

Отчёта по лабораторной работе

Лабораторная работа №4

Диана Садова Алексеевна

Содержание

1	Цель работы	6
2	Задание	7
3	Последовательность выполнения работы	8
3.1	Установка программного обеспечения	8
3.1.1	Установка git-flow	8
3.1.2	Установка Node.js	9
3.1.3	Настройка Node.js	9
3.1.4	Общепринятые коммиты	10
4	Выводы	23
	Список литературы	24

Список иллюстраций

3.1	Устанавливаем коллекции репозитория Copr	8
3.2	Установка gitflow	9
3.3	Установка nodejs	9
3.4	Установка npm	9
3.5	Добавление каталога с исполняемыми файлами	10
3.6	Выполняем программу source ~/.bashrc	10
3.7	Добавляем программу для форматирования коммитов	11
3.8	Добавляем программу для помощи в создании логов	11
3.9	Создаем новый репозиторий с именем git-extended	12
3.10	Делаем коммит и отправляем на github	12
3.11	Подключаем конфигурацию для пакетов Node.js	13
3.12	Переписываем файл package.json	13
3.13	Добавляем новые файлы	13
3.14	Выполняем коммит	14
3.15	Отправляем файл на github	14
3.16	Изменяем конфигурацию git-flow	15
3.17	Проверяем, что мы на ветке develop	15
3.18	Загружаем репозиторий в хранилище	15
3.19	Устанавливаем внешнюю ветку	16
3.20	Создаем релиз версии 1.0.0	16
3.21	Создаем журнал изменений	16
3.22	Добавляем в индекс журнал изменений	17
3.23	Загружаем релизную ветку в основную ветку	17
3.24	Отправляем данные на github	18
3.25	Отправляем данные на github	18
3.26	Создаем релиз на github	18
3.27	Создадим новую ветку	19
3.28	Объединяем две ветки	19
3.29	Создаем новый релиз с версией 1.2.3	20
3.30	Обновляем номер версии в файле package.json	20
3.31	Создаем новый журнал изменений	20
3.32	Добавляем в индекс журнал изменений	21
3.33	Добавляем в индекс журнал изменений	21
3.34	Загружаем релизную ветку в основную ветку	21
3.35	Отправляем данные на github	22
3.36	Отправляем данные на github	22

3.37 Создаем новый релиз на github	22
--	----

Список таблиц

1 Цель работы

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 Последовательность выполнения работы

3.1 Установка программного обеспечения

3.1.1 Установка git-flow

Linux

Fedora

Установка из коллекции репозитория Copr (<https://copr.fedorainfracloud.org/coprs/elegos/gitflow/>), (рис. [-@fig:002]).

```
[root@fedora ~]# 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@fedora ~]#
```

Рис. 3.1: Устанавливаем коллекции репозитория Copr


```
Установлен:
gitflow-1.12.3-1.fc34.x86_64

Выполнено!
[root@fedora ~]#
```

Рис. 3.2: Установка gitflow

После завершения этих программ можем двигаться дальше

3.1.2 Установка Node.js

На Node.js базируется программное обеспечение для семантического версионирования и общепринятых коммитов. (рис. 3.3),(рис. 3.4).

Fedora

```
Установлен:
nodejs-1:20.10.0-3.fc39.x86_64      nodejs-docs-1:20.10.0-3.fc39.noarch      nodejs-full-118n-1:20.10.0-3.fc39.x86_64
nodejs-libs-1:20.10.0-3.fc39.x86_64  nodejs-npm-1:10.2.3-1.20.10.0-3.fc39.x86_64

Выполнено!
[root@fedora ~]#
```

Рис. 3.3: Установка nodejs

```
[root@fedora ~]# apt-get install pnpm
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
E: Невозможно найти пакет pnpm
[root@fedora ~]#
```

Рис. 3.4: Установка pnpm

Установка этих систем позволит выполнить данную лабораторную

3.1.3 Настройка Node.js

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

Запустите:(рис. [-@fig:005]).

```
[root@fedora ~]# 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
```

Рис. 3.5: Добавление каталога с исполняемыми файлами

Перелогиньтесь, или выполните:(рис. [-@fig:006]).

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

Рис. 3.6: Выполняем программу source ~/.bashrc

3.1.4 Общепринятые коммиты

1)commitizen

Данная программа используется для помощи в форматировании коммитов.(рис. [-@fig:007]).

```
[root@fedora ~]# 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 19.5s
[root@fedora ~]#
```

Рис. 3.7: Добавляем программу для форматирования коммитов

При этом устанавливается скрипт `git-cz`, который мы и будем использовать для коммитов

После завершения установки переходим к пункту 2

2)standard-changelog

Данная программа используется для помощи в создании логов.(рис. [-@fig:008]).

```
[root@fedora ~]# 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 7.5s
[root@fedora ~]#
```

Рис. 3.8: Добавляем программу для помощи в создании логов

После завершения установки переходим к пункту 3

3)Практический сценарий использования git

1.Создание репозитория git

1.Подключение репозитория к github

Создайте репозиторий на GitHub. Для примера назовём его git-extended.рис. [-@fig:009]).

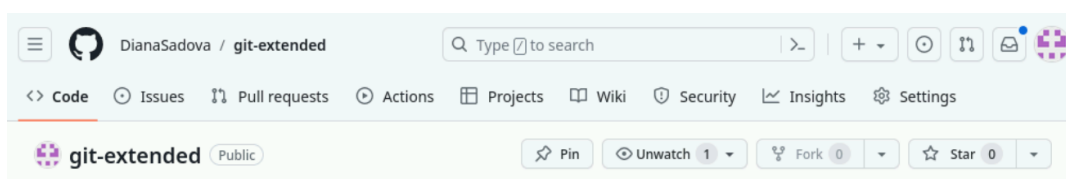


Рис. 3.9: Создаем новый репозиторий с именем git-extended

Делаем первый коммит и выкладываем на github:(рис. [-@fig:010]).

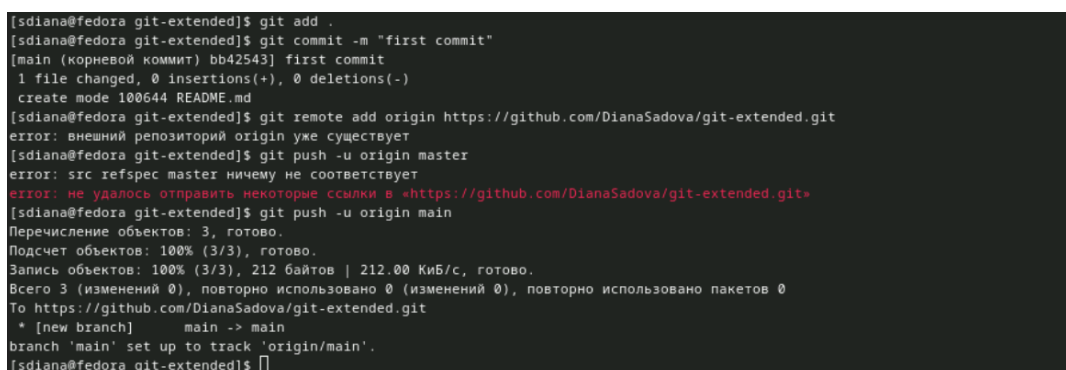


Рис. 3.10: Делаем коммит и отправляем на github

Для данного действия, я клонировала наш новый репозиторий и создала в нем файл README.md

2.Конфигурация общепринятых коммитов

Конфигурация для пакетов Node.js(рис. [-@fig:011]).

```
[sdiana@fedora git-extended]$ pnpm init
ERR_PNPM_PACKAGE_JSON_EXISTS package.json already exists
[sdiana@fedora git-extended]$
```

Рис. 3.11: Подключаем конфигурацию для пакетов Node.js

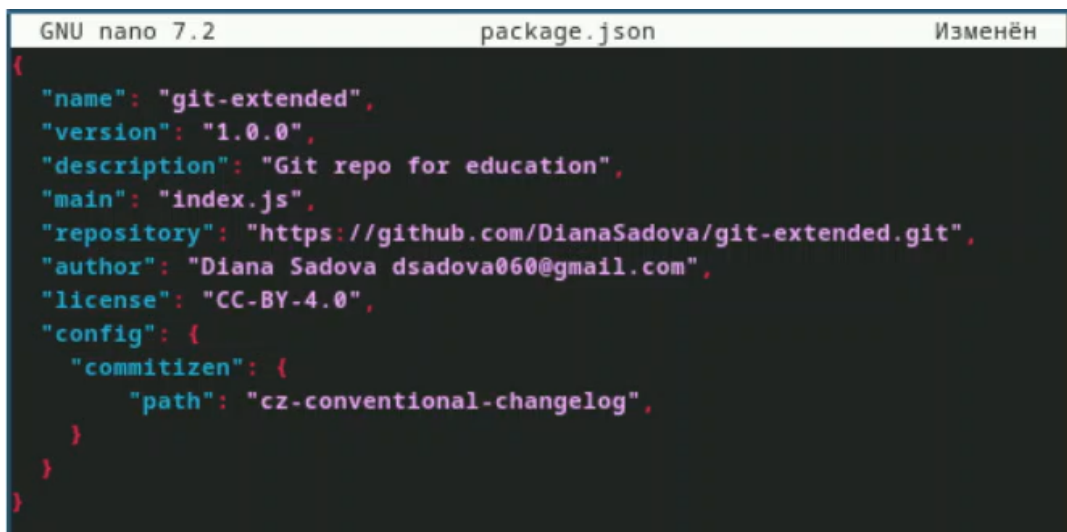
Необходимо заполнить несколько параметров пакета.

Название пакета.

Лицензия пакета. Список лицензий для npm: <https://spdx.org/licenses/>. Пр
BY-4.0.

Сконфигурируем формат коммитов. Для этого добавим в файл `package.json` команду д

Таким образом, файл `package.json` приобретает вид:(рис. [-@fig:012]).



```
GNU nano 7.2 package.json Изменён
{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "Git repo for education",
  "main": "index.js",
  "repository": "https://github.com/DianaSadova/git-extended.git",
  "author": "Diana Sadova dsadova060@gmail.com",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog",
    }
  }
}
```

Рис. 3.12: Переписываем файл `package.json`

Данная процедура нам поможет при дальнейшей работе

Добавим новые файлы:(рис. [-@fig:013]).

```
[sdiana@fedora git-extended]$ git add .
```

Рис. 3.13: Добавляем новые файлы

Выполним коммит:(рис. [-@fig:014]).

```
[sdiana@fedora git-extended]$ git commit -m "Добавят"
[main 9032b5b] Добавят
 2 files changed, 37 insertions(+)
 create mode 100644 package-lock.json
 create mode 100644 package.json
```

Рис. 3.14: Выполняем коммит

Выполняем коммит для нового файла

Отправим на github:(рис. [-@fig:015]).

```
[sdiana@fedora git-extended]$ git push
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (4/4), готово.
Запись объектов: 100% (4/4), 773 байта | 773.00 КиБ/с, готово.
Всего 4 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To https://github.com/DianaSadova/git-extended.git
   bb42543..9032b5b  main -> main
[sdiana@fedora git-extended]$
```

Рис. 3.15: Отправляем файл на github

3.Конфигурация git-flow

Инициализируем git-flow(рис. [-@fig:016]).

```
[sdiana@fedora git-extended]$ git flow init -f

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

Which branch should be used for integration of the "next release"?
  - develop
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/sdiana/git-extended/.git/hooks]
```

Рис. 3.16: Изменяем конфигурацию git-flow

Префикс для ярлыков установим в v.

Проверьте, что Вы на ветке develop:(рис. [-@fig:017]).

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

[sdiana@fedora 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/DianaSadova/git-extended/pull/new/develop
remote:
To https://github.com/DianaSadova/git-extended.git
* [new branch]      develop -> develop
```

Рис. 3.17: Проверяем, что мы на ветке develop

Убедились, что мы на нужной ветке, ведь в последствии она нам понадобится

Загрузите весь репозиторий в хранилище:(рис. [-@fig:018]).

```
[sdiana@fedora 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/DianaSadova/git-extended/pull/new/develop
remote:
To https://github.com/DianaSadova/git-extended.git
* [new branch]      develop -> develop
```

Рис. 3.18: Загружаем репозиторий в хранилище

Установите внешнюю ветку как вышестоящую для этой ветки:(рис. [-@fig:019]).

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

Рис. 3.19: Устанавливаем внешнюю ветку

Создадим релиз с версией 1.0.0(рис. [-@fig:020]).

```
[sdiana@fedora 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'

[sdiana@fedora git-extended]$
```

Рис. 3.20: Создаем релиз версии 1.0.0

Создадим журнал изменений(рис. [-@fig:021]).

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

Рис. 3.21: Создаем журнал изменений

Что бы воспользоваться программой standard-changelog нужно было ее сначала установить (проверенно на собственном опыте)

Добавим журнал изменений в индекс(рис. [-@fig:022]).


```
[sdiana@fedora git-extended]$ git add CHANGELOG.md
[sdiana@fedora git-extended]$ git commit -am 'chore(site): add changelog'
```

Рис. 3.22: Добавляем в индекс журнал изменений

Зальём релизную ветку в основную ветку(рис. [-@fig:023]).

```
[sdiana@fedora git-extended]$ git flow release finish 1.0.0
Branches 'main' and 'origin/main' have diverged.
And local branch 'main' is ahead of 'origin/main'.
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 | 4 +++
1 file changed, 4 insertions(+)
create mode 100644 CHANGELOG.md
Ветка release/1.0.0 удалена (была 611149e).

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'
```

Рис. 3.23: Загружаем релизную ветку в основную ветку

Отправим данные на github(рис. [-@fig:024]),(рис. [-@fig:025]).

```
[sdiana@fedora 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% (4/4), готово.
Запись объектов: 100% (5/5), 516 байтов | 516.00 КиБ/с, готово.
Всего 5 (изменений 3), повторно использовано 0 (изменений 0), повторно и
спользовано пакетов 0
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
To https://github.com/DianaSadova/git-extended.git
  9032b5b..39f2fd8  develop -> develop
  9032b5b..00e8f6f  main -> main
[sdiana@fedora git-extended]$
```

Рис. 3.24: Отправляем данные на github

```
[sdiana@fedora git-extended]$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 158 байтов | 158.00 КиБ/с, готово.
Всего 1 (изменений 0), повторно использовано 0 (изменений 0), повторно и
спользовано пакетов 0
To https://github.com/DianaSadova/git-extended.git
 * [new tag]          v1.0.0 -> v1.0.0
[sdiana@fedora git-extended]$
```

Рис. 3.25: Отправляем данные на github

Создадим релиз на github. Для этого будем использовать утилиты работы с github ([@fig:026]).

```
[sdiana@fedora git-extended]$ gh release create v1.0.0 -F CHANGELOG.mc
open CHANGELOG.mc: no such file or directory
```

Рис. 3.26: Создаем релиз на github

2. Работа с репозиторием git

1. Разработка новой функциональности

Создадим ветку для новой функциональности: (рис. [-@fig:027]).

```
[sdiana@fedora 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
```

Рис. 3.27: Создадим новую ветку

Далее, продолжаем работу с git как обычно.

По окончании разработки новой функциональности следующим шагом следует объединить ветки (рис. 3.28).

```
[sdiana@fedora 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 39f2fd8).

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'
```

Рис. 3.28: Объединяем две ветки

2. Создание релиза git-flow

Создадим релиз с версией 1.2.3:(рис. 3.29).

```
[sdiana@fedora 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'
```

Рис. 3.29: Создаем новый релиз с версией 1.2.3

Обновите номер версии в файле `package.json`. Установите её в 1.2.3., (рис. [-@fig:029]).

```
name": "git-extended",
"version": "1.2.3",
"description": "Git repo for educational purposes",
```

Рис. 3.30: Обновляем номер версии в файле `package.json`

Нам понадобится это обновление, так как без него не будут работать следующие программы

Создадим журнал изменений(рис. [-@fig:031]).

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

Рис. 3.31: Создаем новый журнал изменений

Добавим журнал изменений в индекс(рис. [-@fig:032]),(рис. [-@fig:033]).

```
[sdiana@fedora git-extended]$ git add CHANGELOG.md
[sdiana@fedora git-extended]$ git commit -am 'chore(site): update change log'
```

Рис. 3.32: Добавляем в индекс журнал изменений

```
[release/1.2.3 8f385a4] chore(site): update changelog
 2 files changed, 5 insertions(+), 1 deletion(-)
[sdiana@fedora 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 +-

```

Рис. 3.33: Добавляем в индекс журнал изменений

Зальём релизную ветку в основную ветку(рис. [-@fig:034]).

```
[sdiana@fedora git-extended]$ git flow release finish 1.2.3
Fatal: Branch 'release/1.2.3' does not exist and is required.
```

Рис. 3.34: Загружаем релизную ветку в основную ветку

Отправим данные на github(рис. [-@fig:035]),(рис. [-@fig:036]).

```
[sdiana@fedora 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), 731 байт | 365.00 КиБ/с, готово.
Всего 6 (изменений 3), повторно использовано 0 (изменений 0), повторно и
спользовано пакетов 0
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To https://github.com/DianaSadova/git-extended.git
   39f2fd8..b114f21  develop -> develop
   00e8f6f..da5202a  main -> main
```

Рис. 3.35: Отправляем данные на github

```
[sdiana@fedora git-extended]$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 157 байтов | 157.00 КиБ/с, готово.
Всего 1 (изменений 0), повторно использовано 0 (изменений 0), повторно и
спользовано пакетов 0
To https://github.com/DianaSadova/git-extended.git
 * [new tag]           v1.2.3 -> v1.2.3
```

Рис. 3.36: Отправляем данные на github

Создадим релиз на github с комментарием из журнала изменений: (рис. [-@fig:037]).

```
[sdiana@fedora git-extended]$ gh release create v1.2.3 -F CHANGELOG.mc
open CHANGELOG.mc: no such file or directory
[sdiana@fedora git-extended]$
```

Рис. 3.37: Создаем новый релиз на github

4 Выводы

Получили навыки правильной работы с репозиториями git и освоили новые способы работы с ним

Список литературы