

Лабораторная работа №4

Операционные системы

Верниковская Е. А., НПИбд-01-23

27 февраля 2024

Российский университет дружбы народов, Москва, Россия

Вводная часть

Получить навыки правильной работы с репозиториями git.

1. Установить программное обеспечение.
2. Выполнить работу для тестового репозитория.
3. Преобразовать рабочий репозиторий в репозиторий с git-flow и conventional commits.

Выполнение лабораторной работы

Переключаемся на супер-пользователя и устанавливаем git-flow введя 2 команды: - *dnf copr enable elegos/gitflow* - *dnf install gitflow* (рис. 1), (рис. 2)

```
[eavernikovskaya@eavernikovskaya ~]$ sudo -i
[sudo] пароль для eavernikovskaya:
[root@eavernikovskaya ~]# dnf copr enable elegos/gitflow
Включение репозитория Copr. Обратите внимание, что этот репозиторий
не является частью основного дистрибутива, и качество может отличаться.

Проект Fedora не имеет какого-либо влияния на содержимое этого
репозитория за рамками правил, описанных в Вопросах и Ответах Copr в
<https://docs.pagure.org/copr.copr/user\_documentation.html#what-i-can-build-in-copr>,
а качество и безопасность пакетов не поддерживаются на каком-либо уровне.

Не отправляйте сообщения об ошибках этих пакетов в Fedora
Bugzilla. В случае возникновения проблем обращайтесь к владельцу этого репозитория.

Do you really want to enable copr.fedorainfracloud.org/elegos/gitflow? [y/N]: y
Репозиторий успешно подключен.
[root@eavernikovskaya ~]#
```

Рис. 1: Установка git-flow (1)

```
[root@eavernikovskaya ~]# dnf install gitflow
Copr repo for gitflow owned by elegos                  1.3 kB/s | 1.5 kB   00:01
Fedora 39 - x86_64                                     29 kB/s | 22 kB   00:00
Fedora 39 - x86_64 - Updates                          96 kB/s | 21 kB   00:00
Fedora 39 - x86_64 - Updates                         1.7 MB/s | 5.2 MB   00:03
Зависимости разрешены.
=====
Пакет      Архитектура  Версия      Репозиторий      Размер
=====
Установка:
gitflow    x86_64       1.12.3-1.fc34  copr:copr.fedorainfracloud.org:elegos:gitflow  57 k
Результат транзакции
=====
Установка 1 Пакет

Объем загрузки: 57 k
Объем изменений: 262 k
Продолжить? [д/н]:
```

Рис. 2: Установка git-flow (2)

Далее устанавливаем Node.js введя 2 команды: - *dnf install nodejs* - *dnf install npm* (рис. 3), (рис. 4)

```
[root@eavernikovskaya ~]# dnf install nodejs
Последняя проверка окончания срока действия метаданных: 0:02:54 назад, Вс 03 мар 2024 19:02:54.
Зависимости разрешены.
=====
Пакет      Архитектура  Версия      Репозиторий  Размер
-----
Установка:
nodejs     x86_64       1:20.10.0-3.fc39  updates      48 k
Установка зависимостей:
nodejs-libs x86_64       1:20.10.0-3.fc39  updates      15 М
Установка слабых зависимостей:
nodejs-docs noarch       1:20.10.0-3.fc39  updates      8.1 М
nodejs-full-118n x86_64       1:20.10.0-3.fc39  updates      8.5 М
nodejs-npm  x86_64       1:10.2.3-1.20.10.0.3.fc39 updates      2.2 М

Результат транзакции
=====
Установка 5 Пакетов
Объем загрузки: 34 М
```

Рис. 3: Установка Node.js (1)

Установка Node.js

```
[root@eavernikovskaya ~]# sudo dnf install npm
Последняя проверка окончания срока действия метаданных: 0:07:24 назад, Вс 03 мар 2024 19:02:54.
Зависимости разрешены.
=====
Пакет                Архитектура          Версия                Репозиторий          Размер
=====
Установка:
  npm                noarch                8.12.0-1.fc39         updates                2.6 М
=====
Результат транзакции
=====
Установка 1 Пакет

Объем загрузки: 2.6 М
Объем изменений: 13 М
Продолжить? [д/н]: y
Загрузка пакетов:
  npm-8.12.0-1.fc39.noarch.rpm                                4.7 MB/s | 2.6 MB  00:00
-----
Общий размер                                              1.9 MB/s | 2.6 MB  00:01
-----
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверка транзакции
Тест транзакции проведен успешно.
Выполнение транзакции
Подготовка      :
Установка       : npm-8.12.0-1.fc39.noarch                1/1
Запуск скрипта  : npm-8.12.0-1.fc39.noarch                1/5
Проверка        : npm-8.12.0-1.fc39.noarch                1/1
-----
Установлен:
  npm-8.12.0-1.fc39.noarch

Выполнено!
[root@eavernikovskaya ~]#
```

Рис. 4: Установка Node.js (2)

Запускаем `pnpm` с помощью *pnpm setup*, далее выполняем команду *source ~/.bashrc* (рис. 5), (рис. 6)

```
[root@eavernikovskaya ~]# pnpm setup
Appended new lines to /root/.bashrc

Next configuration changes were made:
export PNPM_HOME="/root/.local/share/pnpm"
case ":$PATH:" in
  *"$PNPM_HOME:") ;;
  *) export PATH="$PNPM_HOME:$PATH" ;;
esac

To start using pnpm, run:
source /root/.bashrc
[root@eavernikovskaya ~]#
```

Рис. 5: Настройка Node.js (1)

```
[root@eavernikovskaya ~]# source ~/.bashrc  
[root@eavernikovskaya ~]#
```

Рис. 6: Настройка Node.js (2)

Общепринятые коммиты

Устанавливаем пакет commitizen с помощью команды `pnpm add -g commitizen`.
Данная программа используется для помощи в форматировании коммитов
(рис. 7)

```
[root@eavernikovskaya ~]# pnpm add -g commitizen

  Update available! 8.12.0 -> 8.15.4.
  Changelog: https://github.com/pnpm/pnpm/releases/tag/v8.15.4
  Run "pnpm add -g pnpm" to update.

  Follow @pnpmjs for updates: https://twitter.com/pnpmjs

Packages: +152
.....
Downloading registry.npmjs.org/typescript/5.3.3: 5.76 MB/5.76 MB, done
Progress: resolved 152, reused 0, downloaded 152, added 152, done

/root/.local/share/pnpm/global/5:
+ commitizen 4.3.0

Done in 9.6s
[root@eavernikovskaya ~]#
```

Рис. 7: Commitizen

Общепринятые коммиты

Устанавливаем пакет `standard-changelog` с помощью команды `pnpm add -g standard-changelog`. Данная программа используется для помощи в создании логов (рис. 8)

```
[root@eavernikovskaya ~]# pnpm add -g standard-changelog
Packages: +56
+++++
Progress: resolved 208, reused 152, downloaded 56, added 56, done

/root/.local/share/pnpm/global/5:
+ standard-changelog 5.0.0

Done in 6.9s
[root@eavernikovskaya ~]#
```

Рис. 8: Standard-changelog

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

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)


Required fields are marked with an asterisk ().*

Repository template

No template ▾

Start your repository with a template repository's contents.

Owner * **Repository name ***

 Katerok27153 ▾ / git-extended

✔ git-extended is available.

Рис. 9: Создание репозитория на github

Далее клонируем созданный репозиторий, с помощью *git clone --recursive* (рис. 10)

```
[eavernikovskaya@eavernikovskaya ~]$ git clone --recursive https://github.com/Katerok27153/git-extended.git
Клонирование в «git-extended»...
warning: Похоже, что вы клонировали пустой репозиторий.
[eavernikovskaya@eavernikovskaya ~]$ cd git-extended/
[eavernikovskaya@eavernikovskaya git-extended]$
```

Рис. 10: Клонирование репозитория

Создаём пустой файл, чтобы активировать репозиторий, делаем первый коммит и выкладываем на github (рис. 11)

```
[eavernikovskaya@eavernikovskaya git-extended]$ touch README.md
[eavernikovskaya@eavernikovskaya git-extended]$ git add .
[eavernikovskaya@eavernikovskaya git-extended]$ git commit -m "first commit"
[main (корневой коммит) b81dad7] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
[eavernikovskaya@eavernikovskaya git-extended]$ git remote add new-origin https://github.com/Katerok27153/git-extended.git
```

Рис. 11: Сохранение изменений + первый коммит


```
[eavernikovskaya@eavernikovskaya git-extended]$ git push -u origin main
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 863 байта | 863.00 КиБ/с, готово.
Всего 3 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To https://github.com/Katerok27153/git-extended.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
[eavernikovskaya@eavernikovskaya git-extended]$
```

Рис. 12: Отправка файлов на github

Общепринятые коммиты

Выполняем конфигурацию для пакетов Node.js, с помощью команды *pnpm init* (рис. 13)

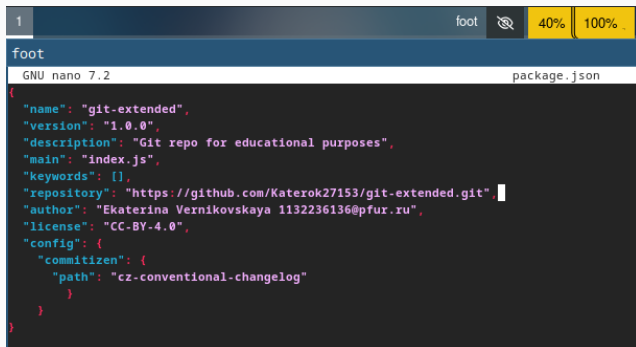
```
[eavernikovskaya@eavernikovskaya git-extended]$ pnpm init
Wrote to /home/eavernikovskaya/git-extended/package.json

{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
[eavernikovskaya@eavernikovskaya git-extended]$
```

Рис. 13: Конфигурация для пакетов Node.js

Общепринятые коммиты

Далее редактируем файл `package.json`, вставив в него команду для формирования коммитов и заполнив информацию о названии, лицензии и т.д. (рис. 14)

A screenshot of a terminal window showing the nano text editor editing the `package.json` file. The editor's title bar shows 'foot', a search icon, and progress indicators for 40% and 100%. The status bar at the top of the editor shows 'GNU nano 7.2' and 'package.json'. The code content is a JSON object with the following fields: 'name' (git-extended), 'version' (1.0.0), 'description' (Git repo for educational purposes), 'main' (index.js), 'keywords' (empty array), 'repository' (https://github.com/Katerok27153/git-extended.git), 'author' (Ekaterina Vernikovskaya 1132236136@pfur.ru), 'license' (CC-BY-4.0), and 'config' (an object containing 'commitizen' with a 'path' of 'cz-conventional-changelog').

```
1
foot 40% 100%
foot
GNU nano 7.2 package.json
{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "keywords": [],
  "repository": "https://github.com/Katerok27153/git-extended.git",
  "author": "Ekaterina Vernikovskaya 1132236136@pfur.ru",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 14: Отредактированный файл `package.json`

Общепринятые коммиты

После добавляем новые файлы, выполняем коммит и отправляем на github, с помощью *git add .*, *git cz* и *git push* (рис. 15), (рис. 16)

```
[eavernikovskaya@eavernikovskaya git-extended]$ git add .  
[eavernikovskaya@eavernikovskaya git-extended]$ git cz  
cz-cli@4.3.0, cz-conventional-changelog@3.3.0  
  
? Select the type of change that you're committing: feat:      A new feature  
? What is the scope of this change (e.g. component or file name): (press enter to skip) readme.md  
? Write a short, imperative tense description of the change (max 83 chars):  
  (8) add file  
? Provide a longer description of the change: (press enter to skip)  
  
? Are there any breaking changes? No  
? Does this change affect any open issues? No  
[main afa7beb] feat(readme.md): add file  
  1 file changed, 16 insertions(+)  
  create mode 100644 package.json  
[eavernikovskaya@eavernikovskaya git-extended]$
```

Рис. 15: Добавление файлов + коммит

```
[eavernikovskaya@eavernikovskaya git-extended]$ git push
Перечисление объектов: 4, готово.
Подсчет объектов: 100% (4/4), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 1.18 КиБ | 1.18 МиБ/с, готово.
Всего 3 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To https://github.com/Katerok27153/git-extended.git
   b81dad7..afa7beb  main -> main
[eavernikovskaya@eavernikovskaya git-extended]$
```

Рис. 16: Отправка на github

Общепринятые коммиты

Инициализируем git-flow введя *git flow init*. Префикс для ярлыков устанавливаем в *v* (рис. 17)

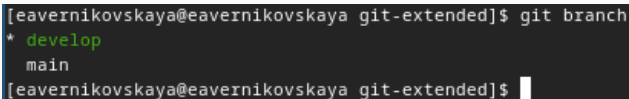
```
[eavernikovskaya@eavernikovskaya git-extended]$ git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? [] v
Hooks and filters directory? [/home/eavernikovskaya/git-extended/.git/hooks]
[eavernikovskaya@eavernikovskaya git-extended]$
```

Рис. 17: Инициализация git-flow

Проверяем, что мы находимся на ветке `develop` с помощью *git branch* (рис. 18)

A terminal window with a dark background. The prompt is [eavernikovskaya@eavernikovskaya git-extended]\$ and the command is git branch. The output shows two branches: * develop (highlighted in green) and main. The prompt returns to [eavernikovskaya@eavernikovskaya git-extended]\$ with a white cursor.

```
[eavernikovskaya@eavernikovskaya git-extended]$ git branch
* develop
  main
[eavernikovskaya@eavernikovskaya git-extended]$
```

Рис. 18: Ветка `develop`

Далее загружаем весь репозиторий в хранилище командой *git push --all* (рис. 19)

```
[eavernikovskaya@eavernikovskaya git-extended]$ git push --all
Всего 0 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/Katerok27153/git-extended/pull/new/develop
remote:
To https://github.com/Katerok27153/git-extended.git
 * [new branch]      develop -> develop
[eavernikovskaya@eavernikovskaya git-extended]$
```

Рис. 19: Загрузка всего репозитория в хранилище

После устанавливаем внешнюю ветку как вышестоящую для ветки develop (рис. 20)

```
[eavernikovskaya@eavernikovskaya git-extended]$ git branch --set-upstream-to=origin/develop develop  
branch 'develop' set up to track 'origin/develop'.  
[eavernikovskaya@eavernikovskaya git-extended]$
```

Рис. 20: Работа с ветками

Создаём релиз с версией 1.0.0 введя *git flow release start 1.0.0* (рис. 21)

```
[eavernikovskaya@eavernikovskaya git-extended]$ git flow release start 1.0.0
Переключились на новую ветку «release/1.0.0»

Summary of actions:
- A new branch 'release/1.0.0' was created, based on 'develop'
- You are now on branch 'release/1.0.0'

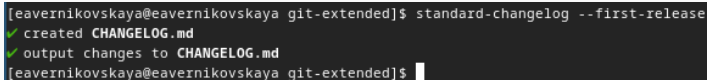
Follow-up actions:
- Bump the version number now!
- Start committing last-minute fixes in preparing your release
- When done, run:

    git flow release finish '1.0.0'

[eavernikovskaya@eavernikovskaya git-extended]$
```

Рис. 21: Релиз с версией 1.0.0

Потом создаём журнал изменений, с помощью *standard-changelog --first-release* (рис. 22)

A terminal window with a dark background and light-colored text. The prompt is [eavernikovskaya@eavernikovskaya git-extended]\$. The command standard-changelog --first-release is entered. The output shows two green checkmarks: the first indicates that CHANGELOG.md was created, and the second indicates that changes were output to it. The prompt returns after the command completes.

```
[eavernikovskaya@eavernikovskaya git-extended]$ standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
[eavernikovskaya@eavernikovskaya git-extended]$
```

Рис. 22: Создание журнала изменений (1)

Добавляем журнал изменений в индекс (рис. 23)

```
[eavernikovskaya@eavernikovskaya git-extended]$ git add CHANGELOG.md
[eavernikovskaya@eavernikovskaya git-extended]$ git commit -am 'chore(site): add changelog'
[release/1.0.0 22ae9d0] chore(site): add changelog
2 files changed, 13 insertions(+), 2 deletions(-)
create mode 100644 CHANGELOG.md
[eavernikovskaya@eavernikovskaya git-extended]$
```

Рис. 23: Добавление журнала изменений в индекс (1)

Общепринятые коммиты

Заливаем релизную ветку в основную ветку введя *git flow release finish 1.0.0* (рис. 24)

```
[eavernikovskaya@eavernikovskaya git-extended]$ git flow release finish 1.0.0
Переключились на ветку «main»
Эта ветка соответствует «origin/main».
Merge made by the 'ort' strategy.
 CHANGELOG.md | 9 ++++++
 package.json | 6 +----
 2 files changed, 13 insertions(+), 2 deletions(-)
 create mode 100644 CHANGELOG.md
Уже на «main»
Ваша ветка опережает «origin/main» на 2 коммита.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
Переключились на ветку «develop»
Эта ветка соответствует «origin/develop».
Merge made by the 'ort' strategy.
 CHANGELOG.md | 9 ++++++
 package.json | 6 +----
 2 files changed, 13 insertions(+), 2 deletions(-)
 create mode 100644 CHANGELOG.md
Ветка release/1.0.0 удалена (была 22ae9d0).

Summary of actions:
- Release branch 'release/1.0.0' has been merged into 'main'
- The release was tagged 'v1.0.0'
- Release tag 'v1.0.0' has been back-merged into 'develop'
- Release branch 'release/1.0.0' has been locally deleted
- You are now on branch 'develop'

[eavernikovskaya@eavernikovskaya git-extended]$
```

Рис. 24: Релизная ветка в основной (1)

Далее отправляем данные на github с помощью *git push --all* и *git push --tags* (рис. 25)

```
[eavernikovskaya@eavernikovskaya git-extended]$ git push --all
Перечисление объектов: 8, готово.
Подсчет объектов: 100% (8/8), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 2.89 КиБ | 2.89 МБ/с, готово.
Всего 6 (изменений 1), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Katerok27153/git-extended.git
   afa7beb..4fe230d  develop -> develop
   afa7beb..60744f3  main -> main
[eavernikovskaya@eavernikovskaya git-extended]$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 155 байтов | 155.00 КиБ/с, готово.
Всего 1 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To https://github.com/Katerok27153/git-extended.git
 * [new tag]          v1.0.0 -> v1.0.0
[eavernikovskaya@eavernikovskaya git-extended]$
```

Рис. 25: Отправка данных на github (1)

Создаём релиз на github (рис. 26)

```
[eavernikovskaya@eavernikovskaya git-extended]$ gh release create v1.0.0 -F CHANGELOG.md  
https://github.com/Katerok27153/git-extended/releases/tag/v1.0.0  
[eavernikovskaya@eavernikovskaya git-extended]$
```

Рис. 26: Создание релиза на github (1)

Далее создаём ветку для новой функциональности введя (рис. 27)

```
[eavernikovskaya@eavernikovskaya git-extended]$ git flow feature start feature_branch
Переключились на новую ветку «feature/feature_branch»

Summary of actions:
- A new branch 'feature/feature_branch' was created, based on 'develop'
- You are now on branch 'feature/feature_branch'

Now, start committing on your feature. When done, use:

    git flow feature finish feature_branch

[eavernikovskaya@eavernikovskaya git-extended]$
```

Рис. 27: Создание ветки

После объединяем ветку `feature_branch` с `develop` (рис. 28)

```
[eavernikovskaya@eavernikovskaya git-extended]$ git flow feature finish feature_branch
Переключились на ветку «develop»
Эта ветка соответствует «origin/develop».
Уже актуально.
Ветка feature/feature_branch удалена (была 4fe230d).

Summary of actions:
- The feature branch 'feature/feature_branch' was merged into 'develop'
- Feature branch 'feature/feature_branch' has been locally deleted
- You are now on branch 'develop'

[eavernikovskaya@eavernikovskaya git-extended]$
```

Рис. 28: Объединение веток

Создаём релиз с версией 1.2.3, введя *git flow release start 1.2.3* (рис. 29)

```
[eavernikovskaya@eavernikovskaya git-extended]$ git flow release start 1.2.3
Переключились на новую ветку «release/1.2.3»

Summary of actions:
- A new branch 'release/1.2.3' was created, based on 'develop'
- You are now on branch 'release/1.2.3'

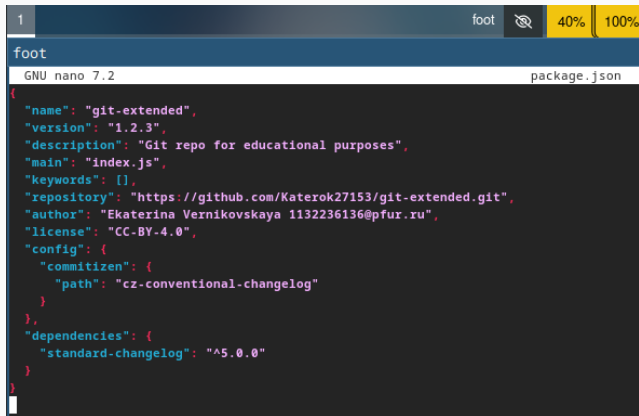
Follow-up actions:
- Bump the version number now!
- Start committing last-minute fixes in preparing your release
- When done, run:

    git flow release finish '1.2.3'

[eavernikovskaya@eavernikovskaya git-extended]$
```

Рис. 29: Релиз с версией 1.2.3

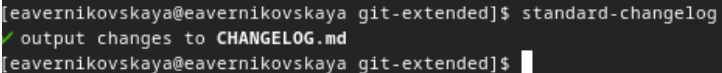
После редактируем файл `package.json`: изменяем версию на 1.2.3 (рис. 30)

A screenshot of a terminal window showing the nano text editor editing a file named package.json. The editor's title bar shows 'foot', a search icon, and progress indicators for 40% and 100%. The status bar at the bottom of the editor shows 'GNU nano 7.2' and 'package.json'. The code being edited is a JSON object for a package named 'git-extended'. The 'version' field is set to '1.2.3'. Other fields include 'description', 'main', 'keywords', 'repository', 'author', 'license', 'config' (with 'commitizen' and 'path'), 'dependencies', and 'standard-changelog'.

```
1 foot 40% 100%
foot
GNU nano 7.2 package.json
{
  "name": "git-extended",
  "version": "1.2.3",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "keywords": [],
  "repository": "https://github.com/Katerok27153/git-extended.git",
  "author": "Ekaterina Vernikovskaya 1132236136@pfur.ru",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  },
  "dependencies": {
    "standard-changelog": "^5.0.0"
  }
}
```

Рис. 30: Редактирование файла `package.json`

Снова создаём журнал изменений (рис. 31)

A terminal window with a dark background. The prompt is [eavernikovskaya@eavernikovskaya git-extended]\$. The command standard-changelog is entered. The output is a green checkmark followed by 'output changes to CHANGELOG.md'. The prompt is repeated on the next line with a cursor.

```
[eavernikovskaya@eavernikovskaya git-extended]$ standard-changelog
✓ output changes to CHANGELOG.md
[eavernikovskaya@eavernikovskaya git-extended]$
```

Рис. 31: Создание журнала изменений (2)

Добавляем журнал изменений в индекс (рис. 32)

```
[eavernikovskaya@eavernikovskaya git-extended]$ git add CHANGELOG.md
[eavernikovskaya@eavernikovskaya git-extended]$ git commit -am 'chore(site): update changelog'
[release/1.2.3 3cb456d] chore(site): update changelog
2 files changed, 5 insertions(+), 1 deletion(-)
[eavernikovskaya@eavernikovskaya git-extended]$
```

Рис. 32: Добавление журнала изменений в индекс (2)

Заливаем релизную ветку в основную введя команду *git flow release finish 1.2.3* (рис. 33)

```
[eavernikovskaya@eavernikovskaya git-extended]$ git flow release finish 1.2.3
Переключились на ветку «main»
Эта ветка соответствует «origin/main».
Merge made by the 'ort' strategy.
  CHANGELOG.md | 4 ++++
  package.json | 2 +-
  2 files changed, 5 insertions(+), 1 deletion(-)
Уже на «main»
Ваша ветка опережает «origin/main» на 3 коммита.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
Переключились на ветку «develop»
Эта ветка соответствует «origin/develop».
Merge made by the 'ort' strategy.
  CHANGELOG.md | 4 ++++
  package.json | 2 +-
  2 files changed, 5 insertions(+), 1 deletion(-)
Ветка release/1.2.3 удалена (была 3cb456d).

Summary of actions:
- Release branch 'release/1.2.3' has been merged into 'main'
- The release was tagged 'v1.2.3'
- Release tag 'v1.2.3' has been back-merged into 'develop'
- Release branch 'release/1.2.3' has been locally deleted
- You are now on branch 'develop'

[eavernikovskaya@eavernikovskaya git-extended]$
```

Рис. 33: Релизная ветка в основной (2)

Отправляем данные на github (рис. 34)

```
[eavernikovskaya@eavernikovskaya git-extended]$ git push --all
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 2.78 КиБ | 710.00 КиБ/с, готово.
Всего 6 (изменений 2), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/Katerok27153/git-extended.git
   4fe230d..ef44e67  develop -> develop
   60744f3..68c69cd  main -> main
[eavernikovskaya@eavernikovskaya git-extended]$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 156 байтов | 156.00 КиБ/с, готово.
Всего 1 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To https://github.com/Katerok27153/git-extended.git
 * [new tag]          v1.2.3 -> v1.2.3
[eavernikovskaya@eavernikovskaya git-extended]$
```

Рис. 34: Отправка данных на github (2)

И последним шагом создаём релиз на github (рис. 35)

```
[eavernikovskaya@eavernikovskaya git-extended]$ gh release create v1.2.3 -F CHANGELOG.md  
https://github.com/Katerok27153/git-extended/releases/tag/v1.2.3  
[eavernikovskaya@eavernikovskaya git-extended]$
```

Рис. 35: Создание релиза на github (2)

Общепринятые коммиты

Далее заходим на github и видим что всё получилось!!! Все релизы на github (рис. 36)

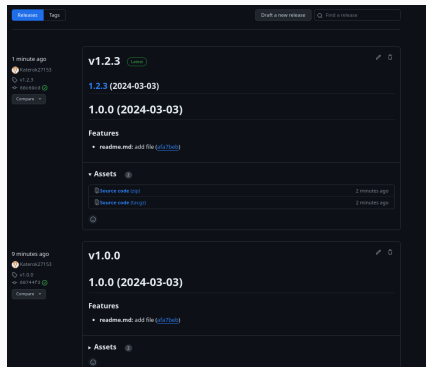


Рис. 36: Мои релизы

Подведение итогов

В ходе выполнения лабораторной работы мы получили навыки правильной работы с репозиториями git, а также научились создавать релизы.

1. Лабораторная работа №4 [Электронный ресурс] URL:
<https://esystem.rudn.ru/mod/page/view.php?id=1098937#org5411099>
2. Список лицензий [Электронный ресурс] URL: <https://spdx.org/licenses/>