

Отчёт по лабораторной работе №2

Операционные системы

Калашникова Ольга Сергеевна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	Установка git-flow	7
3.2	Установка Node.js	8
3.3	Настройка Node.js	9
3.4	Общепринятые коммиты (commitizen)	10
3.5	Общепринятые коммиты (standard-changelog)	10
3.6	Создание репозитория git и подключение репозитория к github .	11
3.7	Конфигурация общепринятых коммитов	12
3.8	Конфигурация git-flow	13
3.9	Разработка новой функциональности	17
3.10	Создание релиза git-flow	18
4	Выводы	22
	Список литературы	23

Список иллюстраций

3.1	Enable the copr repository	7
3.2	Install gitflow	8
3.3	Устанавливаем Node.js	8
3.4	Устанавливаем пакет npnm	9
3.5	Запуск npnm	9
3.6	Выполнение	9
3.7	Установка пакета commitizen	10
3.8	Установка пакета standard-changelog	10
3.9	Копирование репозитория	11
3.10	Первый коммит 1	11
3.11	Первый коммит 2	11
3.12	Проверка	12
3.13	Изменённый файл package.json	12
3.14	Добавление и коммит	13
3.15	Отправка на github	13
3.16	Инициализация	14
3.17	Загружаем репозиторий	14
3.18	Устанавливаем внешнюю ветку как вышестоящую	14
3.19	Создаём релиз	15
3.20	Создаём changelog	15
3.21	Добавление журнала изменений в индекс	15
3.22	git flow	16
3.23	Отправка данных 1	16
3.24	Отправка данных 2	16
3.25	Создадим релиз на github	17
3.26	Создание ветки	17
3.27	Объединение ветки feature_branch с develop	18
3.28	Создание релиза с версией 1.2.3	18
3.29	Обновлённый файл package.json	19
3.30	Журнал изменений	19
3.31	Релизная ветка	20
3.32	Выгрузка данных	20
3.33	Создание релиза	21

Список таблиц

1 Цель работы

Целью данной работы является получение навыков правильной работы с репозиториями git.

2 Задание

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

3 Выполнение лабораторной работы

3.1 Установка git-flow

Установка из коллекции репозитория Copr. Для начала включаем репозиторий copr при помощи команды `dnf copr enable elegos/gitflow` (рис. 3.1).

```
[oskalashnikova@oskalashnikova ~]$ sudo -i
[sudo] пароль для oskalashnikova:
[root@oskalashnikova ~]# 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 р
епозитория.

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

Рис. 3.1: Enable the copr repository

Устанавливаем пакет gitflow при помощи `dnf install gitflow` (рис. 3.2).

```
[root@oskalashnikova ~]# dnf install gitflow
Copr repo for gitflow owned by elegos          1.5 kB/s | 1.5 kB    00:01
Зависимости разрешены.
=====
Пакет
  Архитектура
    Версия      Репозиторий
=====
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
```

Рис. 3.2: Install gitflow

3.2 Установка Node.js

На Node.js базируется программное обеспечение для семантического версионирования и общепринятых коммитов. Устанавливаем Node.js при помощи команды `dnf install nodejs` (рис. 3.3).

```
[root@oskalashnikova ~]# dnf install nodejs
Последняя проверка окончания срока действия метаданных: 0:00:39 назад, Сб 09
мар 2024 09:03:50.
Зависимости разрешены.
=====
Пакет
  Архитектура
    Версия      Репозиторий
=====
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 M
Установка слабых зависимостей:
  nodejs-docs
    ppc64le 1:20.10.0-3.fc39 updates 8.1 M
  nodejs-full-i18n
    x86_64 1:20.10.0-3.fc39 updates 8.5 M
  nodejs-npm
    x86_64 1:10.2.3-1.20.10.0.3.fc39 updates 2.2 M
```

Рис. 3.3: Устанавливаем Node.js

Устанавливаем пакет `npm` при помощи `apt-get install npm` (в моём случае `dnf install npm`) (рис. 3.4).


```
[root@oskalashnikova ~]# dnf install pnpm
Последняя проверка окончания срока действия метаданных: 0:01:26 назад, Сб 09
мар 2024 09:03:50.
Зависимости разрешены.
=====
Пакет      Архитектура  Версия      Репозиторий  Размер
=====
Установка:
pnpm       noarch       8.12.0-1.fc39  updates      2.6 М
Результат транзакции
=====
Установка 1 Пакет
```

Рис. 3.4: Устанавливаем пакет pnpm

3.3 Настройка Node.js

Для работы с Node.js добавляем каталог с исполняемыми файлами, устанавливаемыми yarn, в переменную PATH.

Запускаем при помощи pnpm setup (рис. 3.5).

```
[root@oskalashnikova ~]# 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@oskalashnikova ~]#
```

Рис. 3.5: Запуск pnpm

Выполняем source ~/.bashrc (рис. 3.6).

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

Рис. 3.6: Выполнение

3.4 Общепринятые коммиты (commitizen)

Данная программа используется для помощи в форматировании коммитов. Устанавливаем пакет commitizen при помощи `pnpm add -g commitizen` (При этом устанавливается скрипт `git-cz`, который мы и будем использовать для коммитов) (рис. 3.7).

```
[root@oskalashnikova ~]# 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/5.4.2: 5,82 MB/5,82 MB, done
Progress: resolved 152, reused 0, downloaded 152, added 152, done

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

Done in 12.6s
[root@oskalashnikova ~]#
```

Рис. 3.7: Установка пакета commitizen

3.5 Общепринятые коммиты (standard-changelog)

Данная программа используется для помощи в создании логов. Устанавливаем командой `pnpm add -g standard-changelog` (рис. 3.8).

```
[root@oskalashnikova ~]# 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 5.9s
[root@oskalashnikova ~]#
```

Рис. 3.8: Установка пакета standard-changelog

3.6 Создание репозитория git и подключение репозитория к github

На самом GitHub создаём новый репозиторий с названием git-extended. Копируем его (рис. 3.9).

```
[root@oskalashnikova ~]# git clone --recursive https://github.com/lacrimell/git-extended.git
Клонирование в «git-extended»...
warning: Похоже, что вы клонировали пустой репозиторий.
[root@oskalashnikova ~]#
```

Рис. 3.9: Копирование репозитория

Делаем первый коммит (для начала я создала файл при помощи touch чтобы было изменение) (рис. 3.10).

```
[root@oskalashnikova git-extended]# touch README.md
[root@oskalashnikova git-extended]# git add .
[root@oskalashnikova git-extended]# git commit -m "first commit"
[main (корневой коммит) 090950c] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
[root@oskalashnikova git-extended]#
```

Рис. 3.10: Первый коммит 1

Выкладываем на github (git push -u origin main)(рис. 3.11).

```
[root@oskalashnikova .git]# git push -u origin main
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 861 байт | 861.00 КиБ/с, готово.
Всего 3 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To https://github.com/lacrimell/git-extended.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
[root@oskalashnikova .git]#
```

Рис. 3.11: Первый коммит 2

Проверяем (рис. 3.12).

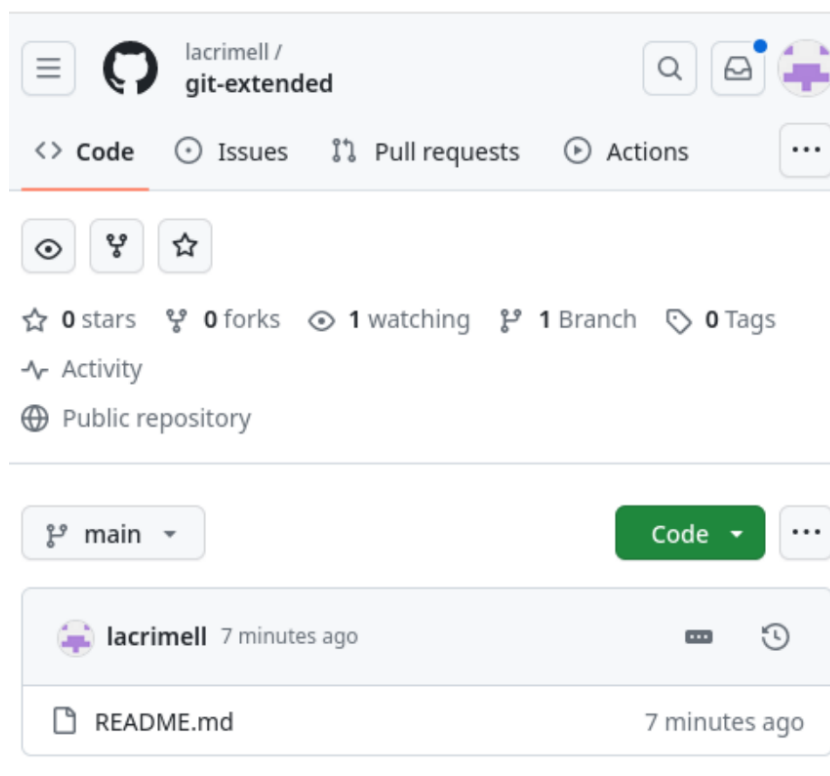


Рис. 3.12: Проверка

3.7 Конфигурация общепринятых коммитов

Конфигурация для пакетов Node.js `npm init` (с помощью текстового редактора меняю нужные поля) (рис. 3.13).

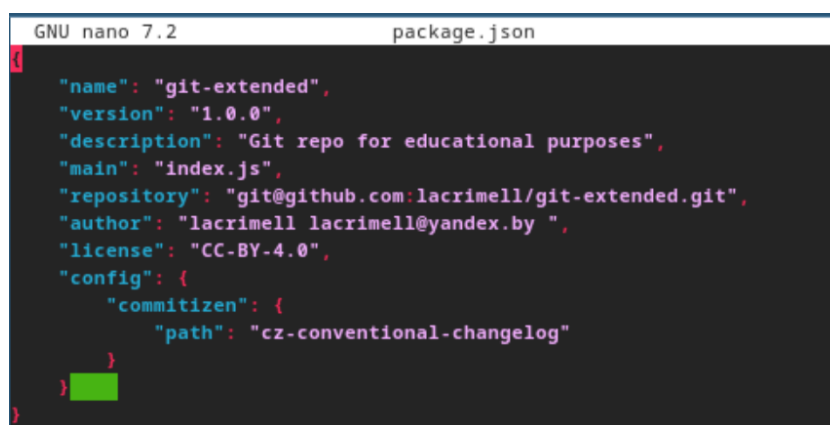


Рис. 3.13: Изменённый файл `package.json`

Добавляем новые файлы при помощи `git add` и выполняем коммит при помощи `git cz` (рис. 3.14).

```
[root@oskalashnikova git-extended]# git add .
[root@oskalashnikova 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)
? Write a short, imperative tense description of the change (max 94
chars):
(4) json
? 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 628c6d1] feat: json
1 file changed, 14 insertions(+)
create mode 100644 package.json
[root@oskalashnikova git-extended]#
```

Рис. 3.14: Добавление и коммит

Отправляем на github при помощи `git push` (рис. 3.15).

```
[root@oskalashnikova git-extended]# git push
Перечисление объектов: 4, готово.
Подсчет объектов: 100% (4/4), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 1.12 КиБ | 1.12 МиБ/с, готово.
Всего 3 (изменений 0), повторно использовано 0 (изменений 0), повторно и
спользовано пакетов 0
To https://github.com/lacrimell/git-extended.git
 72ffad3..628c6d1  main -> main
[root@oskalashnikova git-extended]#
```

Рис. 3.15: Отправка на github

3.8 Конфигурация git-flow

Инициализируем `git-flow` при помощи `git flow init` (Префикс для ярлыков устанавливаем в `v`) и проверяем что мы на той ветке (рис. 3.16).

```
[root@oskalashnikova 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? [/root/git-extended/.git/hooks]
[root@oskalashnikova git-extended]# git branch
* develop
  main
[root@oskalashnikova git-extended]#
```

Рис. 3.16: Инициализация

Загружаем весь репозиторий в хранилище с помощью `git push --all` (рис. 3.17).

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

Рис. 3.17: Загружаем репозиторий

Устанавливаем внешнюю ветку как вышестоящую для этой ветки (рис. 3.18).

```
[root@oskalashnikova git-extended]# git branch --set-upstream-to=origin/
develop develop
branch 'develop' set up to track 'origin/develop'.
[root@oskalashnikova git-extended]#
```

Рис. 3.18: Устанавливаем внешнюю ветку как вышестоящую

Создадим релиз с версией 1.0.0 (рис. 3.19).

```
[root@oskalashnikova 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'

[root@oskalashnikova git-extended]#
```

Рис. 3.19: Создаём релиз

Создадим журнал изменений (рис. 3.20).

```
[root@oskalashnikova git-extended]# standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
[root@oskalashnikova git-extended]#
```

Рис. 3.20: Создаём changelog

Добавим журнал изменений в индекс (рис. 3.21).

```
[root@oskalashnikova git-extended]# git add CHANGELOG.md
[root@oskalashnikova git-extended]# git commit -am 'chore(site): add changelog'
[release/1.0.0 bbb9915] chore(site): add changelog
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md
[root@oskalashnikova git-extended]#
```

Рис. 3.21: Добавление журнала изменений в индекс

Зальём релизную ветку в основную ветку при помощи `git flow release finish 1.0.0` (рис. 3.22).

```
[root@oskalashnikova 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 ++++++++
 1 file changed, 9 insertions(+)
 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 ++++++++
 1 file changed, 9 insertions(+)
 create mode 100644 CHANGELOG.md
Ветка release/1.0.0 удалена (была bbb9915).

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'

[root@oskalashnikova git-extended]#
```

Рис. 3.22: git flow

Отправим данные на github (git push –all) (рис. 3.23).

```
[root@oskalashnikova git-extended]# git push --all
Перечисление объектов: 6, готово.
Подсчет объектов: 100% (6/6), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (5/5), 2.77 КиБ | 2.77 МБ/с, готово.
Всего 5 (изменений 0), повторно использовано 0 (изменений 0), повторно и
спользовано пакетов 0
To https://github.com/lacrimell/git-extended.git
 628c6d1..b180d01 develop -> develop
 628c6d1..2ea142c main -> main
[root@oskalashnikova git-extended]#
```

Рис. 3.23: Отправка данных 1

Отправим данные на github (git push –tags) (рис. 3.24).

```
[root@oskalashnikova git-extended]# git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 151 байт | 151.00 КиБ/с, готово.
Всего 1 (изменений 0), повторно использовано 0 (изменений 0), повторно и
спользовано пакетов 0
To https://github.com/lacrimell/git-extended.git
 * [new tag]          v1.0.0 -> v1.0.0
[root@oskalashnikova git-extended]#
```

Рис. 3.24: Отправка данных 2

Создадим релиз на github. Для этого будем использовать утилиты работы с github (рис. 3.25).

```
[root@oskalashnikova git-extended]# gh release create v1.0.0 -F CHANGELOG.md
https://github.com/lacrimell/git-extended/releases/tag/v1.0.0
[root@oskalashnikova git-extended]#
```

Рис. 3.25: Создадим релиз на github

3.9 Разработка новой функциональности

Создадим ветку для новой функциональности при помощи `git flow feature start feature_branch` (рис. 3.26).

```
[root@oskalashnikova git-extended]# git flow feature start feature_branch
ПереклЮчилиcь 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

[root@oskalashnikova git-extended]#
```

Рис. 3.26: Создание ветки

Далее, продолжаем работу с git как обычно. (В задании ничего больше не указано). По окончании разработки новой функциональности следующим шагом следует объединить ветку `feature_branch` с `develop` при помощи `git flow feature finish feature_branch` (рис. 3.27).

```
[root@oskalashnikova git-extended]# ls
CHANGELOG.md package.json README.md
[root@oskalashnikova 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 b180d01).

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'
[root@oskalashnikova git-extended]#
```

Рис. 3.27: Объединение ветки feature_branch с develop

3.10 Создание релиза git-flow

Создадим релиз с версией 1.2.3 при помощи команды `git flow release start 1.2.3` (рис. 3.28).

```
[root@oskalashnikova 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'

[root@oskalashnikova git-extended]#
```

Рис. 3.28: Создание релиза с версией 1.2.3

Обновите номер версии в файле `package.json`. Установите её в 1.2.3. (рис. 3.29).

```

package.json  [-M--] 21 L:[ 1+ 2 3/ 15] *(51 / 383b) 0034[*][X]
{
  "name": "git-extended",
  "version": "1.2.3",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:lacrimell/git-extended.git",
  "author": "lacrimell lacrimell@yandex.by",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}

```

Рис. 3.29: Обновлённый файл package.json

Создадим и добавим журнал изменений в индекс (рис. 3.30).

```

[root@oskalashnikova git-extended]# standard-changelog
✓ output changes to CHANGELOG.md
[root@oskalashnikova git-extended]# git add CHANGELOG.md
[root@oskalashnikova git-extended]# git commit -am 'chore(site): update
changelog'
[release/1.2.3 bc6946f] chore(site): update changelog
2 files changed, 5 insertions(+), 1 deletion(-)
[root@oskalashnikova git-extended]#

```

Рис. 3.30: Журнал изменений

Зальём релизную ветку в основную ветку при помощи команды `git flow release finish 1.2.3` (рис. 3.31).

```
[root@oskalashnikova 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 bc6946f).

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'

[root@oskalashnikova git-extended]#
```

Рис. 3.31: Релизная ветка

Отправим данные на github (рис. 3.32).

```
[root@oskalashnikova git-extended]# git push --all
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 2.76 КиБ | 2.76 МиБ/с, готово.
Всего 6 (изменений 2), повторно использовано 0 (изменений 0), повторно и
спользовано пакетов 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/lacrimell/git-extended.git
   b180d01..7e25fcd  develop -> develop
   2ea142c..a93e4f5  main -> main
[root@oskalashnikova git-extended]# git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 149 байтов | 149.00 КиБ/с, готово.
Всего 1 (изменений 0), повторно использовано 0 (изменений 0), повторно и
спользовано пакетов 0
To https://github.com/lacrimell/git-extended.git
 * [new tag]         v1.2.3 -> v1.2.3
[root@oskalashnikova git-extended]#
```

Рис. 3.32: Выгрузка данных

Создаём релиз на github с комментарием из журнала изменений (рис. 3.33).

```
[root@oskalashnikova git-extended]# gh release create v1.2.3 -F CHANGELOG.md  
https://github.com/lacrimell/git-extended/releases/tag/v1.2.3  
[root@oskalashnikova git-extended]#
```

Рис. 3.33: Создание релиза

4 Выводы

Я получила навыки правильной работы с репозиториями git.

Список литературы