

Отчёт по лабораторной работе №4

Операционные системы

Луангсуваннавонг Сайпхачан

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
4.1	Установка git-flow	8
4.2	Установка Node.js	8
4.2.1	Настройка Node.js	9
4.3	Создание репозитория git	11
4.4	Работа с репозиторием git	17
5	Выводы	20
	Список литературы	21

Список иллюстраций

4.1	Включение репозитория сорп	8
4.2	Установка Git-flow	8
4.3	Установка Nodejs	9
4.4	Установка npnm	9
4.5	Установка пути в системе	9
4.6	Установка источника в системный файл bashrc	10
4.7	Установка commitizen	10
4.8	Установка standard-changelog	10
4.9	Создание нового репозитория	11
4.10	Создание нового каталога	11
4.11	Выполнение первого коммита	12
4.12	Созданный репозиторий	12
4.13	Конфигурация для пакетов Node.js	13
4.14	Редактирование файла package.json	13
4.15	Добавление файла на GitHub	13
4.16	Выполнение фиксации в файле	14
4.17	Отправка файла на сервер GitHub	14
4.18	Инициализация git-flow	14
4.19	Проверка текущей ветви	15
4.20	Отправка всего на сервер	15
4.21	Установка внешней ветви в качестве восходящей ветви	15
4.22	Создание новой версии выпуска	15
4.23	Создание журнала изменений	16
4.24	Добавление файла в индекс GitHub	16
4.25	Завершаем создание релизной версии	16
4.26	объединение ветви release с основной ветвью	16
4.27	Отправка файла на сервер GitHub	17
4.28	Создание тега репозитория	17
4.29	Создание ветки функциональности	17
4.30	Завершение ветки и создание новой ветки	18
4.31	Редактирование файла package.json	18
4.32	Создание журнала изменений и добавление файла в индекс	18
4.33	Отправка файла на сервер GitHub	19
4.34	Создание тега репозитория	19

Список таблиц

1 Цель работы

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 Дриссеном, оно 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.

4 Выполнение лабораторной работы

4.1 Установка git-flow

Сначала я включаю репозиторий copr, затем устанавливаю gitflow (рис. 4.1 и рис. 4.2)

```
sayprachanhlsnv@sayprachanhlsnv:~$ sudo dnf copr enable elegos/gitflow
[sudo] пароль для sayprachanhlsnv:
https://copr.fedorainfracloud.org/api_3/rpmrepo/elegos/gitflow/fedora-41/ 100% | 741.0 B/s | 654.0 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. 4.1: Включение репозитория copr

```
sayprachanhlsnv@sayprachanhlsnv:~$ sudo dnf install gitflow
Обновление и загрузка репозитория:
Copr repo for gitflow owned by elegos 100% | 91.0 B/s | 2.4 KiB | 00m27s
Репозитории загружены.
Пакет Арх. Версия Репозиторий Размер
Установка:
gitflow x86_64 1.12.3-1.fc34 copr:copr.fedorainfracloud.org:el 261.7 KiB
Сводка транзакции:
Установка: 1 пакета
```

Рис. 4.2: Установка Git-flow

4.2 Установка Node.js

Я устанавливаю Node.js для семантического управления версиями и общепринятых коммитов. (рис. 4.3)


```
sayprachanhlsvnnv@sayprachanhlsvnnv:~$ sudo dnf install nodejs
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет
Установка:
nodejs
Установка зависимостей:
nodejs-libs
Установка слабых зависимостей:
nodejs-docs
nodejs-fulli-118n
nodejs-npm
Арх. Версия Репозиторий Размер
x86_64 1:22.11.0-3.fc41 updates 148.5 KiB
x86_64 1:22.11.0-3.fc41 updates 73.8 MiB
noarch 1:22.11.0-3.fc41 updates 91.9 MiB
x86_64 1:22.11.0-3.fc41 updates 29.3 MiB
x86_64 1:10.9.0-1.22.11.0.3.fc41 updates 9.4 MiB
Сводка транзакции:
Установка: 5 пакетов
Общий размер входящих пакетов составляет 38 MiB. Необходимо загрузить 38 MiB.
После этой операции будут использоваться дополнительные 204 MiB (установка 204 MiB, удаление 0 B).
Is this ok [y/N]: y
[1/5] nodejs-1:22.11.0-3.fc41.x86_64 100% | 469.2 KiB/s | 45.0 KiB | 00m00s
```

Рис. 4.3: Установка Nodejs

а также устанавливаю pnpm для зависимостей установки и для более быстрой установки (рис. 4.4)

```
sayprachanhlsvnnv@sayprachanhlsvnnv:~$ sudo dnf install pnpm
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет
Установка:
pnpm
Арх. Версия Репозиторий Размер
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% | 1.0 MiB/s | 3.0 MiB | 00m03s
```

Рис. 4.4: Установка pnpm

4.2.1 Настройка Node.js

Затем, используя команду pnpm setup, я устанавливаю переменную PATH в системе для использования при работе с Nodejs (рис. 4.5)

```
sayprachanhlsvnnv@sayprachanhlsvnnv:~$ pnpm setup
Appended new lines to /home/sayprachanhlsvnnv/.bashrc

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

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

Рис. 4.5: Установка пути в системе

Я начинаю использовать pnpm, используя команду source ~/.bashrc, я уста-

новил в качестве источника системный файл ~/.bashrc, поскольку это путь для выполнения программы (рис. 4.6)

```
sayprachanhlsnv@sayprachanhlsnv:~$ source ~/.bashrc
sayprachanhlsnv@sayprachanhlsnv:~$
```

Рис. 4.6: Установка источника в системный файл bashrc

Я устанавливаю программу, которая используется для форматирования коммитов с помощью команды `pnpm add`, и она также предоставляет нам `git-cz`, который мы будем использовать для коммитов (рис. 4.7)

```
sayprachanhlsnv@sayprachanhlsnv:~$ 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/sayprachanhlsnv/.local/share/pnpm/global/5:
+ commitizen 4.3.1

Done in 8.3s
sayprachanhlsnv@sayprachanhlsnv:~$
```

Рис. 4.7: Установка commitizen

Я устанавливаю программу, которая использовалась для помощи в создании журнала (рис. 4.8)

```
sayprachanhlsnv@sayprachanhlsnv:~$ 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/sayprachanhlsnv/.local/share/pnpm/global/5:
+ standard-changelog 6.0.0

Done in 4.5s
sayprachanhlsnv@sayprachanhlsnv:~$
```

Рис. 4.8: Установка standard-changelog

4.3 Создание репозитория git

Я захожу в свою учетную запись на github, затем создаю новый репозиторий, присваивая ему имя git-extended (рис. 4.9)

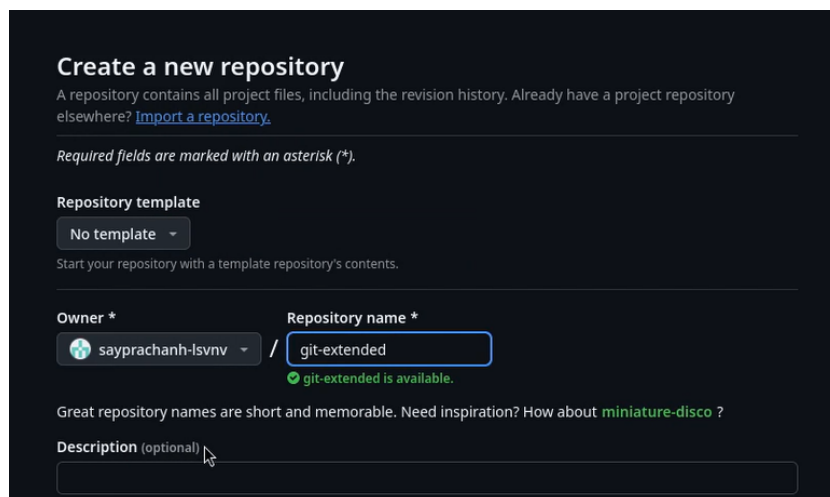


Рис. 4.9: Создание нового репозитория

Я возвращаюсь к терминалу, затем создаю новый каталог, такой же, как и в созданном репозитории (рис. 4.10)

```
sayprachanhlsnvnsayprachanhlsnv:~$ mkdir git-extended
sayprachanhlsnvnsayprachanhlsnv:~$ ls
git-extended  Видео      Загрузки  Музыка     'Рабочий стол'
work          Документы  Изображения  Общедоступные  Шаблоны
sayprachanhlsnvnsayprachanhlsnv:~$ cd git-extended/
```

Рис. 4.10: Создание нового каталога

Я начинаю делать свой первый коммит и публикую его в репозитории GitHub, который я создал ранее (рис. 4.11)

```

sayprachanhlsnv@sayprachanhlsnv:~/git-extended$ echo "# git-extended" >> README.md
sayprachanhlsnv@sayprachanhlsnv:~/git-extended$ git init
Инициализирован пустой репозиторий Git в /home/sayprachanhlsnv/git-extended/.git/
sayprachanhlsnv@sayprachanhlsnv:~/git-extended$ git add README.md
sayprachanhlsnv@sayprachanhlsnv:~/git-extended$ git commit -m "first commit"
[master (корневой коммит) 207f738] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
sayprachanhlsnv@sayprachanhlsnv:~/git-extended$ git remote add origin https://github.com/sayprachanhlsnv/git-extended.git
sayprachanhlsnv@sayprachanhlsnv:~/git-extended$ git push -u origin master
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 895 байтов | 895.00 КиБ/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/sayprachanhlsnv/git-extended.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
sayprachanhlsnv@sayprachanhlsnv:~/git-extended$

```

Рис. 4.11: Выполнение первого коммита

Я вижу, что созданный репозиторий обновлен новым файлом, который я создал во время первой коммита (рис. 4.12)

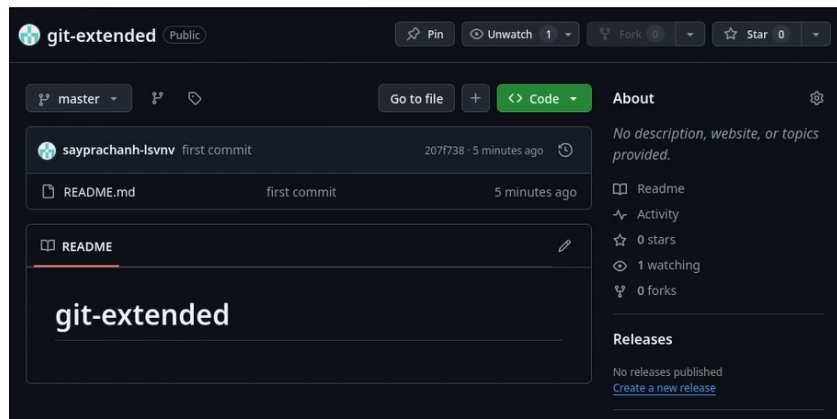


Рис. 4.12: Созданный репозиторий

Используя команду `npm init`, я создаю файл `package.json` для настройки пакета Nodejs (рис. 4.13)

```

sayprachanhlsvnnv@sayprachanhlsvnnv:~/git-extended$ npm init
Wrote to /home/sayprachanhlsvnnv/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"
}
sayprachanhlsvnnv@sayprachanhlsvnnv:~/git-extended$ ls
package.json  README.md

```

Рис. 4.13: Конфигурация для пакетов Node.js

Я открываю файл package.json в текстовом редакторе, затем ввожу несколько параметров пакета, таких как название пакета, лицензия на пакет, а также настраиваю формат коммитов (рис. 4.14)

```

1 {
2   "name": "git-extended",
3   "version": "1.0.0",
4   "description": "Git repo for education purposes",
5   "main": "index.js",
6   "repository": "https://github.com/sayprachanh-lsvnnv/git-extended.git",
7   "author": "Sayprachanh Luangsouvannavong <cocoonlysvnnv@gmail.com>",
8   "license": "CC-BY-4.0",
9   "config": {
10    "commitizen": {
11      "path": "cz-conventional-changelog"
12    }
13  }
14 }
~

```

Рис. 4.14: Редактирование файла package.json

Я добавляю файл на GitHub (рис. 4.15)

```

sayprachanhlsvnnv@sayprachanhlsvnnv:~/git-extended$ git add .

```

Рис. 4.15: Добавление файла на GitHub

Затем я выполняю коммит в файле с помощью команды git-cz (рис. 4.16)

```

sayprachanhlsnv@sayprachanhlsnv:~/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):
  (22) configure package.json
? 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 d982320] feat: configure package.json
1 file changed, 14 insertions(+)
create mode 100644 package.json

```

Рис. 4.16: Выполнение фиксации в файле

Я отправляю файл на сервер GitHub (рис. 4.17)

```

sayprachanhlsnv@sayprachanhlsnv:~/git-extended$ git push
Перечисление объектов: 4, готово.
Подсчет объектов: 100% (4/4), готово.
При сжатии изменений используется до 5 потоков
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 1.18 КиБ | 1.18 МБ/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/sayprachanh-lsvnv/git-extended.git
  207f738..d982320 master -> master
sayprachanhlsnv@sayprachanhlsnv:~/git-extended$

```

Рис. 4.17: Отправка файла на сервер GitHub

После этого я начинаю инициализировать git flow, а также настраиваю названия префиксов (рис. 4.18)

```

sayprachanhlsnv@sayprachanhlsnv:~/git-extended$ git flow init
Which branch should be used for bringing forth production releases?
- master
Branch name for production releases: [master] master
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? [/home/sayprachanhlsnv/git-extended/.git/hooks]
sayprachanhlsnv@sayprachanhlsnv:~/git-extended$

```

Рис. 4.18: Инициализация git-flow

Я объединяю ветку выпуска с основной веткой, завершая создание версии выпуска. Я проверяю, что я уже в ветке develop (рис. 4.19)

```
sayprachanhlsnv@sayprachanhlsnv:~/git-extended$ git branch
* develop
  master
sayprachanhlsnv@sayprachanhlsnv:~/git-extended$
```

Рис. 4.19: Проверка текущей ветви

Затем я попробовал загрузить все репозитории в удаленный репозиторий (рис. 4.20)

```
sayprachanhlsnv@sayprachanhlsnv:~/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/sayprachanh-lsvnv/git-extended/pull/new/develop
remote:
To https://github.com/sayprachanh-lsvnv/git-extended.git
 * [new branch]      develop -> develop
```

Рис. 4.20: Отправка всего на сервер

Я установил внешнюю ветвь в качестве восходящей для более ранней ветви (рис. 4.21)

```
sayprachanhlsnv@sayprachanhlsnv:~/git-extended$ git branch --set-upstream-to=origin/develop
develop
branch 'develop' set up to track 'origin/develop'.
```

Рис. 4.21: Установка внешней ветви в качестве восходящей ветви

Я начинаю создавать новую версию, используя команду `git flow release start`, и устанавливаю версию на 1.0.0. Я также создаю журнал изменений для документирования изменений в новой версии (рис. 4.22 и рис. 4.23)

```
sayprachanhlsnv@sayprachanhlsnv:~/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. 4.22: Создание новой версии выпуска

```
sayprachanhlsnv@sayprachanhlsnv:~/git-extended$ standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
```

Рис. 4.23: Создание журнала изменений

Затем я добавляю файл журнала изменений в индекс GitHub, готовлю его к отправке на сервер GitHub (рис. 4.24)

```
sayprachanhlsnv@sayprachanhlsnv:~/git-extended$ git add CHANGELOG.md
sayprachanhlsnv@sayprachanhlsnv:~/git-extended$ git commit -am 'chore(site): add changelog'
[release/1.0.0 8421ec8] chore(site): add changelog
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md
```

Рис. 4.24: Добавление файла в индекс GitHub

Используя команду `git flow release finish` и следуя за версией выпуска, я объединяю ветку выпуска с основной веткой, завершая создание версии выпуска. (рис. 4.25 и рис. 4.26)

```
sayprachanhlsnv@sayprachanhlsnv:~/git-extended$ git flow release finish 1.0.0
```

Рис. 4.25: Завершаем создание релизной версии

```
GNU nano 8.1 /home/sayprachanhlsnv/git-extended/.git/MERGE_MSG
Merge tag '1.0.0' into develop

Adding changelog
# Пожалуйста, введите сообщение коммита, для объяснения, зачем нужно
# это слияние, особенно, если это слияние обновленной вышестоящей
```

Рис. 4.26: объединение ветви release с основной ветвью

Я отправляю все на сервер GitHub с помощью `git push` (рис. 4.27)


```

sayprachanhlsnv@sayprachanhlsnv:~/git-extended$ git push --all
Перечисление объектов: 6, готово.
Подсчет объектов: 100% (6/6), готово.
При сжатии изменений используется до 5 потоков
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (5/5), 2.85 КиБ | 2.85 МиБ/с, готово.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/sayprachanh-lsvnv/git-extended.git
   d982320..2a42567  develop -> develop
   d982320..7df96ed  master -> master
sayprachanhlsnv@sayprachanhlsnv:~/git-extended$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 182 байта | 182.00 КиБ/с, готово.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/sayprachanh-lsvnv/git-extended.git
   * [new tag]           1.0.0 -> 1.0.0

```

Рис. 4.27: Отправка файла на сервер GitHub

Я создаю тег выпуска в репозитории GitHub и устанавливаю версию выпуска на 1.0.0, поскольку это первый релиз (рис. 4.28)

```

sayprachanhlsnv@sayprachanhlsnv:~/git-extended$ gh release create v1.0.0 -F CHANGELOG.md
https://github.com/sayprachanh-lsvnv/git-extended/releases/tag/v1.0.0

```

Рис. 4.28: Создание тега репозитория

4.4 Работа с репозиторием git

Я создаю новую ветку для разработки новой функциональности для предыдущей версии релиза, используя команду `git flow feature start`. Затем я начинаю работать с новой функциональностью. (рис. 4.29)

```

sayprachanhlsnv@sayprachanhlsnv:~/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. 4.29: Создание ветки функциональности

После завершения работы и разработки новой функциональности я объединяю созданную ветку `feature_branch` с веткой `develop`. Затем я создаю релиз с версией

1.2.3 для обновления старого релиза до версии 1.0.0 (рис. 4.30)

```
sayprachanhlsnv@sayprachanhlsnv:~/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 2a42567).

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'

sayprachanhlsnv@sayprachanhlsnv:~/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. 4.30: Завершение ветки и создание новой ветки

Я редактирую файл package.json, изменяя версию выпуска с версии 1.0.0 на версию 1.2.3 (рис. 4.31)

```
1 {
2   "name": "git-extended",
3   "version": "1.2.3",
4   "description": "Git repo for education purposes",
```

Рис. 4.31: Редактирование файла package.json

После этого я создаю журнал изменений для документирования изменений и добавляю все в индекс GitHub, готовлю к отправке на сервер GitHub (рис. 4.32)

```
sayprachanhlsnv@sayprachanhlsnv:~/git-extended$ standard-changelog
✓ output changes to CHANGELOG.md
sayprachanhlsnv@sayprachanhlsnv:~/git-extended$ git commit -am 'chore(site): update changelog'
[release/1.2.3 c129b56] chore(site): update changelog
2 files changed, 5 insertions(+), 1 deletion(-)
```

Рис. 4.32: Создание журнала изменений и добавление файла в индекс

Я отправляю новый релиз на сервер GitHub (рис. 4.33)

```

sayprachanhlsnv@sayprachanhlsnv:~/git-extended$ git push --all
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 5 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 2.83 КиБ | 2.83 МБ/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 https://github.com/sayprachanh-lsvnv/git-extended.git
  2a42567..fb2d1a7  develop -> develop
  7df96ed..1be5b43  master -> master
sayprachanhlsnv@sayprachanhlsnv:~/git-extended$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 187 байтов | 187.00 КиБ/с, готово.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/sayprachanh-lsvnv/git-extended.git
 * [new tag]          1.2.3 -> 1.2.3

```

Рис. 4.33: Отправка файла на сервер GitHub

Я создаю тег выпуска в репозитории github с комментарием из журнала изменений (рис. 4.34)

```

sayprachanhlsnv@sayprachanhlsnv:~/git-extended$ gh release create v1.2.3 -F CHANGELOG.md
https://github.com/sayprachanh-lsvnv/git-extended/releases/tag/v1.2.3
sayprachanhlsnv@sayprachanhlsnv:~/git-extended$

```

Рис. 4.34: Создание тега репозитория

5 Выводы

Во время выполнения лабораторной работы я понял и получил навыки правильной работы с репозиториями git.

Список литературы

Лабораторная работа №4