Отчёт по лабораторной работе №3

Простейший вариант

Тимур Ринатович Каримов

Содержание

# 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. 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

Установим *gitflow* (рис. 1)

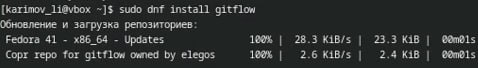


Рис. 1: Установка gitflow

Установим *nodejs* (рис. 2).

Установка nodejs

Рис. 2: Установка nodejs

Установим *pnpm* (рис. 3).

Установка pnpm

Рис. 3: Установка pnpm

Запустим *pnpm* (рис. 4).

Запуск pnpm

Рис. 4: Запуск pnpm

Выполненим команду *source ~/.bashrc* (рис. 5).

Выполнение команды source

Рис. 5: Выполнение команды source

Введем программу для помощи в форматировании коммитов (рис. 6).

Выполнение команды commitizen

Рис. 6: Выполнение команды commitizen

Введем программу для помощи в создании логов (рис. 7).

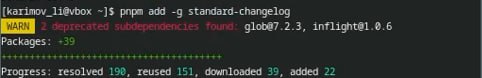


Рис. 7: Выполнение команды standard-changelog

Создадим пустой репозиторий *git-extended* (рис. 8).

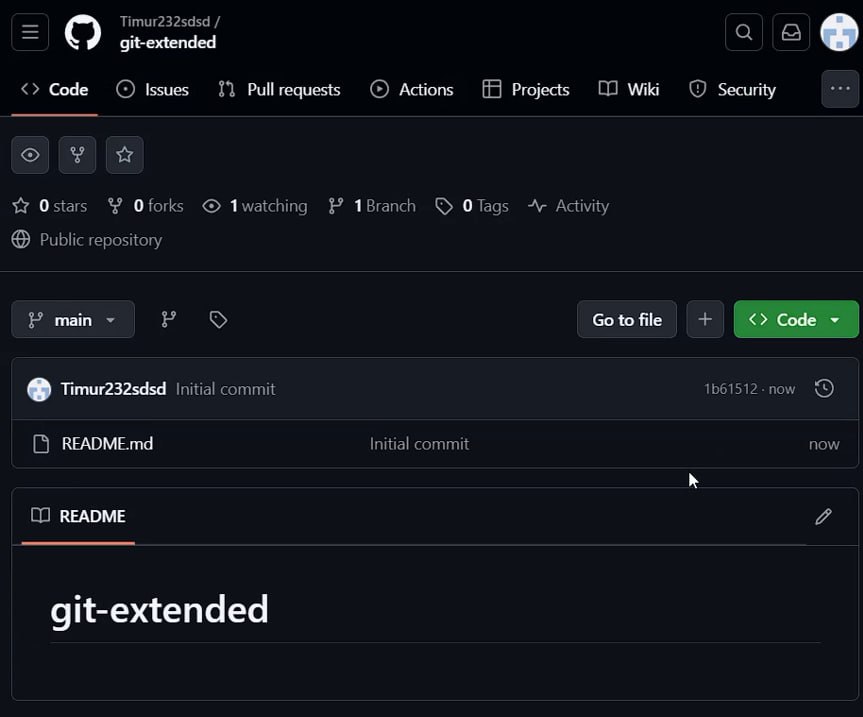


Рис. 8: Пустой репозиторий

Сделаем первый коммит и выложим на *GitHub* (рис. 9).

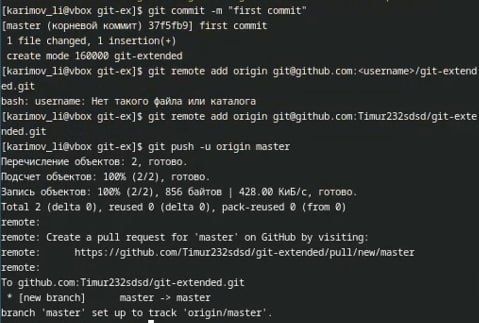


Рис. 9: Коммит и отправка на GitHub

Добавим в файл package.json команду для формирования коммитов (рис. 10).

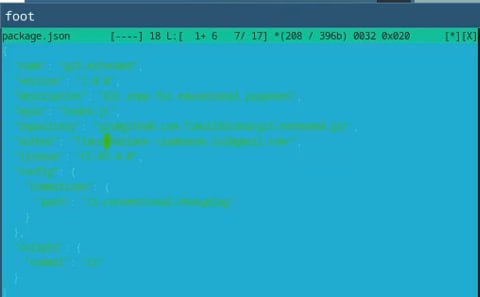


Рис. 10: Изменения данных в package.json

Добавим новые файлы и выполним коммит (рис. 11).

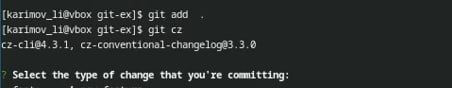


Рис. 11: Добавление новых файлов и выполнение коммит

Отправим файлы на GitHub (рис. 12).

Отправка файлов

Рис. 12: Отправка файлов

Инициализируем git-flow (рис. 13).

Инициализация git-flow

Рис. 13: Инициализация git-flow

Проверим, что мы на ветке *develop* (рис. 14).

Проверка ветки

Рис. 14: Проверка ветки

Загрузим весь репозиторий в хранилище (рис. 15).

Отправка репозитория

Рис. 15: Отправка репозитория

Создадим релиз с версией 1.0.0 (рис. 16).

Создание релиза 1.0.0

Рис. 16: Создание релиза 1.0.0

Создадим журнал изменений (рис. 17).

Создание журнала изменений

Рис. 17: Создание журнала изменений

Добавим журнал изменений в индекс (рис. 18).

Добавление журнала изменений в индекс

Рис. 18: Добавление журнала изменений в индекс

Зальём релизную ветку в основную ветку (рис. 19).

Отправка релизной ветки в основную

Рис. 19: Отправка релизной ветки в основную

Отправим данные на *GitHub* (рис. 20) и (рис. 21).

Отправка данных на сервер

Рис. 20: Отправка данных на сервер

Отправка данных на сервер

Рис. 21: Отправка данных на сервер

Создадим релиз на github. Для этого будем использовать утилиты работы с github (рис. 22).

Создание релиза на GitHub

Рис. 22: Создание релиза на GitHub

Создадим ветку для новой функциональности (рис. 23).

Создание ветки для новой функциональности

Рис. 23: Создание ветки для новой функциональности

Созданим релиз git-flow (рис. 24).

Создание релиза git-flow

Рис. 24: Создание релиза git-flow

Обновим номер версии в файле package.json (рис. 25).

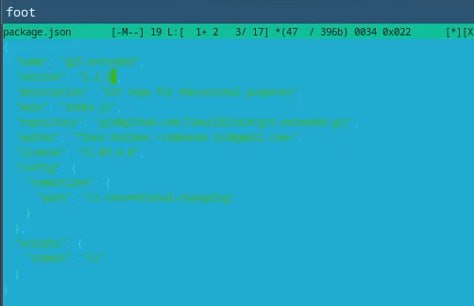


Рис. 25: Обновление номера версии в файле package.json

Создадим журнал изменений (рис. 26).

Создание журнала изменений

Рис. 26: Создание журнала изменений

Добавим журнал изменений в индекс и создадим релиз на github с комментарием из журнала изменений (рис. 27).

Создание релиз на github

Рис. 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](http://www.amazon.com/Learning-bash-Shell-Programming-Nutshell/dp/0596009658). O’Reilly Media, 2005. 354 с.