

# **Отчет по выполнению Лабораторной работы №4**

**Дисциплина: Архитектура компьютеров и операционные системы**

Ефремова Полина Александровна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
4.1	Установка ПО. Установка git-flow, Node.js. Настройка Node.js. Общие- принятые коммиты . . . . .	8
4.2	Практический сценарий использования git. Создание репозитория. Работа с репозиторием. . . . .	10
<b>5</b>	<b>Выводы</b>	<b>16</b>
	<b>Список литературы</b>	<b>17</b>

# Список иллюстраций

4.1	Установка nodejs и npm . . . . .	8
4.2	Установка git-flow . . . . .	8
4.3	Настройка Node.js . . . . .	9
4.4	standard-changelog и commitizen . . . . .	9
4.5	git-extended . . . . .	10
4.6	Конфигурация коммитов . . . . .	11
4.7	Отправка изменений на github . . . . .	11
4.8	Часть 1 . . . . .	12
4.9	v 1.0.0 . . . . .	12
4.10	Залить релизную ветку . . . . .	13
4.11	GitHub . . . . .	13
4.12	Создадим релиз на github. . . . .	14
4.13	Релиз v1.2.3. . . . .	14
4.14	Изменение package.json . . . . .	14
4.15	журнал изменений . . . . .	15
4.16	v 1.2.3 stop . . . . .	15

## **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 Теоретическое введение

Gitflow Workflow опубликована и популяризована Винсентом Дриссенем. Gitflow Workflow предполагает выстраивание строгой модели ветвления с учётом выпуска проекта. Данная модель отлично подходит для организации рабочего процесса на основе релизов. Работа по модели Gitflow включает создание отдельной ветки для исправлений ошибок в рабочей среде.

Семантическое версионирование описывается в манифесте семантического версионирования. Кратко его можно описать следующим образом: Версия задаётся в виде кортежа МАЖОРНАЯ\_ВЕРСИЯ.МИНОРНАЯ\_ВЕРСИЯ.ПАТЧ. Номер версии следует увеличивать: МАЖОРНУЮ версию, когда сделаны обратно несовместимые изменения API. МИНОРНУЮ версию, когда вы добавляете новую функциональность, не нарушая обратной совместимости. ПАТЧ-версию, когда вы делаете обратно совместимые исправления. Дополнительные обозначения для предрелизных и билд-метаданных возможны как дополнения к МАЖОРНАЯ.МИНОРНАЯ.ПАТЧ формату.

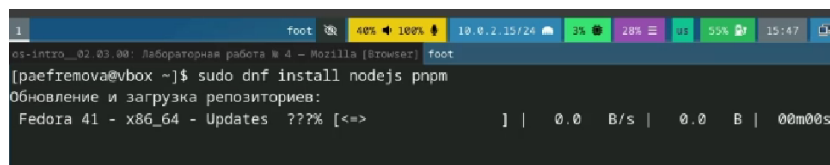
Спецификация Conventional Commits: Соглашение о том, как нужно писать сообщения commit'ов. Совместимо с SemVer. Даже вернее сказать, сильно связано с семантическим версионированием. Регламентирует структуру и основные типы коммитов.

## 4 Выполнение лабораторной работы

### 4.1 Установка ПО. Установка git-flow, Node.js.

#### Настройка Node.js. Общепринятые коммиты

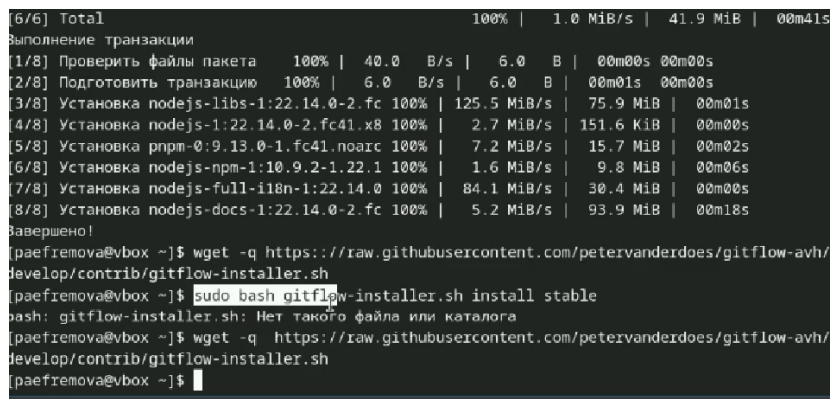
1. Устанавливаем nodejs и npm (рис. fig. 4.1).



```
1 foot 40% 100% 10.0.2.15/24 3% 28% 15:47
os:intro_02 03.00: лабораторная работа № 4 - Mozilla [Browser] foot
[paefremova@vbox ~]$ sudo dnf install nodejs npm
Обновление и загрузка репозитория:
Fedora 41 - x86_64 - Updates ???% [<=>] | 0.0 B/s | 0.0 B | 00m00s
```

Рис. 4.1: Установка nodejs и npm

2. Устанавливаем git-flow (рис. fig. 4.2)



```
[6/6] Total 100% | 1.0 MiB/s | 41.9 MiB | 00m41s
Выполнение транзакции
[1/8] Проверить файлы пакета 100% | 40.0 B/s | 6.0 B | 00m00s 00m00s
[2/8] Подготовить транзакцию 100% | 6.0 B/s | 6.0 B | 00m01s 00m00s
[3/8] Установка nodejs-libs-1:22.14.0-2.fc 100% | 125.5 MiB/s | 75.9 MiB | 00m01s
[4/8] Установка nodejs-1:22.14.0-2.fc41.x8 100% | 2.7 MiB/s | 151.6 KiB | 00m00s
[5/8] Установка npm-0:9.13.0-1.fc41.noar 100% | 7.2 MiB/s | 15.7 MiB | 00m02s
[6/8] Установка nodejs-npm-1:10.9.2-1.22.1 100% | 1.6 MiB/s | 9.8 MiB | 00m06s
[7/8] Установка nodejs-full-i18n-1:22.14.0 100% | 84.1 MiB/s | 30.4 MiB | 00m00s
[8/8] Установка nodejs-docs-1:22.14.0-2.fc 100% | 5.2 MiB/s | 93.9 MiB | 00m18s
Завершено!
[paefremova@vbox ~]$ wget -q https://raw.githubusercontent.com/petervanderdoes/gitflow-avh/
develop/contrib/gitflow-installer.sh
[paefremova@vbox ~]$ sudo bash gitflow-installer.sh install stable
bash: gitflow-installer.sh: Нет такого файла или каталога
[paefremova@vbox ~]$ wget -q https://raw.githubusercontent.com/petervanderdoes/gitflow-avh/
develop/contrib/gitflow-installer.sh
[paefremova@vbox ~]$
```

Рис. 4.2: Установка git-flow

3. Для работы с Node.js добавим каталог с исполняемыми файлами, устанавливаем yarn, в переменную PATH. (рис. fig. 4.3).



```
[paefremova@vbox ~]$ pnpm setup
Appended new lines to /home/paefremova/.bashrc

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

To start using pnpm, run:
source /home/paefremova/.bashrc
[paefremova@vbox ~]$ source ~/.bashrc
[paefremova@vbox ~]$
```

Рис. 4.3: Настройка Node.js

4. Настраиваем commitizen, standard-changelog. (рис. fig. 4.4).

```
[paefremova@vbox ~]$ pnpm add -g commitizen

Update available! 9.13.0 -> 10.5.2.
Changelog: https://github.com/pnpm/pnpm/releases/tag/v10.5.2
Run "pnpm self-update" to update.

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

WARN 2 deprecated subdependencies found: glob@7.2.3, inflight@1.0.6
Packages: +151
Progress: resolved 151, reused 0, downloaded 151, added 151, done

/home/paefremova/.local/share/pnpm/global/5:
+ commitizen 4.3.1

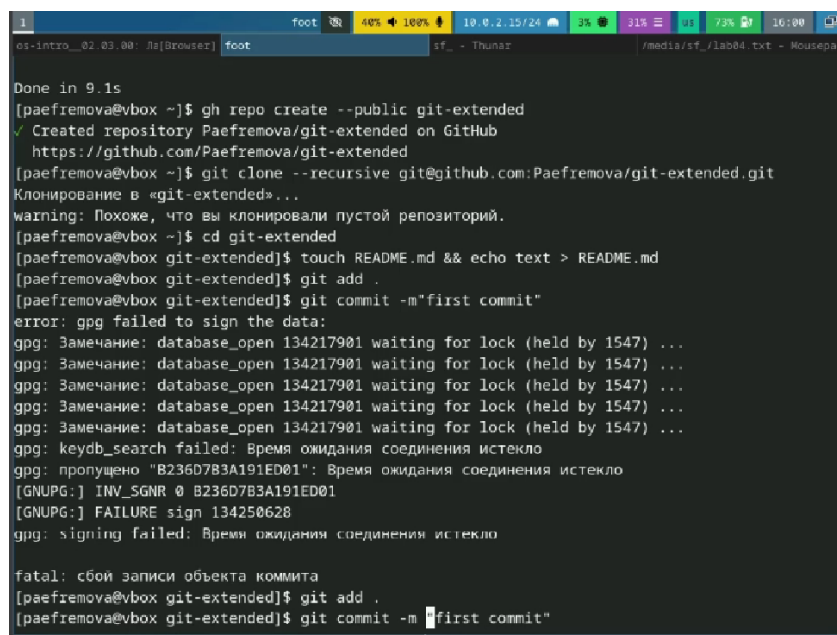
Done in 13.3s
[paefremova@vbox ~]$ pnpm add -g standard-shangelog
```

Рис. 4.4: standard-changelog и commitizen

## 4.2 Практический сценарий использования git.

### Создание репозитория. Работа с репозиторием.

5. Создаем репозиторий на GitHub. Для примера назовём его git-extended.  
Делаем первый коммит и выкладываем на github: (рис. fig. 4.5).



```
Done in 9.1s
[paefremova@vbox ~]$ gh repo create --public git-extended
✓ Created repository Paefremova/git-extended on GitHub
https://github.com/Paefremova/git-extended
[paefremova@vbox ~]$ git clone --recursive git@github.com:Paefremova/git-extended.git
Клонирование в «git-extended»...
warning: Похоже, что вы клонировали пустой репозиторий.
[paefremova@vbox ~]$ cd git-extended
[paefremova@vbox git-extended]$ touch README.md && echo text > README.md
[paefremova@vbox git-extended]$ git add .
[paefremova@vbox git-extended]$ git commit -m "first commit"
error: gpg failed to sign the data:
gpg: Замечание: database_open 134217901 waiting for lock (held by 1547) ...
gpg: Замечание: database_open 134217901 waiting for lock (held by 1547) ...
gpg: Замечание: database_open 134217901 waiting for lock (held by 1547) ...
gpg: Замечание: database_open 134217901 waiting for lock (held by 1547) ...
gpg: Замечание: database_open 134217901 waiting for lock (held by 1547) ...
gpg: keydb_search failed: Время ожидания соединения истекло
gpg: пропущено "B236D7B3A191ED01": Время ожидания соединения истекло
[GNUPG:] INV_SGNR 0 B236D7B3A191ED01
[GNUPG:] FAILURE sign 134250628
gpg: signing failed: Время ожидания соединения истекло

fatal: сбой записи объекта коммита
[paefremova@vbox git-extended]$ git add .
[paefremova@vbox git-extended]$ git commit -m "first commit"
```

Рис. 4.5: git-extended

6. Конфигурация общепринятых коммитов. Для этого добавим в файл package.json команду для формирования коммитов: (рис. fig. 4.6).

```
[paefremova@vbox git-extended]$ git push
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 880 байтов | 880.00 КиБ/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Paefremova/git-extended.git
 * [new branch]      main -> main
[paefremova@vbox git-extended]$ pnpm init
Wrote to /home/paefremova/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"
}
[paefremova@vbox git-extended]$ nano package.json
```

Рис. 4.6: Конфигурация коммитов

7. Добавляем файлы, выполняем коммит, отправляем на Github. (рис. fig. 4.7).

```
1 foot 48% 100% 10.0.2.15/24 2% 29% 0% 99% 19:25
[paefremova@vbox git-extended]$ git flow init
Fatal: Working tree contains unstaged changes. Aborting.
[paefremova@vbox git-extended]$ git add .
[paefremova@vbox git-extended]$ git cz
cz-cli@4.3.1, cz-conventional-changelog@3.3.0

? Select the type of change that you're committing: docs: Documentation only changes
? 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):
(27) improve information content
? 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 c455a69] docs: improve information content
1 file changed, 78 deletions(-)
delete mode 100755 gitflow-installer.sh
[paefremova@vbox git-extended]$ git push
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
Сжатие объектов: 100% (4/4), готово.
Запись объектов: 100% (4/4), 2.58 КиБ | 377.00 КиБ/с, готово.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Paefremova/git-extended.git
 60bc599..c455a69 main -> main
[paefremova@vbox git-extended]$ git flow init
```

Рис. 4.7: Отправка изменений на github

8. Инициализируем git-flow Префикс для ярлыков установим в v. Проверьте, что Вы на ветке develop: Загрузите весь репозиторий в хранилище: Установите внешнюю ветку как вышестоящую для этой ветки: (рис. fig. 4.8).

```
1 foot 48% 100% 10.0.2.15/24 3% 29% 99% 19:27
[paefremova@vbox 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/paefremova/git-extended/.git/hooks]
[paefremova@vbox git-extended]$ git branch
* develop
  main
[paefremova@vbox git-extended]$ git push --all
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/Paefremova/git-extended/pull/new/develop
remote:
to github.com:Paefremova/git-extended.git
* [new branch]      develop -> develop
[paefremova@vbox git-extended]$ git branch --set-upstream-to=origin/develop develop
```

Рис. 4.8: Часть 1

9. Создадим релиз с версией 1.0.0, Создадим журнал изменений Добавим журнал изменений в индекс Зальём релизную ветку в основную ветку (рис. fig. 4.9). (рис. fig. 4.10).

```
branch 'develop' set up to track 'origin/develop'.
[paefremova@vbox git-extended]$ git flow release start 1.0.0
Переключились на новую ветку «release/1.0.0»

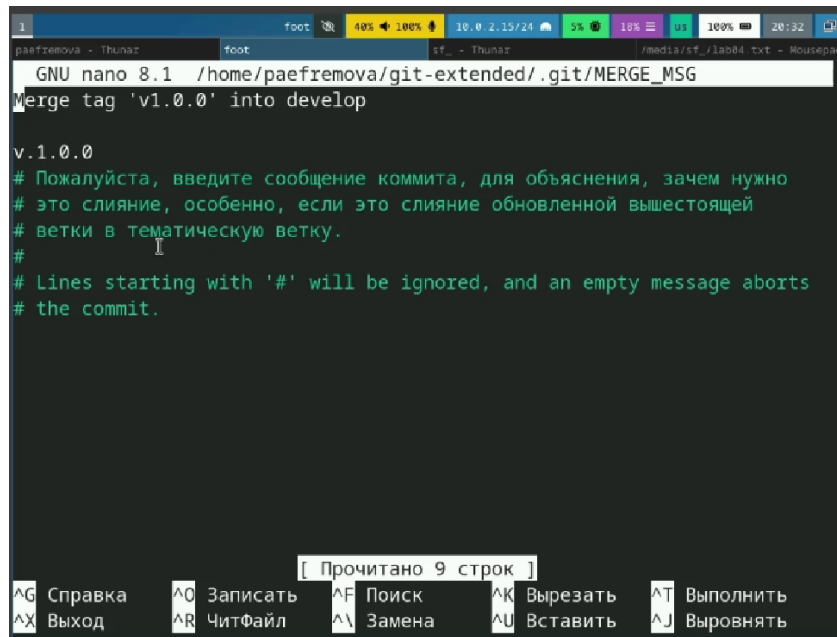
Summary of actions:
- A new branch 'release/1.0.0' was created, based on 'develop'
- You are now on branch 'release/1.0.0'

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'

[paefremova@vbox git-extended]$ standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
[paefremova@vbox git-extended]$ git add .
[paefremova@vbox git-extended]$ git cz chore site add changelog
```

Рис. 4.9: v 1.0.0

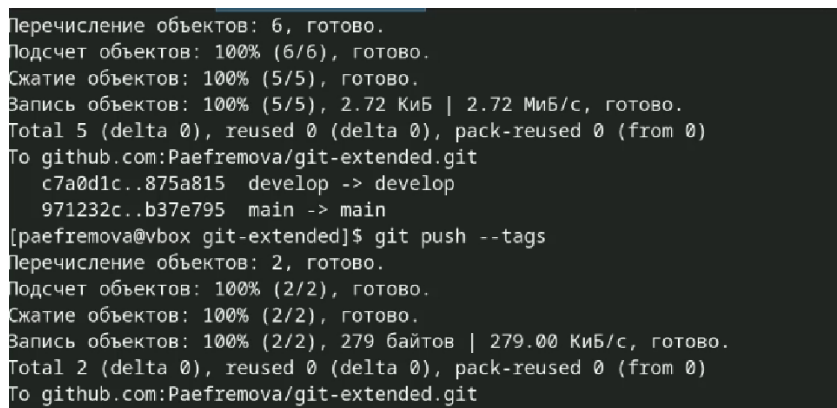


```
GNU nano 8.1 /home/paefremova/git-extended/.git/MERGE_MSG
Merge tag 'v1.0.0' into develop

v.1.0.0
# Пожалуйста, введите сообщение коммита, для объяснения, зачем нужно
# это слияние, особенно, если это слияние обновленной вышестоящей
# ветки в тематическую ветку.
#
# Lines starting with '#' will be ignored, and an empty message aborts
# the commit.
```

Рис. 4.10: Залить релизную ветку

10. Отправим данные на github (рис. fig. 4.11).



```
Перечисление объектов: 6, готово.
Подсчет объектов: 100% (6/6), готово.
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (5/5), 2.72 КиБ | 2.72 МБ/с, готово.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Paefremova/git-extended.git
   c7a0d1c..875a815  develop -> develop
   971232c..b37e795  main -> main
[paefremova@vbox git-extended]$ git push --tags
Перечисление объектов: 2, готово.
Подсчет объектов: 100% (2/2), готово.
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (2/2), 279 байтов | 279.00 КиБ/с, готово.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Paefremova/git-extended.git
```

Рис. 4.11: GitHub

11. Создадим релиз на github, Создадим ветку для новой функциональности. объединить ветку feature\_branch с develop. (рис. fig. 4.12).

```
Everything up-to-date
[paefremova@vbox git-extended]$ gh release create v1.0.0 -F CHANGELOG.md
https://github.com/Paefremova/git-extended/releases/tag/v1.0.0
[paefremova@vbox git-extended]$ git flow feature start feature_branch
Переключились на новую ветку «feature/feature_branch»
```

Рис. 4.12: Создадим релиз на github.

12. Создадим релиз с версией 1.2.3. Обновим номер версии в файле package.json. Установим её в 1.2.3. (рис. fig. 4.13). (рис. fig. 4.14).

```
[paefremova@vbox 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'

[paefremova@vbox git-extended]$ nano package.json > version 1.2.3
```

Рис. 4.13: Релиз v1.2.3.

```
package.json  [-M--] 21 L: [ 1+ 2 3/ 14] *(51 / 394b) 0034 0x022  [*][X]
{
  "name": "git-extended",
  "version": "1.2.3",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:Paefremova/git-extended.git",
  "author": "Polina Efremova polinaefremova68990@gmail.com",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 4.14: Изменение package.json

13. Создадим журнал изменений Добавим журнал изменений в индекс, зальем релизную ветку в основную ветку (рис. fig. 4.15).

```
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

[paefremova@vbox git-extended]$ git flow release start 1.2.3
Fatal: Tag 'v1.2.3' already exists. Pick another name.
[paefremova@vbox git-extended]$ nano package.json
[paefremova@vbox git-extended]$ standard-changelog
✓ output changes to CHANGELOG.md
[paefremova@vbox git-extended]$ git add .
[paefremova@vbox git-extended]$ git add CHANGELOG.md
[paefremova@vbox git-extended]$ git commit -am 'chore(site): update changelog'
[feature/feature_branch 5b571c4] chore(site): update changelog
2 files changed, 5 insertions(+), 1 deletion(-)
[paefremova@vbox git-extended]$ git flow release finish 1.2.3
```

Рис. 4.15: журнал изменений

14.Отправим данные на github Создадим релиз на github с комментарием из журнала изменений: (рис. fig. 4.16).

```
Перечисление объектов: 7, готово.
Подсчет объектов: 100% (7/7), готово.
Сжатие объектов: 100% (4/4), готово.
Запись объектов: 100% (4/4), 1.11 КиБ | 1.11 МБ/с, готово.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'feature/feature_branch' on GitHub by
visiting:
remote:   https://github.com/Paefremova/git-extended/pull/new/feature
/feature_branch
remote:
To github.com:Paefremova/git-extended.git
* [new branch]      feature/feature_branch -> feature/feature_branch
[paefremova@vbox git-extended]$ git push --tags
To github.com:Paefremova/git-extended.git
! [rejected]        v1.0.0 -> v1.0.0 (already exists)
error: не удалось отправить некоторые ссылки в «github.com:Paefremova/git-extended.git»
hint: Обновления были отклонены, так как метка уже существует во внешнем репозитории.
[paefremova@vbox git-extended]$ gh release create v1.2.3 -F CHANGELOG
```

Рис. 4.16: v 1.2.3 stop

## 5 Выводы

Мы получили навыки правильной работы с репозиториями git, в особенности с git-flow. Теперь я могу загружать данные через release.



# Список литературы

Рабочий процесс с Gitflow(электронный ресурс)