Лабораторная работа № 4

Отчет

Казначеев Сергей Ильич

Содержание

# 1 Цель работы

Получение навыков правильной работы с репозиториями git.

# 2 Задание

Выполнить работу для тестового репозитория. Преобразовать рабочий репозиторий в репозиторий с git-flow и conventional commits.

# 3 Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы.

Например, в табл. 1 приведено краткое описание стандартных каталогов Unix.

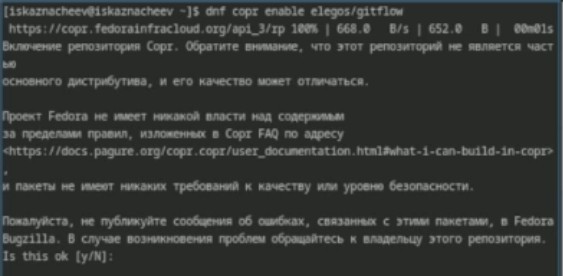
Таблица 1: Описание некоторых каталогов файловой системы GNU Linux

| Имя каталога | Описание каталога |
| --- | --- |
| / | Корневая директория, содержащая всю файловую |
| /bin | Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям |
| /etc | Общесистемные конфигурационные файлы и файлы конфигурации установленных программ |
| /home | Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя |
| /media | Точки монтирования для сменных носителей |
| /root | Домашняя директория пользователя root |
| /tmp | Временные файлы |
| /usr | Вторичная иерархия для данных пользователя |

Более подробно про Unix см. в [1–4].

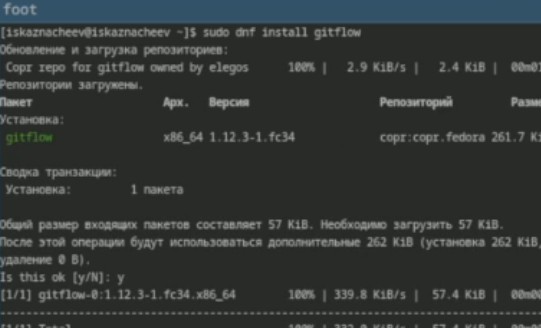
# 4 Выполнение лабораторной работы

Для начала подключим репозиторий из которого можно скачать gitflow



sc1

После установим gitflow



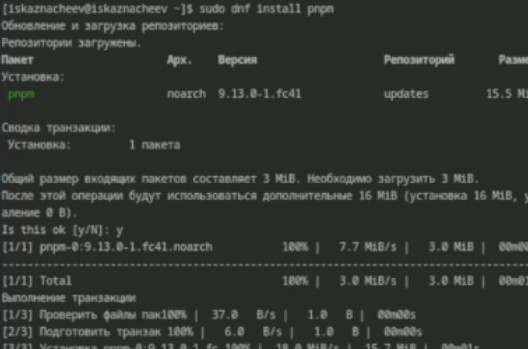
sc2

Теперь установим Nodejs



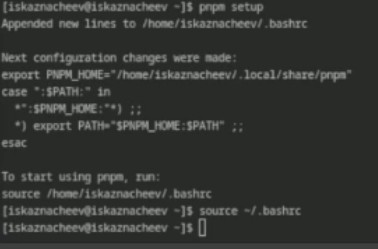
sc3

Установим pnpm



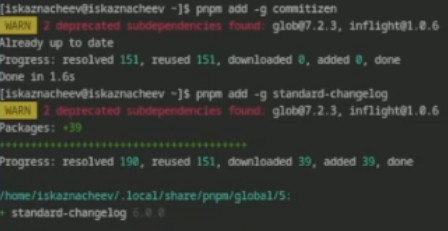
sc4

Запустим pnpm



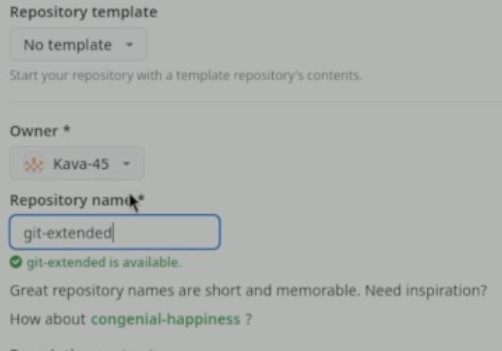
sc5

Теперь установим с помощью него Commitizien



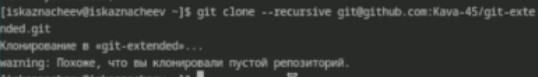
sc6

Создаем тестовый репозиторий git-extended



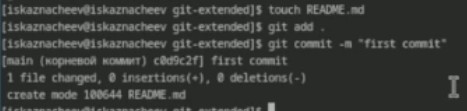
sc7

И клонируем его себе на компьютер



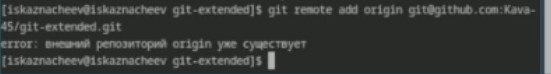
sc8

Создадим какой нибудь файл и проиндексируем его с помощью git add сделав при этом commit



sc9

Теперь добавим ветку



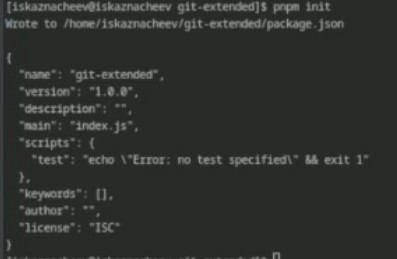
sc10

Теперь обратно запушим на github

sc11

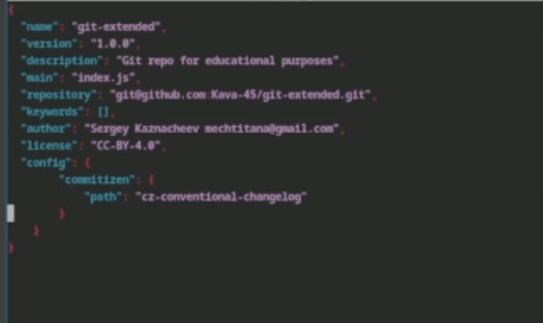
sc11

Теперь проинициализируем pnpm



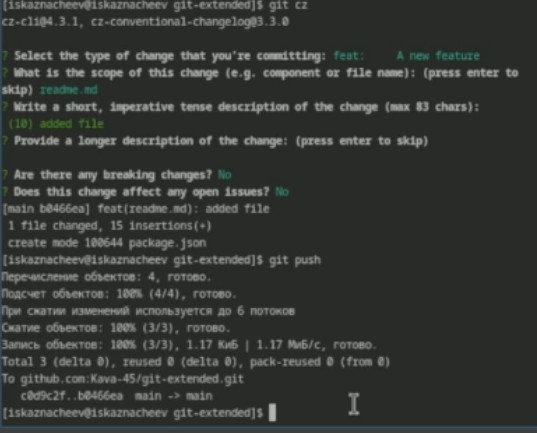
sc12

После инициальзации создается файл packege.json который меняем следующим образом



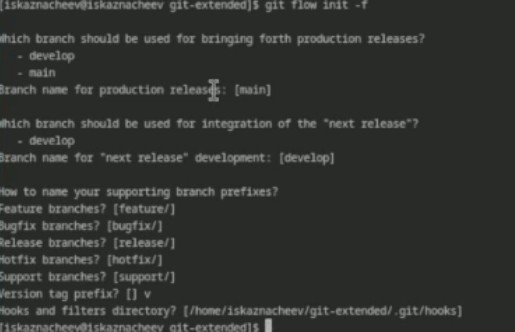
sc13

Делаем коммит с помощью cz



sc14

Проинициализируем gitflow. Укажем название веток и префикс для версий



sc15

Выведем список веток и убедимся, что мы находимся в develop, и запушим изменения на сервер

sc16

sc16

Переключимся на ветку develop после чего создадим ветку релиза где создадим changelog

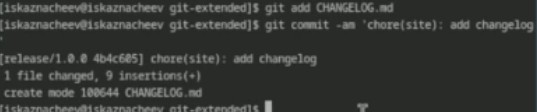
sc17

sc17

sc18

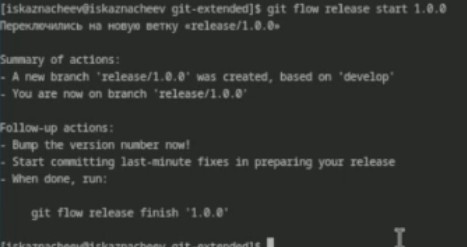
sc18

Проиндексируем changelog и сделаем коммит



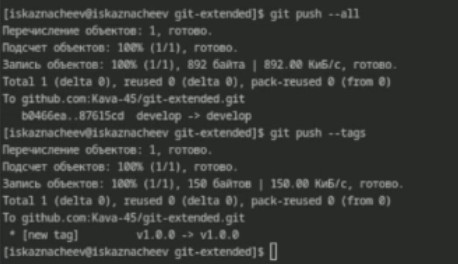
sc19

Теперь сольем ветку relaese с веткой changelog



sc21

Загружаем данные на github



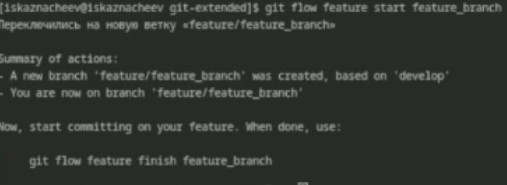
sc22

Создаем релиз из changelog’a

sc23

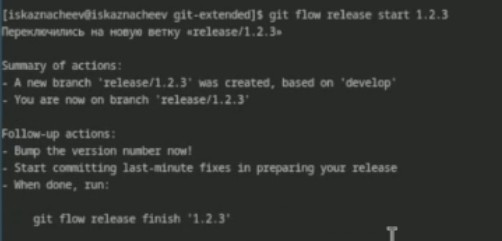
sc23

Создаем ветку feature и сразу сливаем ее с develop



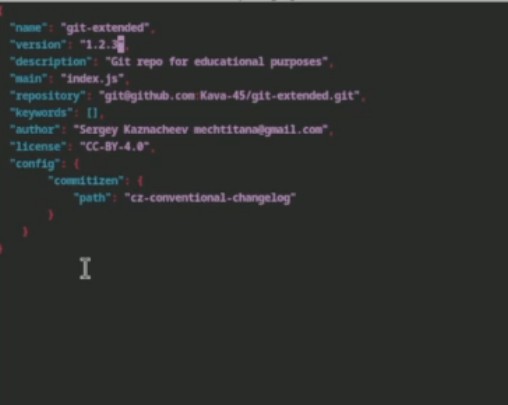
sc24

Создаем ветку релиза



sc25

Меняем версию в packege.json



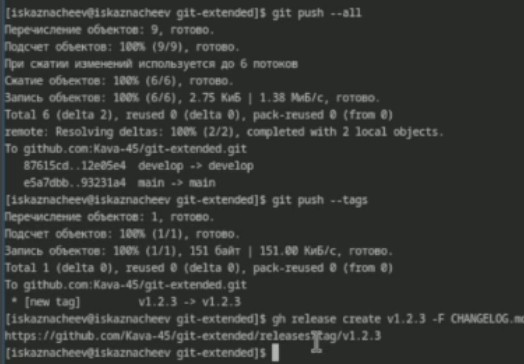
sc26

Создаем ветку журнал изменений, проиндексируем его и сольем ветку с ним в ветку develop



sc27

Загрузим изменения на гитхаб и создадим релиз



sc28

# 5 Вывод

Я получил правильные навыки работы, с репозиториями git

# Список литературы

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.

2. Robbins A. Bash Pocket Reference. O’Reilly Media, 2016. 156 с.

3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.

4. Newham C. [Learning the bash Shell: Unix Shell Programming](http://www.amazon.com/Learning-bash-Shell-Programming-Nutshell/dp/0596009658). O’Reilly Media, 2005. 354 с.