

Отчёта по лабораторной работе № 4

Операционные системы

Сахно Алёна Юрьевна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	10
4.1	Установка программного обеспечения	10
5	Практический сценарий использования git	14
6	Конфигурация git-flow	17
7	Выводы	23
	Список литературы	24

Список иллюстраций

Список таблиц

1 Цель работы

Получение навыков правильной работы с репозиториями git.

2 Задание

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

3 Теоретическое введение

Рабочий процесс Gitflow

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

Процесс работы с Gitflow Основные ветки (master) и ветки разработки (develop)

Для фиксации истории проекта в рамках этого процесса вместо одной ветки master используются две ветки. В ветке master хранится официальная история релиза, а ветка develop предназначена для объединения всех функций. Кроме того, для удобства рекомендуется присваивать всем коммитам в ветке master номер версии.

При использовании библиотеки расширений git-flow нужно инициализировать структуру в существующем репозитории:

```
git flow init
```

Для github параметр Version tag prefix следует установить в v.

После этого проверьте, на какой ветке Вы находитесь:

```
git branch
```

Функциональные ветки (feature)

Под каждую новую функцию должна быть отведена собственная ветка, которую можно отправлять в центральный репозиторий для создания резервной копии или совместной работы команды. Ветки feature создаются не на основе master,

а на основе develop. Когда работа над функцией завершается, соответствующая ветка сливается обратно с веткой develop. Функции не следует отправлять напрямую в ветку master. Как правило, ветки feature создаются на основе последней ветки develop.

Создание функциональной ветки

Создадим новую функциональную ветку:

```
git flow feature start feature_branch
```

Далее работаем как обычно.

Окончание работы с функциональной веткой

По завершении работы над функцией следует объединить ветку feature_branch с develop:

```
git flow feature finish feature_branch
```

Ветки выпуска (release)

Когда в ветке develop оказывается достаточно функций для выпуска, из ветки develop создаётся ветка release. Создание этой ветки запускает следующий цикл выпуска, и с этого момента новые функции добавить больше нельзя — допускается лишь отладка, создание документации и решение других задач. Когда подготовка релиза завершается, ветка release сливается с master и ей присваивается номер версии. После нужно выполнить слияние с веткой develop, в которой с момента создания ветки релиза могли возникнуть изменения. Благодаря тому, что для подготовки выпусков используется специальная ветка, одна команда может дорабатывать текущий выпуск, в то время как другая команда продолжает работу над функциями для следующего.

Создать новую ветку release можно с помощью следующей команды:

```
git flow release start 1.0.0
```

Для завершения работы на ветке release используются следующие команды:


```
git flow release finish 1.0.0
```

Ветки исправления (hotfix)

Ветки поддержки или ветки hotfix используются для быстрого внесения исправлений в рабочие релизы. Они создаются от ветки master. Это единственная ветка, которая должна быть создана непосредственно от master. Как только исправление завершено, ветку следует объединить с master и develop. Ветка master должна быть помечена обновлённым номером версии. Наличие специальной ветки для исправления ошибок позволяет команде решать проблемы, не прерывая остальную часть рабочего процесса и не ожидая следующего цикла релиза.

Ветку hotfix можно создать с помощью следующих команд:

```
git flow hotfix start hotfix_branch
```

По завершении работы ветка hotfix объединяется с master и develop:

```
git flow hotfix finish hotfix_branch
```

4 Выполнение лабораторной работы

4.1 Установка программного обеспечения

Установка git-flow

Linux

Fedora

Установка из коллекции репозитория Copr

```
foot os-intro__02.03.00: Лабораторная раб
[aysakhno@aysakhno ~]$ sudo -i
[sudo] пароль для aysakhno:
[root@aysakhno ~]# sudo dnf copr enable elegos/gitflow
https://copr.fedorainfracloud.org/api_3/rp 100% | 876.0  B/s | 652.0  B | 00m01s
Включение репозитория Copr. Обратите внимание, что этот репозиторий не является частью
основного дистрибутива, и его качество может отличаться.

Проект Fedora не имеет никакой власти над содержимым
за пределами правил, изложенных в Copr FAQ по адресу
<https://docs.pagure.org/copr.copr/user_documentation.html#what-i-can-build-in-copr>,
и пакеты не имеют никаких требований к качеству или уровню безопасности.

Пожалуйста, не публикуйте сообщения об ошибках, связанных с этими пакетами, в Fedora
Bugzilla. В случае возникновения проблем обращайтесь к владельцу этого репозитория.
Is this ok [y/N]: y
```

(рис.1 ??).

```
[root@aysakhno ~]# sudo dnf install gitflow
Обновление и загрузка репозитория:
Copr repo for gitflow owned by elegos 100% | 2.4 KiB/s | 2.4 KiB | 00m01s
Репозитории загружены.
Пакет Арх. Версия Репозиторий Размер
Установка:
gitflow x86_64 1.12.3-1.fc34 copr:copr.fedora 261.7 KiB

Сводка транзакции:
Установка: 1 пакета

Общий размер входящих пакетов составляет 57 KiB. Необходимо загрузить 57 KiB.
После этой операции будут использоваться дополнительные 262 KiB (установка 262 KiB, удаление 0 B).
Is this ok [y/N]: y
[1/1] gitflow-0:1.12.3-1.fc34.x86_64 100% | 73.1 KiB/s | 57.4 KiB | 00m01s
-----
[1/1] Total 100% | 72.7 KiB/s | 57.4 KiB | 00m01s
[1/2] https://download.copr.fedorainfracloud.org/results/elegos/gitflow/pubkey.gpg 777% | 0.0 B/s | 0.0 B | 00m00s
[1/2] https://download.copr.fedorainfracloud.org/results/elegos/gitflow/pubkey.gpg 100% | 5.5 KiB/s | 998.0 B | 00m00s
-----
[2/2] Total 100% | 72.7 KiB/s | 57.4 KiB | 00m01s

Импорт ключа OpenPGP 0x80F63AA3:
UserID : "elegos_gitflow (None) <elegos@gitflow@copr.fedorahosted.org>"
Отпечаток: 935799B049C5C36CF39A22823CA8E680F63AA3
Имя : https://download.copr.fedorainfracloud.org/results/elegos/gitflow/pubkey.gpg
Is this ok [y/N]: y
Ключ был успешно импортирован.
[1/3] Проверить файлы пакетов 100% | 111.0 B/s | 1.0 B | 00m00s
[2/3] Подготовить транзакцию 100% | 1.0 B/s | 1.0 B | 00m01s
[3/3] Установка gitflow-0:1.12.3-1.fc34.x86_64 100% | 489.2 KiB/s | 271.0 KiB | 00m01s
```

(рис.2 ??).

Установка Node.js

На Node.js базируется программное обеспечение для семантического версионирования и общепринятых коммитов.

Fedora

```
[root@aysakhno ~]# dnf install nodejs
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет Арх. Версия Репозиторий Размер
Установка:
nodejs x86_64 1:22.14.0-2.fc41 updates 149.5 KiB
Установка зависимостей:
nodejs-libs x86_64 1:22.14.0-2.fc41 updates 75.9 MiB
Установка слабых зависимостей:
nodejs-docs noarch 1:22.14.0-2.fc41 updates 93.7 MiB
nodejs-full-i18n x86_64 1:22.14.0-2.fc41 updates 30.4 MiB
nodejs-npm x86_64 1:10.9.2-1.22.14.0-2.fc41 updates 9.3 MiB

Сводка транзакции:
Установка: 5 пакетов

Общий размер входящих пакетов составляет 39 MiB. Необходимо загрузить 39 MiB.
После этой операции будут использоваться дополнительные 210 MiB (установка 210 MiB, удаление 0 B).
Is this ok [y/N]: y
[1/5] nodejs-1:22.14.0-2.fc41.x86_64 100% | 141.6 KiB/s | 46.7 KiB | 00m00s
[2/5] nodejs-npm-1:10.9.2-1.22.14.0-2.fc41.x86_64 100% | 306.6 KiB/s | 2.4 MiB | 00m08s
[3/5] nodejs-docs-1:22.14.0-2.fc41.noarch 100% | 421.7 KiB/s | 8.8 MiB | 00m21s
[4/5] nodejs-full-i18n-1:22.14.0-2.fc41.x86_64 100% | 238.9 KiB/s | 8.5 MiB | 00m37s
[5/5] nodejs-libs-1:22.14.0-2.fc41.x86_64 100% | 284.6 KiB/s | 19.2 MiB | 01m09s
-----
[5/5] Total 100% | 567.7 KiB/s | 39.0 MiB | 01m10s

Выполнение транзакции
[1/7] Проверить файлы пакетов 100% | 11.0 B/s | 5.0 B | 00m00s
[2/7] Подготовить транзакцию 100% | 2.0 B/s | 5.0 B | 00m02s
```

(рис.3 ??).

```
[root@aysakhno ~]# dnf install pnpm
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет Арх. Версия Репозиторий Размер
Установка:
  pnpm noarch 9.13.0-1.fc41 updates 15.5 MiB

Сводка транзакции:
Установка: 1 пакета

Общий размер входящих пакетов составляет 3 MiB. Необходимо загрузить 3 MiB.
После этой операции будут использоваться дополнительные 16 MiB (установка 16 MiB, удаление 0 B).
Is this ok [y/N]: y
[1/1] pnpm-9.13.0-1.fc41.noarch 100% | 1.3 MiB/s | 3.0 MiB | 00m02s
-----
[1/1] Total 100% | 1.0 MiB/s | 3.0 MiB | 00m03s
Выполнение транзакции
[1/3] Проверить файлы пак100% | 22.0 B/s | 1.0 B | 00m00s
[2/3] Подготовить транзак 100% | 1.0 B/s | 1.0 B | 00m01s
[3/3] Установка pnpm-9.13.0-1.fc 100% | 7.6 MiB/s | 15.7 MiB | 00m02s
Завершено!
```

(рис.4 ??).

Настройка Node.js

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

Запустите:

```
[root@aysakhno ~]# 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
```

(рис.5 ??).

Перелогиньтесь, или выполните:

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

/root/.local/share/pnpm/global/5:
+ commitizen 4.3.1

Done in 22.3s
[root@aysakhno ~]# pnpm add -g standard-changelog
WARN GET https://registry.npmjs.org/picocolors error (ETIMEDOUT). Will retry in 10 seconds. 2 retries left.
WARN 2 deprecated subdependencies found: glob@7.2.3, inflight@1.0.6
Packages: +39
Progress: resolved 190, reused 151, downloaded 39, added 39, done

/root/.local/share/pnpm/global/5:
+ standard-changelog 0.0.0

Done in 28.2s
```

(рис.6 ??).

Общепринятые коммиты

comitizen

Данная программа используется для помощи в форматировании коммитов.

```
pnpm add -g comitizen
```

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

standard-changelog

Данная программа используется для помощи в создании логов.

```
pnpm add -g standard-changelog
```

5 Практический сценарий

использования git

Создание репозитория git

Подключение репозитория к github

Создайте репозиторий на GitHub. Для примера назовём его git-extended.

Делаем первый коммит и выкладываем на github:

```
[root@aysakhno git-extended]# git commit -m "first commit"
текущая ветка: main
эта ветка соответствует «origin/main».
```

(рис.7 ??).

ничего коммитить, нет изменений в рабочем каталоге

```
[root@aysakhno git-extended]# git branch
* main

[root@aysakhno ~]# mc

[root@aysakhno git-extended]# git push -u origin main
branch 'main' set up to track 'origin/main'.
Everything up-to-date

[root@aysakhno git-extended]# git checkout -b master
Переключились на новую ветку «master»

[root@aysakhno git-extended]# git push -u origin master
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/Asakhn0/git-extended/pull/new/master
remote:
To github.com:Asakhn0/git-extended.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

[root@aysakhno ~]# mc

[root@aysakhno git-extended]# git branch -r
origin/HEAD -> origin/main
origin/main
origin/master
```

(рис.8 ??).

Конфигурация общепринятых коммитов

Конфигурация для пакетов Node.js

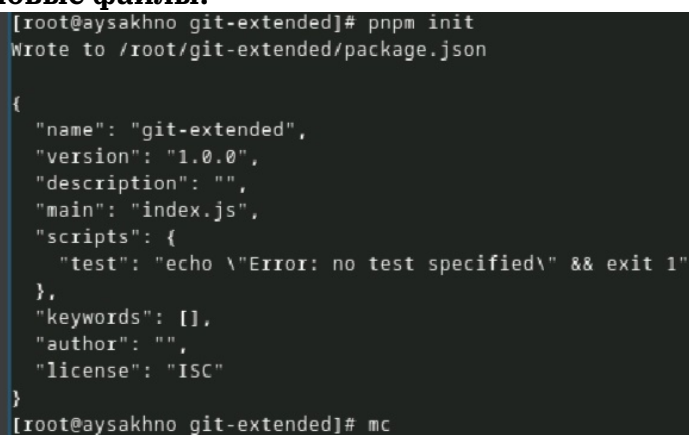
Сконфигурируем формат коммитов. Для этого добавим в файл `package.json` команду для формирования коммитов:

```
"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" } } }
```

Добавим новые файлы:

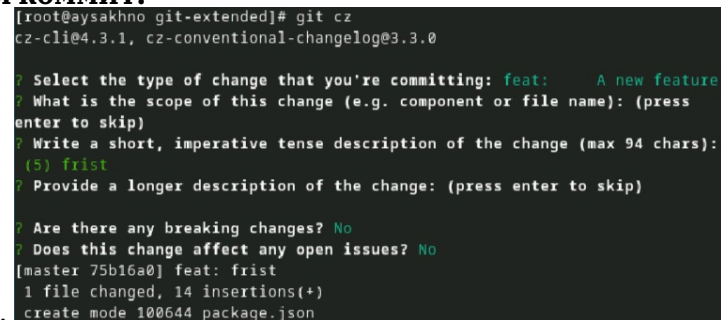


```
[root@aysakhno git-extended]# npm init
Wrote to /root/git-extended/package.json

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

(рис.9 ??). [root@aysakhno git-extended]# git add .

Выполним коммит:



```
[root@aysakhno git-extended]# git cz
cz-cli@4.3.1, 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):
(5) frist
? 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
[master 75b16a0] feat: frist
1 file changed, 14 insertions(+)
create mode 100644 package.json
```

(рис.10 ??).

Отправим на github:

```
[root@aysakhno git-extended]# git push
Перечисление объектов: 4, готово.
Подсчет объектов: 100% (4/4), готово.
При сжатии изменений используется до 10 потоков
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 1.14 КиБ | 1.14 МБ/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Asakhno/git-extended.git
   b6ec844..75b16a0 master -> master
[root@aysakhno git-extended]# git flow init

Which branch should be used for integration of the "next release"?
- main
Branch name for "next release" development: [] develop
Fatal: Local branch 'develop' does not exist.
```

(рис.11 ??).

6 Конфигурация git-flow

Инициализируем git-flow

```
git flow init
```

Префикс для ярлыков установим в v.

Проверьте, что Вы на ветке develop:

```
git branch
```

Загрузите весь репозиторий в хранилище:

```
git push --all
```

Установите 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
```

```

[root@aysakhno git-extended]# git flow init

Which branch should be used for integration of the "next release"?
- develop
- main
Branch name for "next release" development: [develop] 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@aysakhno git-extended]# git branch
* develop
  main
  master
[root@aysakhno git-extended]# git push --all
Everything up-to-date
[root@aysakhno git-extended]# git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
[root@aysakhno 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'

```

(рис.12 ??).

Создадим журнал изменений

```
standard-changelog --first-release
```

Добавим журнал изменений в индекс

```
git add CHANGELOG.md
```

```
git commit -am 'chore(site): add changelog'
```

Зальём релизную ветку в основную ветку

```
git flow release finish 1.0.0
```

Отправим данные на github

```
git push --all
```

```
git push --tags
```

```

root@aysakhno: /thunar
[root@aysakhno git-extended]# standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
[root@aysakhno git-extended]# git add CHANGELOG.md
[root@aysakhno git-extended]# git commit -am 'chore(site): add changelog'
[release/1.0.0 1798deb] chore(site): add changelog
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md
[root@aysakhno git-extended]# git flow release finish 1.0.0
Переключились на ветку «master»
Эта ветка соответствует «origin/master».
Merge made by the 'ort' strategy.
CHANGELOG.md | 9 ++++++++
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md
Уже на «master»
Ваша ветка опережает «origin/master» 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 1798deb).

Summary of actions:
- Release branch 'release/1.0.0' has been merged into 'master'
- 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@aysakhno git-extended]# git push --all
Перечисление объектов: 6, готово.
Подсчет объектов: 100% (6/6), готово.
При сжатии изменений используется до 10 потоков
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (5/5), 3.01 КиБ | 308.00 КиБ/с, готово.

```

(рис.13 ??).

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
```

```

[root@aysakhno git-extended]# git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 230 байтов | 76.00 КиБ/с, готово.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Asakhno/git-extended.git
 * [new tag]         v1.0.0 -> v1.0.0
[root@aysakhno git-extended]# gh release create v1.0.0 -F CHANGELOG.md
https://github.com/Asakhno/git-extended/releases/tag/v1.0.0
[root@aysakhno 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

[root@aysakhno 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 e52eed6).

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.14 ??).

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.

```

package.json  [-----] 21 L: [ 1+ 2 3/ 14] *(51 / 389b) 0034 0x022

  "name": "git-extended",
  "version": "1.2.3",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:Asakhno/git-extended.git",
  "author": "Alyona Sakhno <saxno.06@icloud.com>",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}

```

(рис.15 ??).

Создадим журнал изменений

```
standard-changelog
```

```

[root@aysakhno git-extended]# it flow release start 1.2.3
-bash: it: команда не найдена
[root@aysakhno 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@aysakhno git-extended]# mc

[root@aysakhno git-extended]# standard-changelog
✓ output changes to CHANGELOG.md
[root@aysakhno git-extended]# git add CHANGELOG.md
[root@aysakhno git-extended]# git commit -am 'chore(site): update changelog'
[release/1.2.3 39213db] chore(site): update changelog
2 files changed, 5 insertions(+), 1 deletion(-)

```

(рис.16 ??).

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
```

```

[root@aysakhno git-extended]# git flow release finish 1.2.3
Переключились на ветку «master»
Эта ветка соответствует «origin/master».
Merge made by the 'ort' strategy.
  CHANGELOG.md | 4 +++
  package.json | 2 +-
  2 files changed, 5 insertions(+), 1 deletion(-)
Уже на «master»
Ваша ветка опережает «origin/master» 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 39213db).

Summary of actions:
- Release branch 'release/1.2.3' has been merged into 'master'
- 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@aysakhno git-extended]# git push --all
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 10 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 2.99 Киб | 2.99 Миб/с, готово.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:Asakhno/git-extended.git
 e52eed6..e33992b develop -> develop
 a0ba75e..8af6c09 master -> master

```

(рис.17 ??).

Создадим релиз на github с комментарием из журнала изменений:

gh release create v1.2.3 -F CHANGELOG.md

```

[root@aysakhno git-extended]# git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 368 байтов | 368.00 КиБ/с, готово.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Asakhno/git-extended.git
 * [new tag]         v1.2.3 -> v1.2.3
[root@aysakhno git-extended]# gh release create v1.2.3 -F CHANGELOG.md
https://github.com/Asakhno/git-extended/releases/tag/v1.2.3

```

(рис.18 ??).

7 Выводы

Я получила навыков правильной работы с репозиториями git.

Список литературы

::: {#refs} <https://esystem.rudn.ru/mod/page/view.php?id=1224375> :::