

Отчет по лабораторной работе №4

Шубина София Антоновна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	19
	Список литературы	20

Список иллюстраций

4.1	Установка из коллекции репозитория Corp	8
4.2	Установка Node.js	9
4.3	Запуск npm setup	9
4.4	Установка программы и скрипта	10
4.5	Скачивание standard-changelog	10
4.6	Создание репозитория	10
4.7	Клонируем репозиторий	11
4.8	Делаем первый commit	11
4.9	Конфигурация для пакетов Node.js	11
4.10	Вносим изменения	12
4.11	Добавим новые файлы	13
4.12	Инициализируем git-flow	14
4.13	Проверим, что мы на ветке	14
4.14	Загрузим весь репозиторий в хранилище	14
4.15	Установим внешнюю ветку как вышестоящую для этой ветки	15
4.16	Создадим журнал изменений и добавим его в индекс	15
4.17	Зальем релизную ветку в основную ветку и отправим данные	16
4.18	Обновление номера версии	17
4.19	Создание журнала изменений, добавление в индекс, заливаем ветку в основную и отправляем данные	18

Список таблиц

1 Цель работы

Получение навыков правильной работы с репозиториями git.

2 Задание

Здесь приводится описание задания в соответствии с рекомендациями методического пособия и выданным вариантом.

3 Теоретическое введение

Рабочий процесс Gitflow Workflow. Будем описывать его с использованием пакета git-flow [1]

4 Выполнение лабораторной работы

Установка программного обеспечения Установка git-flow Linux Fedora Установка из коллекции репозитория Copr (<https://copr.fedorainfracloud.org/coprs/elegos/gitflow/>)
Enable the copr repository dnf copr enable elegos/gitflow # Install gitflow dnf install gitflow (рис. 4.1).

```
[root@sashubina ~]# 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>,
и 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@sashubina ~]# dnf install gitflow
Copr repo for gitflow owned by elegos
Fedora 39 - x86_64 - Updates 3.3 kB/s | 1.5 kB 00:00
Fedora 39 - x86_64 - Updates 71 kB/s | 15 kB 00:00
Зависимости разрешены 2.0 MB/s | 2.9 MB 00:01

=====
Пакет Архитектура Версия Репозиторий Размер
=====
Установка:
gitflow x86_64 1.12.3-1.fc34 copr.fedorainfracloud.org/elegos:gitflow 57 k
=====
Результат транзакции
=====
Установка 1 Пакет

Объем загрузки: 57 k
Объем изменений: 262 k
Продолжить? [д/н]: д
Загрузка пакетов:
gitflow-1.12.3-1.fc34.x86_64.rpm 381 kB/s | 57 kB 00:00
=====
Общий размер 376 kB/s | 57 kB 00:00
Copr repo for gitflow owned by elegos 13 kB/s | 998 B 00:00
Импорт GPG-ключа 0x80F63AA3:
Идентификатор пользователя: "elegos:gitflow (None) <elegos@gitflow@copr.fedorahosted.org>"
Отпечаток: 9357 9D88 49C5 C563 9A22 823C AB66 88F6 3AA3
Источник: https://download.copr.fedorainfracloud.org/results/elegos/gitflow/pubkey.gpg
Продолжить? [д/н]: д
Импорт ключа успешно завершен
Проверка транзакции
```

Рис. 4.1: Установка из коллекции репозитория Copr

Установка Node.js На Node.js базируется программное обеспечение для семантического версионирования и общепринятых коммитов. Fedora dnf install nodejs apt-get install npm (рис. 4.2).


```

nodejs-docs      noarch      1:20.10.0-3.fc39      updates      8.1 M
nodejs-fulli18n  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

=====
Результат транзакции
=====
Установка 5 Пакетов

Объем загрузки: 34 М
Объем изменений: 183 М
Продолжить? [д/н]: д
Загрузка пакетов:
(1/5): nodejs-20.10.0-3.fc39.x86_64.rpm      402 kB/s | 48 kB   00:00
(2/5): nodejs-docs-20.10.0-3.fc39.noarch.rpm  2.1 MB/s | 8.1 MB  00:03
(3/5): nodejs-fulli18n-20.10.0-3.fc39.x86_64.rpm 2.1 MB/s | 8.5 MB  00:04
(4/5): nodejs-libs-20.10.0-3.fc39.x86_64.rpm    3.5 MB/s | 15 MB   00:04
(5/5): nodejs-npm-10.2.3-1.20.10.0-3.fc39.x86_64.rpm 3.6 MB/s | 2.2 MB  00:00
-----
Общий размер                               7.0 MB/s | 34 MB   00:04
Проверка транзакции
Проверка транзакции успешно завершена.
дет проверка транзакции
тест транзакции проведен успешно.
Выполнение транзакции
Запуск скриплетта: nodejs-1:20.10.0-3.fc39.x86_64      1/1
Подготовка :                                           1/1
Установка : nodejs-docs-1:20.10.0-3.fc39.noarch        1/5
Установка : nodejs-libs-1:20.10.0-3.fc39.x86_64        2/5
Установка : nodejs-fulli18n-1:20.10.0-3.fc39.x86_64    3/5
Установка : nodejs-npm-1:10.2.3-1.20.10.0-3.fc39.x86_64 4/5
Установка : nodejs-1:20.10.0-3.fc39.x86_64            5/5
Запуск скриплетта: nodejs-1:20.10.0-3.fc39.x86_64      5/5
Проверка : nodejs-1:20.10.0-3.fc39.x86_64            1/5
Проверка : nodejs-docs-1:20.10.0-3.fc39.noarch        2/5
Проверка : nodejs-fulli18n-1:20.10.0-3.fc39.x86_64    3/5
Проверка : nodejs-libs-1:20.10.0-3.fc39.x86_64        4/5
Проверка : nodejs-npm-1:10.2.3-1.20.10.0-3.fc39.x86_64 5/5

Установлен:
nodejs-1:20.10.0-3.fc39.x86_64      nodejs-docs-1:20.10.0-3.fc39.noarch      nodejs-fulli18n-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!
[sashubina@sashubina ~]#

```

Рис. 4.2: Установка Node.js

Настройка Node.js Для работы с Node.js добавим каталог с исполняемыми файлами, устанавливаемыми yarn, в переменную PATH. Запустим: `pnpm setup` (рис. 4.3).

```

[sashubina@sashubina ~]$ pnpm setup
Appended new lines to /home/sashubina/.bashrc

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

To start using pnpm, run:
source /home/sashubina/.bashrc

```

Рис. 4.3: Запуск pnpm setup

Перелогинемся или выполним: `source ~/.bashrc`

Общепринятые коммиты commitizen Данная программа используется для помощи в форматировании коммитов. `pnpm add -g commitizen` При этом устанавливается скрипт `git-cz`, который мы и будем использовать для коммитов. (рис. 4.4).

```
[root@sashubina ~]# 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 6.4s
[root@sashubina ~]#
```

Рис. 4.4: Установка программы и скрипта

standard-changelog Данная программа используется для помощи в создании логов. `pnpm add -g standard-changelog` (рис. 4.5).

```
[root@sashubina ~]# 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 4.3s
[root@sashubina ~]#
```

Рис. 4.5: Скачивание standard-changelog

Практический сценарий использования git Создание репозитория git Подключение репозитория к github Создадим репозиторий на GitHub. Для примера назовём его git-extended. (рис. 4.6).

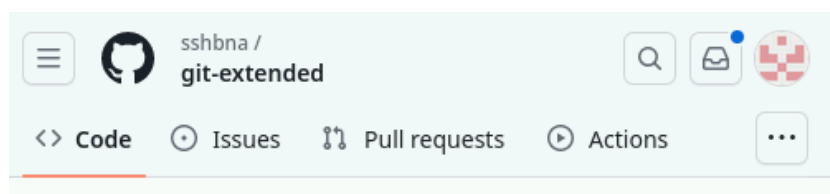


Рис. 4.6: Создание репозитория

Делаем первый коммит и выкладываем на github: Для начала клонируем репозиторий (рис. 4.7).

```
[sashubina@sashubina ~]$ git clone --recursive git@github.com:sshbna/git-extended.git
Клонирование в «git-extended»...
```

Рис. 4.7: Клонирование репозитория

git commit -m "first commit" git remote add origin git@github.com:/git-extended.git git push -u origin master (рис. 4.8).

```
[sashubina@sashubina git-extended]$ git commit -m "first commit"
[main b804235] first commit
1 file changed, 1 insertion(+)
[sashubina@sashubina git-extended]$ git remote add origin git@github.com:sshbna/git-extended.git
error: внешний репозиторий origin уже существует
[sashubina@sashubina git-extended]$ git push -u origin master
error: src refspec master ничему не соответствует
error: не удалось отправить некоторые ссылки в «github.com:sshbna/git-extended.git»
[sashubina@sashubina git-extended]$ git push -u origin main
Перечисление объектов: 6, готово.
Подсчет объектов: 100% (6/6), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (6/6), 1.71 КиБ | 1.71 МиБ/с, готово.
Всего 6 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To github.com:sshbna/git-extended.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
[sashubina@sashubina git-extended]$
```

Рис. 4.8: Делаем первый commit

Конфигурация общепринятых коммитов Конфигурация для пакетов Node.js
pnpm init (рис. 4.9).

```
[sashubina@sashubina git-extended]$ pnpm init
Wrote to /home/sashubina/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"
}
[sashubina@sashubina git-extended]$
```

Рис. 4.9: Конфигурация для пакетов Node.js

Необходимо заполнить несколько параметров пакета. Название пакета. Лицензия пакета. Список лицензий для npm: <https://spdx.org/licenses/>. Предлагается выбирать лицензию CC-BY-4.0. Сконфигурируем формат коммитов. Для этого добавим в файл package.json команду для формирования коммитов: "config": { "commitizen": { "path": "cz-conventional-changelog" } } Таким образом, файл package.json приобретает вид: { "name": "git-extended", "version": "1.0.0", "description": "Git repo for educational purposes", "main": "index.js", "repository": "git@github.com:username/git-extended.git", "author": "Name Surname username@gmail.com", "license": "CC-BY-4.0", "config": { "commitizen": { "path": "cz-conventional-changelog" } } } (рис. 4.10).

```
{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "keywords": [],
  "repository": "git@github.com:sshbna/git-extended.git",
  "author": "Shubina Sofia 1132232885@pfur.ru",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 4.10: Вносим изменения

Добавим новые файлы: git add . Выполним коммит: git cz Отправим на github: git push (рис. 4.11).

```

foot
  "author": "",
  "license": "ISC"
}
[sashubina@sashubina git-extended]$ nano package.json
[sashubina@sashubina git-extended]$ git add .
[sashubina@sashubina git-extended]$ git cz
Parsing JSON at /home/sashubina/git-extended/package.json for commitizen
config failed:

[sashubina@sashubina git-extended]$ nano ~/czrc
[sashubina@sashubina git-extended]$ nano package.json
[sashubina@sashubina git-extended]$ git cz
Parsing JSON at /home/sashubina/git-extended/package.json for commitizen
config failed:

[sashubina@sashubina git-extended]$ nano package.json
[sashubina@sashubina 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):
  (10) added 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 baacc87] feat(readme.md): added file
 1 file changed, 15 insertions(+)
 create mode 100644 package.json
[sashubina@sashubina git-extended]$ git push
Перечисление объектов: 4, готово.
Подсчет объектов: 100% (4/4), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 1.17 КиБ | 170.00 КиБ/с, готово.
Всего 3 (изменений 0), повторно использовано 0 (изменений 0), повторно и
спользовано пакетов 0
To github.com:sashubina/git-extended.git
   b804235..baacc87  main -> main
[sashubina@sashubina git-extended]$ 

```

Рис. 4.11: Добавим новые файлы

Конфигурация git-flow Инициализируем git-flow git flow init Префикс для ярлыков установим в v. (рис. 4.12).

```

To force reinitialization, use: git flow init -f
[sashubina@sashubina 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/sashubina/git-extended/.git/hooks]
[sashubina@sashubina git-extended]$ 

```

Рис. 4.12: Инициализируем git-flow

Проверьте, что Вы на ветке develop: git branch (рис. 4.13).

```

[sashubina@sashubina git-extended]$ git branch
* develop
  main

```

Рис. 4.13: Проверим, что мы на ветке

Загрузите весь репозиторий в хранилище: git push --all (рис. 4.14).

```

[sashubina@sashubina 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/sshbna/git-extended/pull/new/develop
remote:
To github.com:sshbna/git-extended.git
 * [new branch]      develop -> develop
[sashubina@sashubina git-extended]$ 

```

Рис. 4.14: Загрузим весь репозиторий в хранилище

Установим внешнюю ветку как вышестоящую для этой ветки: git branch --set-upstream-to=origin/develop develop Создадим релиз с версией 1.0.0 git flow release start 1.0.0 (рис. 4.15).

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

[sashubina@sashubina git-extended]$ git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
[sashubina@sashubina git-extended]$ git flow release start 1.0.0
Fatal: There is an existing release branch '1.0.0'. Finish that one first.
[sashubina@sashubina git-extended]$
```

Рис. 4.15: Установим внешнюю ветку как вышестоящую для этой ветки

Создадим журнал изменений `standard-changelog --first-release` Добавим журнал изменений в индекс `git add CHANGELOG.md git commit -am 'chore(site): add changelog'` (рис. 4.16).

```
[sashubina@sashubina git-extended]$ standard-changelog
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
[sashubina@sashubina git-extended]$ standard-changelog --first-release
✓ output changes to CHANGELOG.md
[sashubina@sashubina git-extended]$ git add CHANGELOG.md
[sashubina@sashubina git-extended]$ git commit -am 'chore(site):update changelog'
[release/1.0.0 6f42435] chore(site):update changelog
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md
[sashubina@sashubina git-extended]$
```

Рис. 4.16: Создадим журнал изменений и добавим его в индекс

Зальём релизную ветку в основную ветку `git flow release finish 1.0.0` Отправим данные на github `git push --all git push --tags` (рис. 4.17).

```

(используйте «git push», чтобы опубликовать ваши локальные коммиты)
fatal: нет описания метки?
Fatal: Tagging failed. Please run finish again to retry.
[sashubina@sashubina 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 | 9 ++++++++
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md
Ветка release/1.0.0 удалена (была 6f42435).

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'

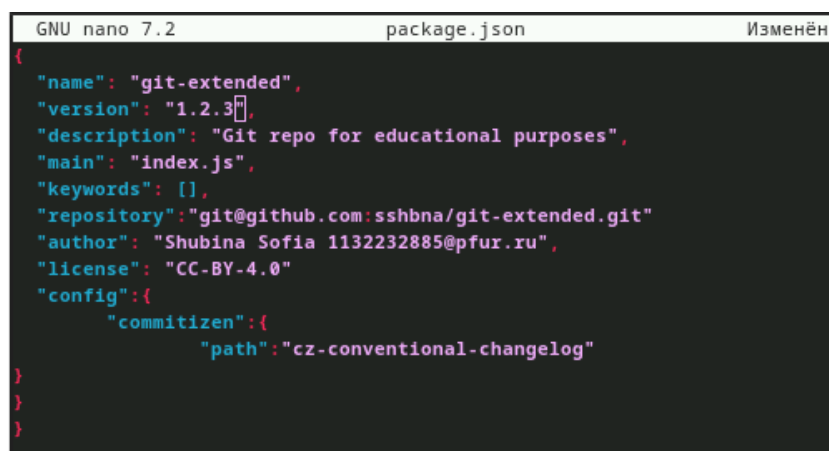
[sashubina@sashubina git-extended]$ git push --all
Перечисление объектов: 6, готово.
Подсчет объектов: 100% (6/6), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (5/5), 2.80 КиБ | 2.80 МиБ/с, готово.
Всего 5 (изменений 0), повторно использовано 0 (изменений 0), повторно и
спользовано пакетов 0
To github.com:sashubina/git-extended.git
   baacc87..ff941f4  develop -> develop
   baacc87..d3e008a  main -> main
[sashubina@sashubina 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 github.com:sashubina/git-extended.git
 * [new tag]         v1.0.0 -> v1.0.0
[sashubina@sashubina git-extended]$ 

```

Рис. 4.17: Зальем релизную ветку в основную ветку и отправим данные

Создадим релиз на github. Для этого будем использовать утилиты работы с github: `gh release create v1.0.0 -F CHANGELOG.md` Работа с репозиторием git Разработка новой функциональности Создадим ветку для новой функциональности: `git flow feature start feature_branch` Далее, продолжаем работу с git как обычно. По окончании разработки новой функциональности следующим шагом следует объединить ветку `feature_branch` с `develop`: `git flow feature finish feature_branch`

Создание релиза git-flow Создадим релиз с версией 1.2.3: `git flow release start 1.2.3` Обновите номер версии в файле `package.json`. Установите её в 1.2.3. (рис. 4.18).



```
GNU nano 7.2 package.json Изменён
{
  "name": "git-extended",
  "version": "1.2.3",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "keywords": [],
  "repository": "git@github.com:sshbna/git-extended.git",
  "author": "Shubina Sofia 1132232885@pfur.ru",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 4.18: Обновление номера версии

Создадим журнал изменений standard-changelog Добавим журнал изменений в индекс `git add CHANGELOG.md git commit -am 'chore(site): update changelog'`

Зальём релизную ветку в основную ветку

`git flow release finish 1.2.3`

Отправим данные на github

`git push --all git push --tags` Создадим релиз на github с комментарием из журнала изменений: `gh release create v1.2.3 -F CHANGELOG.md`

(рис. 4.19).

```

[sashubina@sashubina git-extended]$ git flow release finish 1.2.3
Fatal: Branch 'release/1.2.3' does not exist and is required.
[sashubina@sashubina git-extended]$ git push --all
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 2.77 КиБ | 2.77 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 github.com:sshbna/git-extended.git
   ff941f4..b768612  develop -> develop
   d3e008a..0b5b270  main -> main
[sashubina@sashubina 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 github.com:sshbna/git-extended.git
 * [new tag]          v1.2.3 -> v1.2.3
[sashubina@sashubina git-extended]$ gh release create v1.2.3 -F CHANGELO
G.md
https://github.com/sshbna/git-extended/releases/tag/v1.2.3
[sashubina@sashubina git-extended]$ 

```

Рис. 4.19: Создание журнала изменений, добавление в индекс, заливаем ветку в основную и отправляем данные

5 Выводы

Я получила правильные навыки работы с репозиториями git.

Список литературы

1. Learning the bash Shell: Unix Shell Programming. O'Reilly Media. 354 с.