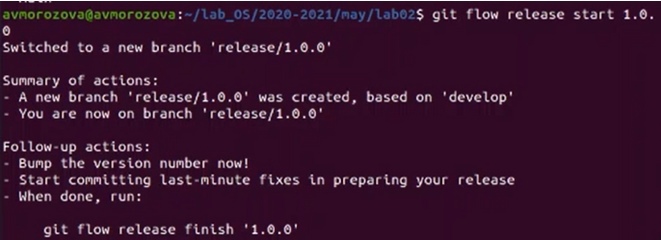
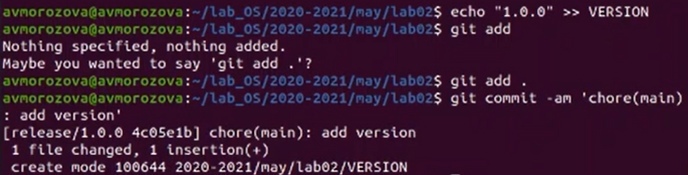
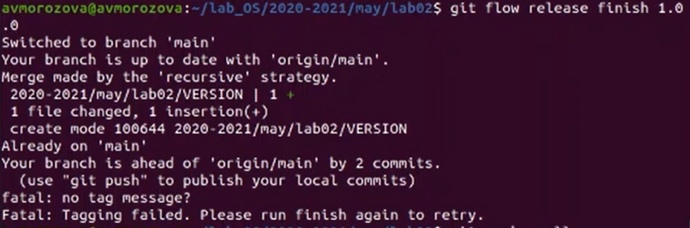
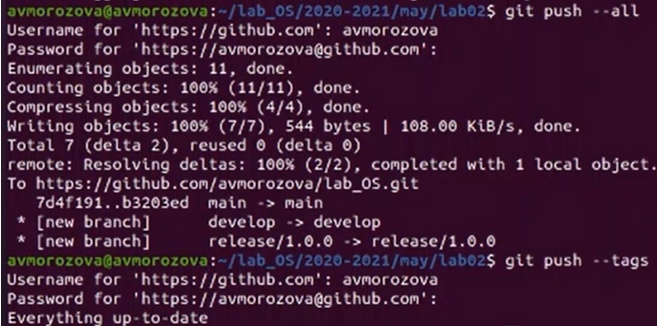
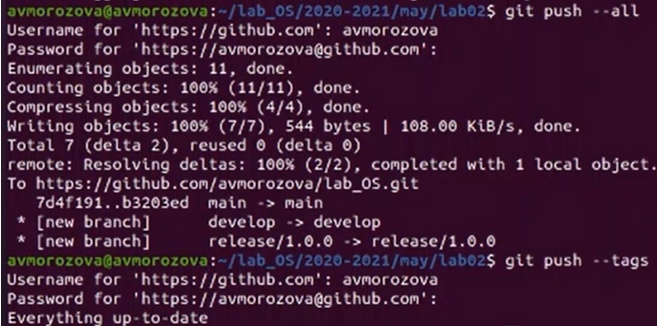


Figure 9: Шаблоны

* Скачиваю шаблон, например, для C. Также добавляю новые файлы и выполняю коммит: (рис. -fig. **¿fig:014?**) 
* Отправляю на github (git push): (рис. -fig. **¿fig:015?**) 

1. **Конфигурация git-flow**

* Инициализирую git-flow, используя команду git flow init -f (префикс для ярлыков установлен в v): (рис. -fig. **¿fig:016?**) 
* Проверяю, что нахожусь на ветке develop (git branch): (рис. -fig. **¿fig:017?**) 
* Создаю релиз с версией 1.0.0: (рис. -fig. **¿fig:018?**) 
* Записываю версию и добавляю в индекс: (рис. -fig. **¿fig:019?**) 
* Заливаю релизную ветку в основную ветку: (рис. -fig. **¿fig:020?**) 
* Отправляю данные на github: (рис. -fig. **¿fig:021?**) 
* Отправляю данные на github: (рис. -fig. **¿fig:021?**) 
* 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. **¿fig:022?**) 
* 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. **¿fig:022?**)(рис. -fig. **¿fig:023?**)  

# Выводы

В ходе выполнения лабораторной работы я научилась оформлять отчёты с помощью легковесного языка разметки Markdown.