

Отчёт по лабораторной работе

Лабораторная №4

Полина Витальевна Барабаш

Содержание

<i>1</i>	<i>Цель работы</i>	<i>5</i>
<i>2</i>	<i>Задание</i>	<i>6</i>
<i>3</i>	<i>Выполнение лабораторной работы</i>	<i>7</i>
<i>4</i>	<i>Выводы</i>	<i>21</i>
	<i>Список литературы</i>	<i>22</i>

Список иллюстраций

3.1	Включение репозитория <code>corp</code>	7
3.2	Установка <code>git-flow</code>	8
3.3	Установка <code>Node.js</code>	8
3.4	Установка <code>pnpm</code>	9
3.5	Настройка <code>pnpm</code>	9
3.6	Перезагрузка <code>pnpm</code>	10
3.7	Добавление программы <code>commitizen</code>	10
3.8	Добавление программы <code>standard-changelog</code>	10
3.9	Создание и копирования репозитория на GitHub	11
3.10	Добавление первого коммита	11
3.11	Выгрузка первого коммита на github	11
3.12	Настройка конфигурации общепринятых коммитов	12
3.13	Добавление новых файлов и коммита	12
3.14	Отправка файлов на github	13
3.15	Инициализация <code>git-flow</code>	13
3.16	Проверка местонахождения	14
3.17	Проверка местонахождения	14
3.18	Проверка местонахождения	14
3.19	Отправка файлов на сервер	15
3.20	Создание журнала изменений	15
3.21	Добавление журнала изменений в индекс	15
3.22	Слитие релизной ветки в основную	16
3.23	Отправка данных на github	16
3.24	Создание релиза на github	17
3.25	Создание ветки для новой функциональности	17
3.26	Объединение веток	17
3.27	Создание релиза с версией 1.2.3	18
3.28	Обновление номера версии	18
3.29	Создание журнала изменений	18
3.30	Добавление журнала изменений в индекс	19
3.31	Объединение веток	19
3.32	Отправление данных на github	20
3.33	Создание релиза на github	20

Список таблиц

1 Цель работы

Целью данной работы является получение навыков правильной работы с репозиториями git [1].

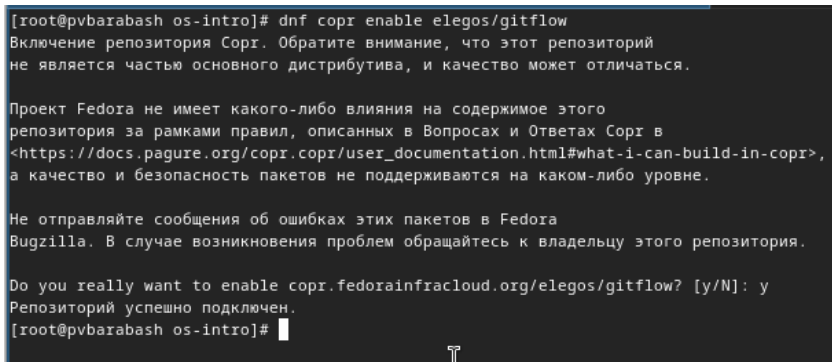
2 Задание

Выполнить работу для тестового репозитория.

3 Выполнение лабораторной работы

Задание 1. Установить git-flow.

Из режима суперпользователя я установила git-flow, сначала введя команду `dnf copr enable elegos/gitflow` (рис. [3.1]), а затем `dnf install gitflow` (рис. [3.2]).



```
[root@pvbarabash os-intro]# 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>,
а 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@pvbarabash os-intro]#
```

Рис. 3.1: Включение репозитория copr

```
[root@pubarabash ~]# dnf install gitflow
Copr repo for gitflow owned by elegos                                     1.9 kB/s | 1.5 kB    00:00
Зависимости разрешены.
=====
Пакет      Архитектура  Версия      Репозиторий      Размер
-----
Установка: gitflow      x86_64      1.12.3-1.fc34  copr.copr.fedorainfracloud.org:elegos:gitflow  57 k
=====
Результат транзакции
Установка 1 Пакет

Объем загрузки: 57 k
Объем изменений: 262 k
Продолжить? [д/н]: y
Загрузка пакетов:
gitflow-1.12.3-1.fc34.x86_64.rpm                                     161 kB/s | 57 kB    00:00
-----
Общий размер
Copr repo for gitflow owned by elegos                               159 kB/s | 57 kB    00:00
Импорт GPG-ключа 0x8BF63AA3:
Идентификатор пользователя: "elegos@gitflow@copr.fedorahosted.org"
Отпечаток: 9357 9980 49C5 C5C3 6CF3 9A22 823C A8E6 88F6 3AA3
Источник: https://download.copr.fedorainfracloud.org/results/elegos/gitflow/pubkey.gpg
Продолжить? [д/н]: y
Импорт ключа успешно завершен
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверка транзакции
Тест транзакции проведен успешно.
Выполнение транзакции
Подготовка      : gitflow-1.12.3-1.fc34.x86_64      1/1
Установка       : gitflow-1.12.3-1.fc34.x86_64      1/1
Проверка        : gitflow-1.12.3-1.fc34.x86_64      1/1
Установлен:
gitflow-1.12.3-1.fc34.x86_64
Выполнено!
```

Рис. 3.2: Установка git-flow

Задание 2. Установить Node.js.

Я установила Node.js с помощью команды `dnf install nodejs` (рис. [3.3]).

```
nodejs-docs      noarch      1:20.10.0-3.fc39  updates      8.1 M
nodejs-full-libs 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]: y
Загрузка пакетов:
(1/5): nodejs-20.10.0-3.fc39.x86_64.rpm      743 kB/s | 48 kB    00:00
(2/5): nodejs-full-libs-20.10.0-3.fc39.x86_64.rpm  5.2 MB/s | 8.5 MB    00:01
(3/5): nodejs-npm-10.2.3-1.20.10.0-3.fc39.x86_64.rpm  4.7 MB/s | 2.2 MB    00:00
(4/5): nodejs-docs-20.10.0-3.fc39.noarch.rpm    3.6 MB/s | 8.1 MB    00:02
(5/5): nodejs-libs-20.10.0-3.fc39.x86_64.rpm    6.6 MB/s | 15 MB    00:02
-----
Общий размер
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверка транзакции
Тест транзакции проведен успешно.
Выполнение транзакции
Запуск скрипглета: nodejs-1:20.10.0-3.fc39.x86_64      1/1
Подготовка      : nodejs-1:20.10.0-3.fc39.noarch      1/1
Установка       : nodejs-libs-1:20.10.0-3.fc39.x86_64  1/5
Установка       : nodejs-full-libs-1:20.10.0-3.fc39.x86_64  2/5
Установка       : nodejs-npm-1:10.2.3-1.20.10.0-3.fc39.x86_64  3/5
Установка       : nodejs-1:20.10.0-3.fc39.x86_64      4/5
Запуск скрипглета: 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-full-libs-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-full-libs-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!
```

Рис. 3.3: Установка Node.js

Затем я установила `pnpm` с помощью команды `dnf install pnpm` (рис. [3.4]).


```
[root@pvbarabash ~]# dnf install pnpm
Последняя проверка окончания срока действия метаданных: 0:02:20 назад, Ср 06 мар 2024 16:27:59.
Зависимости разрешены.
=====
Пакет                Архитектура      Версия            Репозиторий        Размер
-----
Установка:
pnpm                 noarch            8.12.0-1.fc39     updates             2.6 М
=====
Результат транзакции
Установка 1 Пакет
Объем загрузки: 2.6 М
Объем изменений: 13 М
Продолжить? [д/н]: y
Загрузка пакетов:
pnpm-8.12.0-1.fc39.noarch.rpm                6.5 MB/s | 2.6 MB  00:00
-----
Общий размер                                2.1 MB/s | 2.6 MB  00:01
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверка транзакции
Тест транзакции проведен успешно.
Выполнение транзакции
Подготовка      :
Установка       : pnpm-8.12.0-1.fc39.noarch             1/1
Запуск скрипта  : pnpm-8.12.0-1.fc39.noarch             1/1
Проверка        : pnpm-8.12.0-1.fc39.noarch             1/1
Установлен:
pnpm-8.12.0-1.fc39.noarch
Выполнено!
[root@pvbarabash ~]#
```

Рис. 3.4: Установка pnpm

Задание 3. Настроить Node.js.

Я выполнила команду настройки pnpm setup (рис. [3.5]).

```
[root@pvbarabash ~]# 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

To start using pnpm, run:
source /root/.bashrc
[root@pvbarabash ~]#
```

Рис. 3.5: Настройка pnpm

А затем выполнила команду `source ~/.bashrc`, чтобы использовать pnpm (рис. [3.6]).

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

Рис. 3.6: Перезагрузка pnpm

Задание 4. Настроить общепринятые коммиты.

Я добавила программу commitizen с помощью команды `pnpm add -g commitizen` (рис. [3.7]).

```
[root@pvbarabash ~]# 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 28.6s
```

Рис. 3.7: Добавление программы commitizen

Я добавила программу standard-changelog с помощью команды `pnpm add -g standard-changelog` (рис. [3.8]).

```
[root@pvbarabash ~]# 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 36.9s
```

Рис. 3.8: Добавление программы standard-changelog

Задание 5. Создать репозиторий git.

Я создала новый репозиторий на GitHub под названием git-extended и клонировала его в каталог git-extended своей виртуальной машины (рис. [3.9]).

```
[root@pvbarabash ~]# git clone --recursive git@github.com:pvbarabash/git-extended.git git-extended
Клонирование в «git-extended»...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Получение объектов: 100% (3/3), готово.
[root@pvbarabash ~]#
```

Рис. 3.9: Создание и копирования репозитория на GitHub

Задание 6. Сделать первый коммит и выложить на github.

Я создала файл firstcommit.txt для того, чтобы были какие-то изменения в репозитории. Затем я выполнила команды git add . и git commit -m "first commit" (рис. [3.10]).

```
[root@pvbarabash git-extended]# git add .
[root@pvbarabash git-extended]# git commit -m "first commit"
[main 414290b] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 firstcommit.txt
[root@pvbarabash git-extended]#
```

Рис. 3.10: Добавление первого коммита

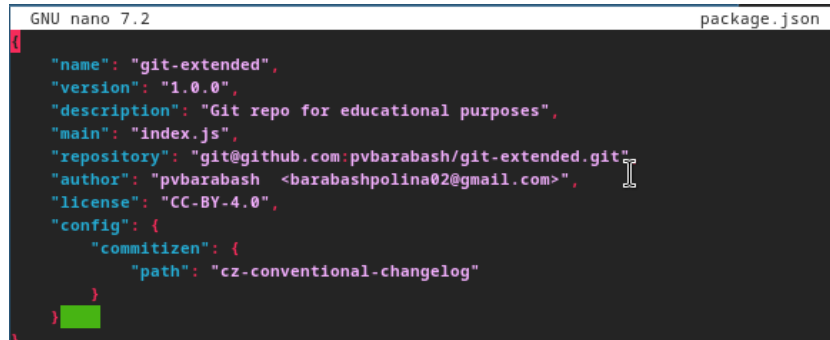
Затем я выложила изменения на github с помощью команды git push -u origin main (рис. [3.11]).

```
[root@pvbarabash git-extended]# git push -u origin main
Перечисление объектов: 4, готово.
Подсчет объектов: 100% (4/4), готово.
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 944 байта | 944.00 КиБ/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:pvbarabash/git-extended.git
6faa434..414290b main -> main
branch 'main' set up to track 'origin/main'.
[root@pvbarabash git-extended]#
```

Рис. 3.11: Выгрузка первого коммита на github

Задание 7. Настроить конфигурацию общепринятых коммитов.

Сначала я инициализировала `pnpm` с помощью команды `pnpm init`. Затем открыла файл `package.json` в `nano` и изменила текст в соответствии с тем, что выложен на ТУИС (рис. [3.12]).

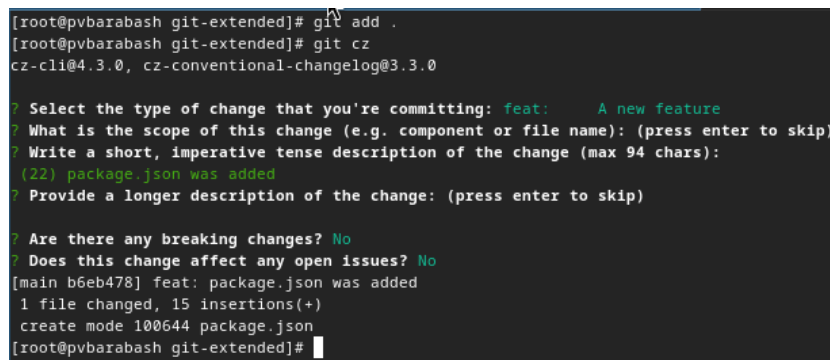


```
GNU nano 7.2 package.json
{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:pvbarabash/git-extended.git",
  "author": "pvbarabash <barabashpolina02@gmail.com>",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 3.12: Настройка конфигурации общепринятых коммитов

Задание 8. Добавить новые файлы, выполнить коммит и отправить на github.

Я добавила новые файлы с помощью команды `git add .`. Выполнила коммит с помощью команды `git cz` и ответила на выпадающие вопросы (рис. [3.13]).



```
[root@pvbarabash git-extended]# git add .
[root@pvbarabash 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)
? Write a short, imperative tense description of the change (max 94 chars):
  (22) package.json was added
? 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 b6eb478] feat: package.json was added
1 file changed, 15 insertions(+)
create mode 100644 package.json
[root@pvbarabash git-extended]#
```

Рис. 3.13: Добавление новых файлов и коммита

И затем отправила на github с помощью команды `git push` (рис. [3.14]).

```
[root@pvbarabash git-extended]# git push
Перечисление объектов: 4, готово.
Подсчет объектов: 100% (4/4), готово.
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 1.19 КиБ | 407.00 КиБ/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:pvbarabash/git-extended.git
  414290b..b6eb478  main -> main
[root@pvbarabash git-extended]#
```

Рис. 3.14: Отправка файлов на github

Задание 9. Задать конфигурацию git-flow.

Я инициализировала git-flow с помощью команды `git flow init`, установив префикс для ярлыков в `v` (рис. [3.15]).

```
[root@pvbarabash 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? [/root/git-extended/.git/hooks]
[root@pvbarabash git-extended]#
```

Рис. 3.15: Инициализация git-flow

Я проверила, что нахожусь на ветке `develop` с помощью команды `git branch` (рис. [3.16]).

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

Рис. 3.16: Проверка местонахождения

Я загрузила весь репозиторий в хранилище с помощью команды `git push --all` (рис. [3.17]).

```
[root@pvbarabash 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/pvbarabash/git-extended/pull/new/develop
remote:
To github.com:pvbarabash/git-extended.git
 * [new branch]      develop -> develop
[root@pvbarabash git-extended]#
```

Рис. 3.17: Проверка местонахождения

Я установила внешнюю ветку как вышестоящую для этой ветки с помощью `git branch --set-upstream-to=origin/develop develop` (рис. [3.18]).

```
[root@pvbarabash git-extended]# git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
[root@pvbarabash git-extended]#
```

Рис. 3.18: Проверка местонахождения

Я создала релиз с версией 1.0.0 с помощью `git flow release start 1.0.0` (рис. [3.19]).

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

Рис. 3.19: Отправка файлов на сервер

Я создала журнал изменений с помощью `standard-changelog --first-release` (рис. [3.20]).

```
[root@pvbarabash git-extended]# standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
[root@pvbarabash git-extended]#
```

Рис. 3.20: Создание журнала изменений

Я добавила журнал изменений в индекс с помощью команд `git add CHANGELOG.md` и `git commit -am 'chore(site): add changelog'` (рис. [3.21]).

```
[root@pvbarabash git-extended]# git add CHANGELOG.md
[root@pvbarabash git-extended]# git commit -am 'chore(site): add changelog'
[release/1.0.0 21dc730] chore(site): add changelog
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md
[root@pvbarabash git-extended]#
```

Рис. 3.21: Добавление журнала изменений в индекс

Я залила релизную ветку в основную ветку с помощью `git flow release finish 1.0.0` (рис. [3.22]).

```

[root@pvbarabash 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 | 9 ++++++++
  1 file changed, 9 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)
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 21dc730).

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'

[root@pvbarabash git-extended]#

```

Рис. 3.22: Слитие релизной ветки в основную

Я отправила данные на github с помощью команд `git push --all` и `git push --tags` (рис. [3.23]).

```

[root@pvbarabash git-extended]# git push --all
Перечисление объектов: 6, готово.
Подсчет объектов: 100% (6/6), готово.
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (5/5), 2.76 КиБ | 2.76 МБ/с, готово.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:pvbarabash/git-extended.git
   b6eb478..332447d  develop -> develop
   b6eb478..da2fac9  main -> main
[root@pvbarabash git-extended]# git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 162 байта | 162.00 КиБ/с, готово.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:pvbarabash/git-extended.git
 * [new tag]         v1.0.0 -> v1.0.0
[root@pvbarabash git-extended]#

```

Рис. 3.23: Отправка данных на github

Я создала релиз на github. Для этого использовала утилиты работы с github: `gh`

release create v1.0.0 -F CHANGELOG.md (рис. [3.24]).

```
[root@pvbarabash git-extended]# gh release create v1.0.0 -F CHANGELOG.md
https://github.com/pvbarabash/git-extended/releases/tag/v1.0.0
[root@pvbarabash git-extended]#
```

Рис. 3.24: Создание релиза на github

Я создала ветку для новой функциональности с помощью команды `git flow feature start feature_branch` (рис. [3.25]).

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

Рис. 3.25: Создание ветки для новой функциональности

Следующим шагом я объединила ветку `feature_branch` с `develop` с помощью `git flow feature finish feature_branch` (рис. [3.26]).

```
[root@pvbarabash 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 332447d).

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@pvbarabash git-extended]#
```

Рис. 3.26: Объединение веток

Я создала релиз с версией 1.2.3 с помощью команды `git flow release start 1.2.3` (рис. [3.27]).

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

Рис. 3.27: Создание релиза с версией 1.2.3

Я обновила номер версии в файле package.json: установила её в 1.2.3 (рис. [3.28]).

```
GNU nano 7.2 package.json
{
  "name": "git-extended",
  "version": "1.2.3",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:pvbarabash/git-extended.git",
  "author": "pvbarabash <barabashpolina02@gmail.com>",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 3.28: Обновление номера версии

Я создала журнал изменений с помощью команды standard-changelog (рис. [3.29]).

```
[root@pvbarabash git-extended]# standard-changelog
✓ output changes to CHANGELOG.md
[root@pvbarabash git-extended]#
```

Рис. 3.29: Создание журнала изменений

Я добавила журнал изменений в индекс с помощью команд git add

CHANGELOG.md и `git commit -am 'chore(site): update changelog'` (рис. [3.30]).

```
[root@pvbarabash git-extended]# git add CHANGELOG.md
[root@pvbarabash git-extended]# git commit -am 'chore(site): update changelog'
[release/1.2.3 5b9be17] chore(site): update changelog
2 files changed, 5 insertions(+), 1 deletion(-)
[root@pvbarabash git-extended]#
```

Рис. 3.30: Добавление журнала изменений в индекс

Я залила релизную ветку в основную ветку с помощью команды `git flow release finish 1.2.3` (рис. [3.31]).

```
[root@pvbarabash git-extended]# git flow release finish 1.2.3
Переключились на ветку «main»
Эта ветка соответствует «origin/main».
Merge made by the 'ort' strategy.
 CHANGELOG.md | 4 +++++
 package.json | 2 +-
 2 files changed, 5 insertions(+), 1 deletion(-)
Уже на «main»
Ваша ветка опережает «origin/main» 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 5b9be17).

Summary of actions:
- Release branch 'release/1.2.3' has been merged into 'main'
- 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@pvbarabash git-extended]#
```

Рис. 3.31: Объединение веток

Я отправила данные на github с помощью команд `git push -all` и `git push -tags` (рис. [3.32]).

```
[root@pvbarabash git-extended]# git push --all
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 2.74 КиБ | 935.00 КиБ/с, готово.
Total 6 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:pvbarabash/git-extended.git
   332447d..cad0646  develop -> develop
   da2fac9..0762fec  main -> main
[root@pvbarabash git-extended]# git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 160 байтов | 160.00 КиБ/с, готово.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:pvbarabash/git-extended.git
 * [new tag]         v1.2.3 -> v1.2.3
[root@pvbarabash git-extended]#
```

Рис. 3.32: Отправление данных на github

Я создала релиз на github с комментарием из журнала изменений с помощью команды `gh release create v1.2.3 -F CHANGELOG.md` (рис. [3.33]).

```
[root@pvbarabash git-extended]# gh release create v1.2.3 -F CHANGELOG.md
https://github.com/pvbarabash/git-extended/releases/tag/v1.2.3
[root@pvbarabash git-extended]#
```

Рис. 3.33: Создание релиза на github

4 Выводы

При выполнении данной лабораторной работы я получила навыки правильной работы с репозиториями git, настроила конфигурацию общепринятых коммитов.

Список литературы

1. ТУИС [Электронный ресурс]. РУДН. URL: <https://esystem.rudn.ru/mod/page/view.php?id=1098937>.