

Отчёт по лабораторной работе № 4

Веретенников Дмитрий Олегович НКАбд-01-24

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	10
5	Выводы	19
	Список литературы	20

Список иллюстраций

4.1	Процесс подготовки к установке	10
4.2	Процесс установки	11
4.3	Процесс установки и настройки	11
4.4	Программа	12
4.5	Программа	12
4.6	Отправка коммита	13
4.7	Исправленный файл package.json	13
4.8	Выполнение коммита	14
4.9	Выполнение коммита	14
4.10	Загрузка репозитория, создание релиза	15
4.11	Создание журнала, добавление журнала, заливка релизной ветки	15
4.12	Создаем и объединяем ветки	16
4.13	Обновленный номер версии в файле	16
4.14	Создание релиза, создание журнала	17
4.15	Создание релиза, создание журнала	17
4.16	Отправка данных на Github	18

Список таблиц

1 Цель работы

Получение навыков правильной работы с репозиториями git.

2 Задание

1. Установка git-flow
2. Установка Node.js
3. Настройка Node.js
4. Общепринятые коммиты
5. Создание репозитория git
6. Работа с репозиторием git

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. Последовательность действий при работе по модели Gitflow: Из ветки master создаётся ветка develop. Из ветки develop создаётся ветка release. Из ветки develop создаются ветки feature. Когда работа над веткой feature завершена, она сливается с веткой develop. Когда работа над веткой релиза release завершена, она сливается в ветки develop и master. Если в master обнаружена проблема, из master создаётся ветка hotfix. Когда работа над веткой исправления hotfix завершена, она сливается в ветки develop и master.

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 ис-

правление 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

Устанавливаем git-flow из коллекции репозитория Copr (рис. 4.1).

```
foot
[doveretennikov@vbox ~]$ sudo -i
[root@vbox ~]# # Enable the copr repository
[root@vbox ~]# dnf copr enable elegos/gitflow
https://copr.fedorainfracloud.org/api_3/ipmrepo/elegos/gitflow/fedora-41/ 100% | 243.0 B/s | 652.0 B | 00m03s
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
[root@vbox ~]#
[root@vbox ~]# dnf copr enable elegos/gitflow
https://copr.fedorainfracloud.org/api_3/ipmrepo/elegos/gitflow/fedora-41/ 100% | 302.0 B/s | 652.0 B | 00m02s
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
[root@vbox ~]# # Install gitflow
[root@vbox ~]#
```

Рис. 4.1: Процесс подготовки к установке

Установка git-flow (рис. 4.2).

```

foot
Установка:
  gitflow                                x86_64 1.12.3-1.fc34                copr:copr.fedorainfracloud.org: 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% | 45.0 KiB/s | 57.4 KiB | 00m01s
-----
[1/1] Total                                          100% | 44.9 KiB/s | 57.4 KiB | 00m01s
[1/2] https://download.copr.fedorainfracloud.org/resul ???% [ <=> ] | 0.0 B/s | 0.0 B | 00m00s
[1/2] https://download.copr.fedorainfracloud.org/resul ???% [ <=> ] | 0.0 B/s | 0.0 B | 00m00s
[1/2] https://download.copr.fedorainfracloud.org/resul ???% [ <=> ] | 0.0 B/s | 0.0 B | 00m00s
[1/2] https://download.copr.fedorainfracloud.org/resul ???% [ <=> ] | 0.0 B/s | 0.0 B | 00m01s
[1/2] https://download.copr.fedorainfracloud.org/resul ???% [ <=> ] | 0.0 B/s | 0.0 B | 00m01s
[1/2] https://download.copr.fedorainfracloud.org/results/elegos/gitflow/pub 100% | 665.0 B/s | 998.0 B | 00m02s
-----
[2/2] Total                                          100% | 44.9 KiB/s | 57.4 KiB | 00m01s
Importing OpenPGP key 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 пакета                100% | 250.0 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% | 144.2 KiB/s | 271.0 KiB | 00m02s
Завершено!
[root@vbox ~]#

```

Рис. 4.2: Процесс установки

Устанавливаем Nodejs и npm, далее настраиваем Nodejs (рис. 4.3).

```

foot
Установка:
  npm                                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] npm-0:9.13.0-1.fc41.noarch                100% | 153.4 KiB/s | 3.0 MiB | 00m20s
-----
[1/1] Total                                          100% | 136.2 KiB/s | 3.0 MiB | 00m22s
Выполнение транзакции
[1/3] Проверить файлы пакета                100% | 45.0 B/s | 1.0 B | 00m00s
[2/3] Подготовить транзакцию                100% | 1.0 B/s | 1.0 B | 00m01s
[3/3] Установка npm-0:9.13.0-1.fc41.noarch 100% | 3.5 MiB/s | 15.7 MiB | 00m04s
Завершено!
[doveretennikov@vbox ~]$ npm setup
Appended new lines to /home/doveretennikov/.bashrc

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

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

```

Рис. 4.3: Процесс установки и настройки

Воспользуемся программой для помощи в форматировании коммитов (рис. 4.4).

```
foot
export PNPM_HOME="/home/doveretennikov/.local/share/pnpm"
case "::$PATH:" in
  *"$PNPM_HOME:") ;;
  *) export PATH="$PNPM_HOME:$PATH" ;;
esac

To start using pnpm, run:
source /home/doveretennikov/.bashrc
[doveretennikov@vbox ~]$ source ~/.bashrc
[doveretennikov@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/doveretennikov/.local/share/pnpm/global/5:
+ commitizen 4.3.1

Done in 31.5s
[doveretennikov@vbox ~]$
```

Рис. 4.4: Программа

Воспользуемся программой для помощи в создании логов (рис. 4.5).

```
foot

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

Done in 31.5s
[doveretennikov@vbox ~]$ pnpm add -g standard-changelog
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

/home/doveretennikov/.local/share/pnpm/global/5:
+ standard-changelog 6.0.0

Done in 8.7s
[doveretennikov@vbox ~]$
```

Рис. 4.5: Программа

Создаем репозиторий, делаем первый коммит и выкладываем на гитхаб (рис. 4.6).

```
foot
[root@vbox git-extended]# cd git-extended
[root@vbox git-extended]# git commit -m "first commit"
Текущая ветка: main

Начальный коммит

нечего коммитить (создайте/скопируйте файлы, затем запустите
«git add», чтобы отслеживать их)
[root@vbox git-extended]# git remote add origin git@github.com:doveretennikov/git-extended.git
error: внешний репозиторий origin уже существует
[root@vbox git-extended]# git push -u origin master
error: src refspec master ничему не соответствует
error: не удалось отправить некоторые ссылки в «github.com:doveretennikov/git-extended.git»
[root@vbox git-extended]# npm init
Wrote to /root/git-extended/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@vbox git-extended]# mc
[root@vbox git-extended]# git add .
```

Рис. 4.6: Отправка коммита

Сконфигурируем формат коммитов. Для этого добавим в файл package.json команду для формирования коммитов (рис. 4.7).

```
mc [root@vbox]:~/git-extended/git-extended
/root/git-extended/git-extended/package.json 397/397 100%
{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:doveretennikov/git-extended.git",
  "author": "doveretennikov <domiswmm2018@gmail.com>",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 4.7: Исправленный файл package.json

Добавим новые файлы, выполним коммит, отправим на github (рис. 4.8).

```
foot
No authorization header was set for the request.
Progress: resolved 1, reused 1, downloaded 0, added 0
[root@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: 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) test
? Provide a longer description of the change: (press enter to skip)

? Are there any breaking changes? Yes
? A BREAKING CHANGE commit requires a body. Please enter a longer description of the commit itself:
n
? Describe the breaking changes:
n
? Does this change affect any open issues? No
[main (корневой коммит) e90194c] feat: test
1 file changed, 14 insertions(+)
create mode 100644 package.json
[root@vbox git-extended]# git push
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 1.09 КиБ | 1.09 МБ/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:doveretennikov/git-extended.git
 * [new branch]      main -> main
[root@vbox git-extended]#
```

Рис. 4.8: Выполнение коммита

Инициализируем git-flow (рис. 4.9).

```
foot
create mode 100644 package.json
[root@vbox git-extended]# git push
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 1.09 КиБ | 1.09 МБ/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:doveretennikov/git-extended.git
 * [new branch]      main -> main
[root@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? []
Hooks and filters directory? [/root/git-extended/git-extended/.git/hooks]
[root@vbox git-extended]# mc

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

Рис. 4.9: Выполнение коммита

Загружаем весь репозиторий в хранилище, установим внешнюю ветку как вышестоящую для этой ветки, создадим релиз с версией 1.0.0 (рис. 4.10).

```
foot
Hooks and filters directory? [/root/git-extended/git-extended/.git/hooks]
[root@vbox git-extended]# mc

[root@vbox git-extended]# git branch
* develop
  main
[root@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/doveretennikov/git-extended/pull/new/develop
remote:
To github.com:doveretennikov/git-extended.git
 * [new branch]      develop -> develop
[root@vbox git-extended]# git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
[root@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'

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

Рис. 4.10: Загрузка репозитория, создание релиза

Создаем журнал изменений, добавляем журнал изменений в индекс, заливаем релизную ветку в основную ветку (рис. 4.11).

```
foot
✓ output changes to CHANGELOG.md
[root@vbox git-extended]# git add CHANGELOG.md
[root@vbox git-extended]# git commit -am 'chore(site): add changelog'
[release/1.0.0 debc61e] chore(site): add changelog
1 file changed, 14 insertions(+)
create mode 100644 CHANGELOG.md
[root@vbox 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 | 14 ++++++
1 file changed, 14 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)
fatal: нет описания метки?
Fatal: Tagging failed. Please run finish again to retry.
[root@vbox git-extended]# git push --all
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), 1.94 КиБ | 1.94 МБ/с, готово.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:doveretennikov/git-extended.git
 e90194c..c622a19  main -> main
 * [new branch]      release/1.0.0 -> release/1.0.0
[root@vbox git-extended]# git push --tags
Everything up-to-date
[root@vbox git-extended]#
```

Рис. 4.11: Создание журнала, добавление журнала, заливка релизной ветки

Создаем ветку для новой функциональности, по окончании разработки новой

функциональности объединяем ветку feature_branch с develop (рис. 4.12).

```
foot
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:doveretennikov/git-extended.git
   e90194c..c622a19  main -> main
* [new branch]      release/1.0.0 -> release/1.0.0
[root@vbox git-extended]# git push --tags
Everything up-to-date
[root@vbox git-extended]# gh release create v1.0.0 -F CHANGELOG.md
https://github.com/doveretennikov/git-extended/releases/tag/v1.0.0
[root@vbox 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@vbox 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 e90194c).

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@vbox git-extended]#
```

Рис. 4.12: Создаем и объединяем ветки

Обновите номер версии в файле package.json. Установите её в 1.2.3 (рис. 4.13).

```
mc [root@vbox]:~/git-extended/git-extended
root/git-extended/git-extended/package.json 397/397 100%
{
  "name": "git-extended",
  "version": "1.2.3",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:doveretennikov/git-extended.git",
  "author": "doveretennikov <domiswmm2018@gmail.com>",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 4.13: Обновленный номер версии в файле

Создадим релиз с версией 1.2.3, создадим журнал изменений (рис. 4.14).


```
foot
git flow feature finish feature_branch

[root@vbox 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 e90194c).

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@vbox git-extended]# git flow release start 1.2.3
Fatal: There is an existing release branch '1.0.0'. Finish that one first.
[root@vbox git-extended]# mc

[root@vbox git-extended]# git flow release start 1.2.3
Fatal: There is an existing release branch '1.0.0'. Finish that one first.
[root@vbox git-extended]# mc

[root@vbox git-extended]# standard-changelog
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
[root@vbox git-extended]# git add CHANGELOG.md
[root@vbox git-extended]# git commit -am 'chore(site): update changelog'
[develop fec70e] chore(site): update changelog
2 files changed, 16 insertions(+), 1 deletion(-)
create mode 100644 CHANGELOG.md
[root@vbox git-extended]# git flow release finish 1.2.3
Fatal: Branch 'release/1.2.3' does not exist and is required.
```

Рис. 4.14: Создание релиза, создание журнала

Добавляем журнал изменений в индекс, заливаем релизную ветку в основную ветку (рис. 4.15).

```
foot
[root@vbox git-extended]# git flow release start 1.2.3
Branches 'develop' and 'origin/develop' have diverged.
And local branch 'develop' is ahead of 'origin/develop'.
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@vbox git-extended]# standard-changelog
✓ output changes to CHANGELOG.md
[root@vbox git-extended]# git add CHANGELOG.md
[root@vbox git-extended]# git commit -am 'chore(site): update changelog'
[release/1.2.3 17584f8] chore(site): update changelog
1 file changed, 4 insertions(+)
[root@vbox git-extended]# git flow release finish 1.2.3
Branches 'develop' and 'origin/develop' have diverged.
And local branch 'develop' is ahead of 'origin/develop'.
Переключились на ветку «main»
Эта ветка соответствует «origin/main».
Merge made by the 'ort' strategy.
CHANGELOG.md | 12 ++++++
package.json | 2 +-
2 files changed, 13 insertions(+), 1 deletion(-)
```

Рис. 4.15: Создание релиза, создание журнала

Отправляем данные на Github (рис. 4.16).

```

foot
And local branch 'develop' is ahead of 'origin/develop'.
Уже на «main»
Ваша ветка опережает «origin/main» на 4 коммита.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
Переключились на ветку «develop»
Ваша ветка опережает «origin/develop» на 3 коммита.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
Merge made by the 'ort' strategy.
CHANGELOG.md | 4 ++++
1 file changed, 4 insertions(+)
Ветка release/1.2.3 удалена (была 17584f8).

Summary of actions:
- Release branch 'release/1.2.3' has been merged into 'main'
- The release was tagged '1.2.3'
- Release tag '1.2.3' has been back-merged into 'develop'
- Release branch 'release/1.2.3' has been locally deleted
- You are now on branch 'develop'

[root@vbox git-extended]# git push --all
Перечисление объектов: 12, готово.
Подсчет объектов: 100% (12/12), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (8/8), готово.
Запись объектов: 100% (8/8), 3.79 Киб | 3.79 Миб/с, готово.
Total 8 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To github.com:doveretennikov/git-extended.git
   fec70e..b544a30  develop -> develop
   c622a19..69403b3  main -> main
[root@vbox git-extended]#

```

Рис. 4.16: Отправка данных на Github

5 Выводы

После выполнения данной лабораторной работы я получил навыки правильной работы с репозиториями git.

Список литературы

Курс Архитектура компьютеров и операционные системы. Раздел “Операционные системы” Лабораторная работа № 4 ТУИС РУДН.