Отчёт по лабораторной работе №3

Простейший вариант

Тимур Ринатович Каримов

Содержание

1 Цель работы **5**

2 Задание **6**

3 Теоретическое введение **7**

4 Выполнение лабораторной работы **8**

5 Выводы **14**

Список литературы **15**

Список иллюстраций

4.1 Установка gitflow 8
4.2 Установка nodejs 8
4.3 Установка pnpm 8
4.4 Запуск pnpm 8
4.5 Выполнение команды source 8
4.6 Выполнение команды commitizen 9
4.7 Выполнение команды standard-changelog 9
4.8 Пустой репозиторий 9
4.9 Коммит и отправка на GitHub 10
4.10 Изменения данных в package.json 10
4.11 Добавление новых файлов и выполнение коммит 10
4.12 Отправка файлов 11
4.13 Инициализация git-flow 11
4.14 Проверка ветки 11
4.15 Отправка репозитория 11
4.16 Создание релиза 1.0.0 11
4.17 Создание журнала изменений 11
4.18 Добавление журнала изменений в индекс 12
4.19 Отправка релизной ветки в основную 12
4.20 Отправка данных на сервер 12
4.21 Отправка данных на сервер 12
4.22 Создание релиза на GitHub 12
4.23 Создание ветки для новой функциональности 12
4.24 Создание релиза git-flow 13
4.25 Обновление номера версии в файле package.json 13
4.26 Создание журнала изменений 13
4.27 Создание релиз на github 13

Список таблиц

3.1 Описание некоторых каталогов файловой системы GNU Linux . . . 7

1 Цель работы

Освоение навыков правильной работы с репозиториями git.

2 Задание

Установка git-flow Установка Node.js Настройка Node.js Общепринятые коммиты Создание репозитория git Работа с репозиторием git

3 Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы.

Например, в табл. 3.1 приведено краткое описание стандартных каталогов Unix.

Таблица 3.1: 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

Установим *gitflow* (рис. 4.1)

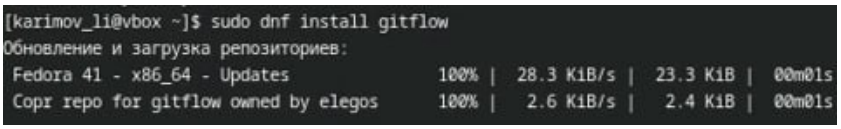

Рис. 4.1: Установка gitflow

Установим *nodejs* (рис. 4.2).

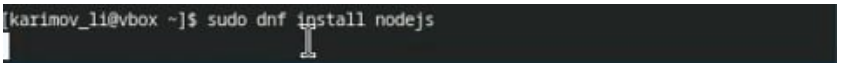

Рис. 4.2: Установка nodejs

Установим *pnpm* (рис. 4.3).

Рис. 4.3: Установка pnpm

Запустим *pnpm* (рис. 4.4).

Рис. 4.4: Запуск pnpm

Выполненим команду *source ~/.bashrc* (рис. 4.5).

Рис. 4.5: Выполнение команды source

Введем программу для помощи в форматировании коммитов (рис. 4.6).

Рис. 4.6: Выполнение команды commitizen

Введем программу для помощи в создании логов (рис. 4.7).

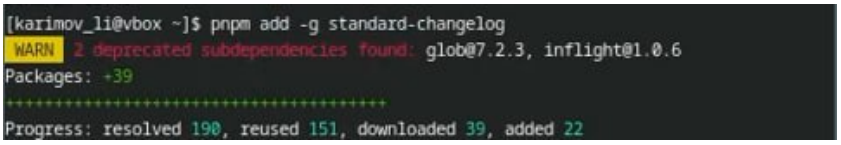

Рис. 4.7: Выполнение команды standard-changelog

Создадим пустой репозиторий *git-extended* (рис. 4.8).

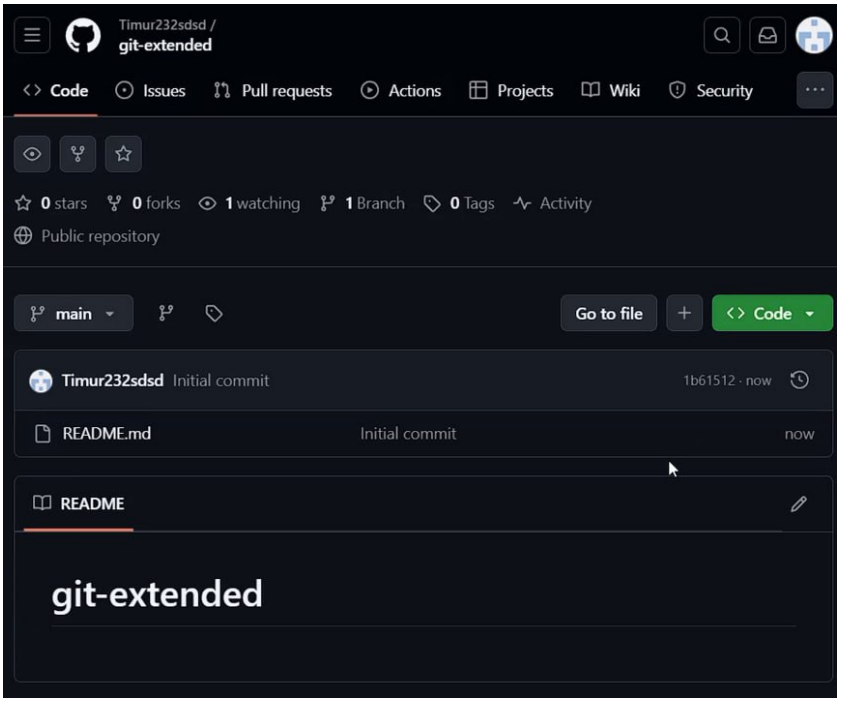

Рис. 4.8: Пустой репозиторий

Сделаем первый коммит и выложим на *GitHub* (рис. 4.9).

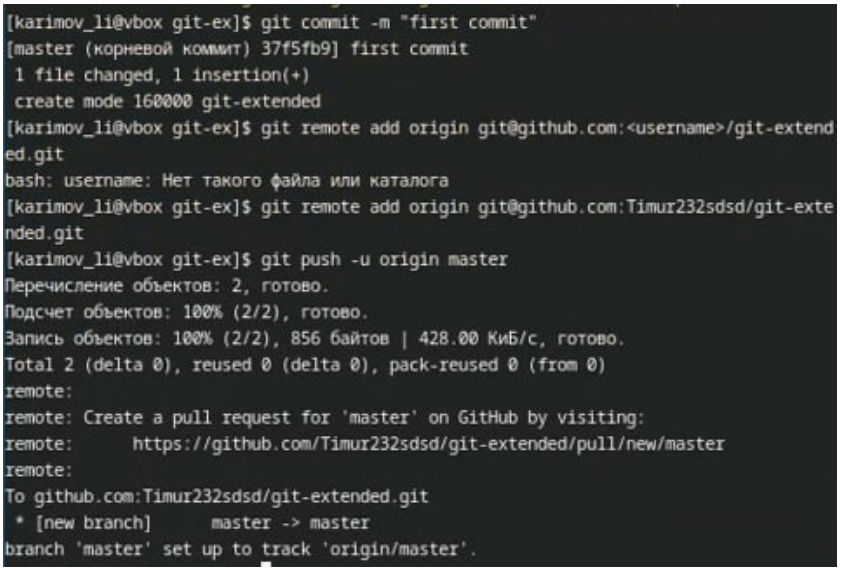

Рис. 4.9: Коммит и отправка на GitHub

Добавим в файл package.json команду для формирования коммитов (рис. 4.10).

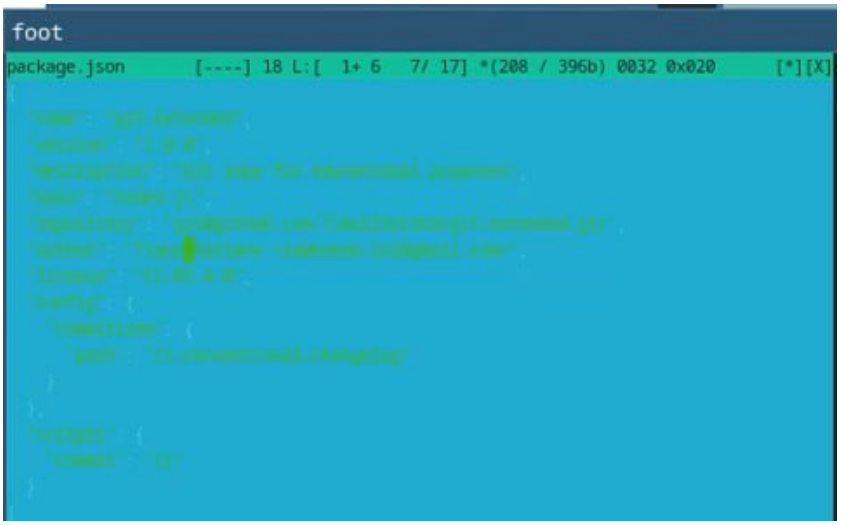

Рис. 4.10: Изменения данных в package.json

Добавим новые файлы и выполним коммит (рис. 4.11).

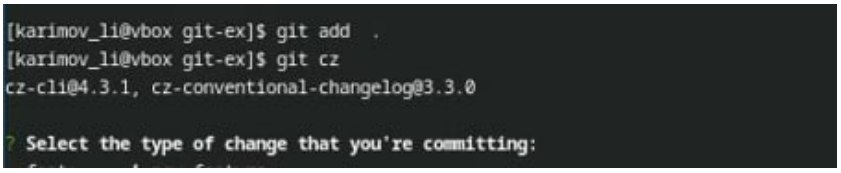

Рис. 4.11: Добавление новых файлов и выполнение коммит

Отправим файлы на GitHub (рис. 4.12).

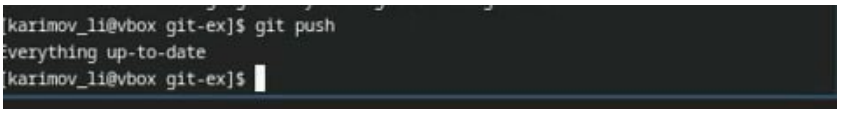

Рис. 4.12: Отправка файлов

Инициализируем git-flow (рис. 4.13).

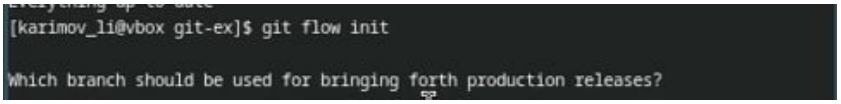

Рис. 4.13: Инициализация git-flow

Проверим, что мы на ветке *develop* (рис. 4.14).

Рис. 4.14: Проверка ветки

Загрузим весь репозиторий в хранилище (рис. 4.15).

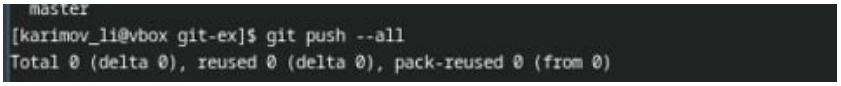

Рис. 4.15: Отправка репозитория

Создадим релиз с версией 1.0.0 (рис. 4.16).

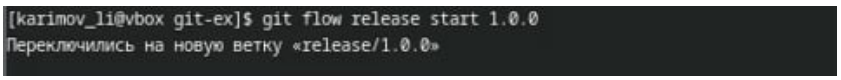

Рис. 4.16: Создание релиза 1.0.0

Создадим журнал изменений (рис. 4.17).

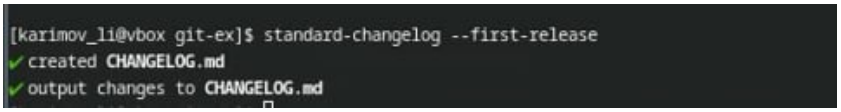

Рис. 4.17: Создание журнала изменений

Добавим журнал изменений в индекс (рис. 4.18).

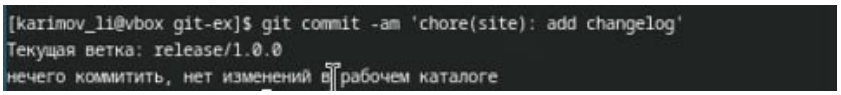

Рис. 4.18: Добавление журнала изменений в индекс

Зальём релизную ветку в основную ветку (рис. 4.19).

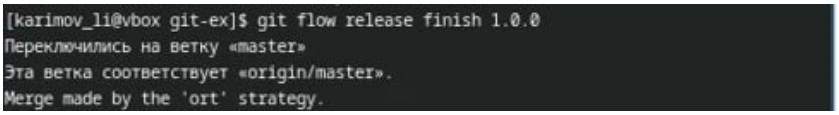

Рис. 4.19: Отправка релизной ветки в основную

Отправим данные на *GitHub* (рис. 4.20) и (рис. 4.21).

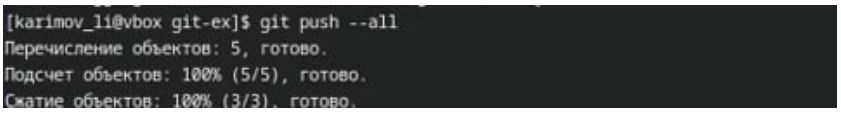

Рис. 4.20: Отправка данных на сервер

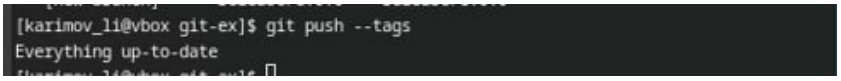

Рис. 4.21: Отправка данных на сервер

Создадим релиз на github. Для этого будем использовать утилиты работы с github (рис. 4.22).

Рис. 4.22: Создание релиза на GitHub

Создадим ветку для новой функциональности (рис. 4.23).

Рис. 4.23: Создание ветки для новой функциональности

Созданим релиз git-flow (рис. 4.24).

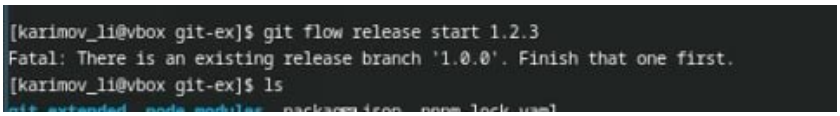

Рис. 4.24: Создание релиза git-flow

Обновим номер версии в файле package.json (рис. 4.25).

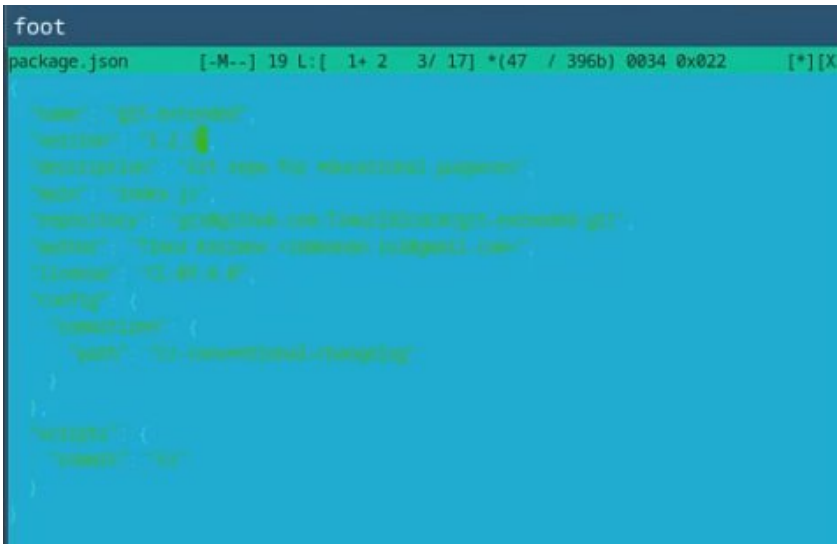

Рис. 4.25: Обновление номера версии в файле package.json

Создадим журнал изменений (рис. 4.26).

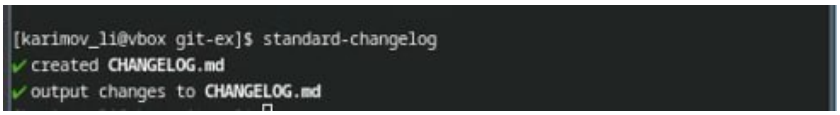

Рис. 4.26: Создание журнала изменений

Добавим журнал изменений в индекс и создадим релиз на github с комментарием из журнала изменений (рис. 4.27).

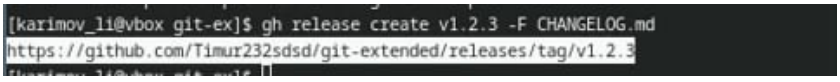

Рис. 4.27: Создание релиз на github

5 Выводы

Работа продемонстрировала эффективность использования современных инструментов разработки, таких как git-flow, commitizen, standard-changelog и pnpm, для автоматизации и стандартизации процессов разработки. Все задачи были выполнены успешно, репозиторий настроен для дальнейшей работы, а процесс создания коммитов, управления ветками и выпуска релизов стал более структурированным и удобным.

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. O’Reilly Media, 2005. 354 с.