Oтчёт по лабораторной работе 4

Операционные системы

Кижваткина Анна Юрьевна

Содержание

# 1 Цель работы

Получение практических навыков правильной работы с репозиторием git.

# 2 Выполнение лабораторной работы

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 1).

Заходим в виртуальную машину и открываем консоль с помощью комбинации Win+Enter. (рис. 1)

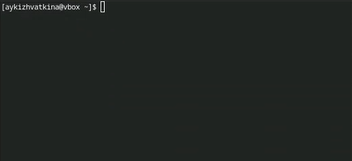


Рис. 1: Консоль

Устанавливаем git-flow. (рис. 2)

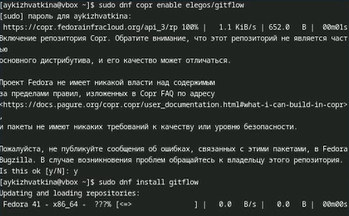


Рис. 2: Установка git-flow

Устанавливаем Node.js. (рис. 3 рис. 4)



Рис. 3: Установка Node.js

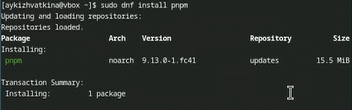


Рис. 4: Установка Node.js

Для работы с Node.js добавим каталог с исполняемыми файлами, устанавливаем yarn, в переменную path. (рис. 5 рис. 6)

Настройка Node.js

Рис. 5: Настройка Node.js

Настройка Node.js

Рис. 6: Настройка Node.js

Устанавливаем программу для помощи в форматировании коммитов. (рис. 7)

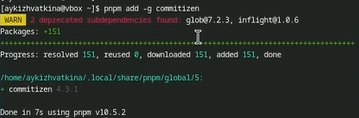


Рис. 7: Установка программы

Устанавливаем Standard-changelog для помощи в создании логов. (рис. 8)

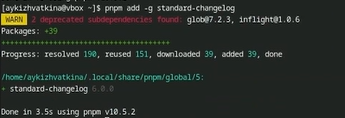


Рис. 8: Установка программы

Создаем репозиторий на GitHub. Называем его git-extended. (рис. 9)



Рис. 9: Создание репозитория

Делаем первый коммит и выкладываем на github. (рис. 10)

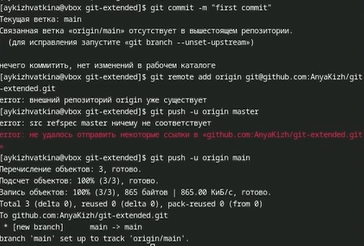


Рис. 10: Первый коммит

Делаем конфигурацию для пакетов Node.js. Необходимо заполнить несколько параметров пакета. (рис. 11)



Рис. 11: Конфигурация Node.js

Сконфигурируем формат коммитов. Для этого добавим в файл package.json команду для формирования коммитов. (рис. 12)

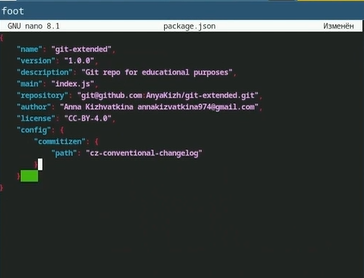


Рис. 12: Добавление команды для формирования коммитов

Добавим новые файлы. (рис. 13)

Добавление новых файлов

Рис. 13: Добавление новых файлов

Выполняем коммит. (рис. 14)

Выполнение коммита

Рис. 14: Выполнение коммита

Отправляем на github. (рис. 15)

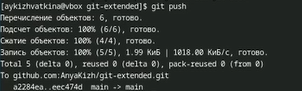


Рис. 15: Отправка на github

Переходим к конфигурации git-flow. Инициализируем git-flow. (рис. 16)

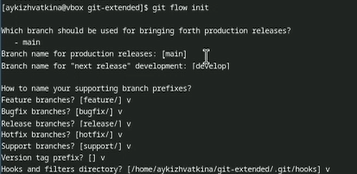


Рис. 16: Инициализация git-flow

Префикс для ярлыков установим в v.

Проверяем, что мы на ветке develop. (рис. 17)

Проверка ветки

Рис. 17: Проверка ветки

Загружаем весь репозиторий в хранилище. (рис. 18)

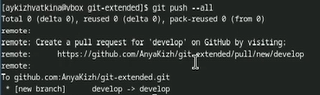


Рис. 18: Загрузка репозитория в хранилище

Устанавливаем внешнюю ветку как вышестоящую для этой ветки. (рис. 19)

Установка внешней ветки

Рис. 19: Установка внешней ветки

Создадим релиз с версией 1.0.0. (рис. 20)

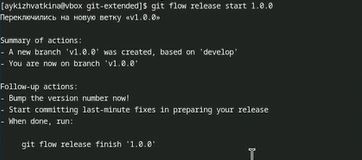


Рис. 20: Создание релиза

Создадим журнал изменений. (рис. 21)

Создание журнала изменений

Рис. 21: Создание журнала изменений

Добавим журнал изменений в индекс. (рис. 22)

Добавление журнала изменений

Рис. 22: Добавление журнала изменений

Зальём релизную ветку в основную ветку. (рис. 23)

Добавление релизной ветки

Рис. 23: Добавление релизной ветки

Отправим данные на GitHub. (рис. 24)

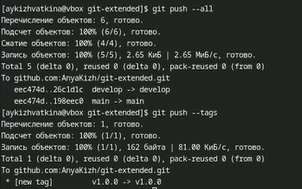


Рис. 24: Отправка данных

Создадим релиз на GitHub. Для этого используем утилиты работы с Github. (рис. 25)

Создание релиза

Рис. 25: Создание релиза

Создадим ветку для новой функциональности. Объединим созданную ветку с develop. (рис. 26)



Рис. 26: Создание новой ветки и объединение со старой

Перейдем к созданию релиза git-flow. Создадим релиз с версией 1.2.3. (рис. 27)

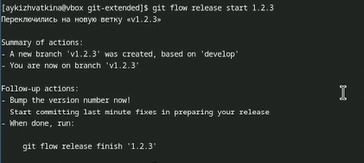


Рис. 27: Создание релиза

Обновим номер версии в файле package.json. Установим её в 1.2.3. (рис. 28)

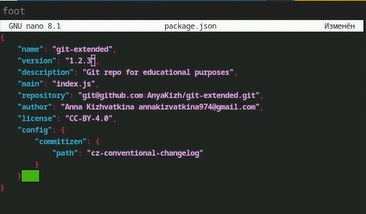


Рис. 28: Обновление номера версии

Создадим журнал изменений. (рис. 29)

Создание журнала изменений

Рис. 29: Создание журнала изменений

Добавим журнал изменений в индекс. (рис. 30)

Добавление журнала изменений

Рис. 30: Добавление журнала изменений

Зальём релизную ветку в основную ветку. (рис. 31)

Добавление релизной ветки в основную

Рис. 31: Добавление релизной ветки в основную

Отправляем данные на GitHub. (рис. 32)

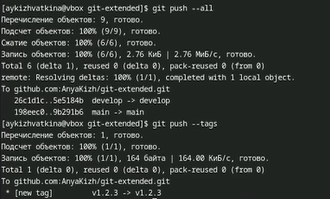


Рис. 32: Отправка данных на GitHub

Создадим релиз на GitHub с комментарием из журнала изменений. (рис. 33)

Создание релиза

Рис. 33: Создание релиза

# 3 Выводы

В ходе лабораторной работы я получила навыки правильной работы с репозиторием git.