

Лабораторная работа №4

Дисциплина: Операционные системы

Панина Жанна Валерьевна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
4.1	Установка программного обеспечения	8
4.1.1	Установка git-flow	8
4.1.2	Установка Node.js	9
4.1.3	Настройка Node.js	10
4.1.4	Общепринятые коммиты	10
4.2	Практический сценарий использования git	11
4.2.1	Создание репозитория git	11
4.2.2	Работа с репозиторием git	15
5	Выводы	19
	Список литературы	20

Список иллюстраций

4.1	Команда <code>dnf copr enable elegos/gitflow</code>	8
4.2	Команда <code>dnf install gitflow</code>	9
4.3	Команда <code>dnf install nodejs</code>	9
4.4	Команда <code>dnf install pnpm</code>	10
4.5	Настройка Node.js	10
4.6	Программа <code>commitizen</code>	11
4.7	Программа <code>standard-changelog</code>	11
4.8	Создание репозитория	11
4.9	Первый коммит	12
4.10	Конфигурация общепринятых коммитов	12
4.11	Заполнение параметров пакета	13
4.12	Отправка файлов на GitHub	13
4.13	Конфигурация <code>git-flow</code>	14
4.14	Создание релиза и журнала изменений	14
4.15	Редактирование коммита	14
4.16	Команда <code>finish</code>	15
4.17	Отправка данных на github	15
4.18	Разработка новой функциональности	16
4.19	Создание релиза <code>git-flow</code>	16
4.20	Обновление номера версии в файле	16
4.21	Создание журнала изменений	17
4.22	Редактирование коммита	17
4.23	Команда <code>finish</code>	17
4.24	Создание релиза на github	18

Список таблиц

1 Цель работы

Получение навыков правильной работы с репозиториями git.

2 Задание

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

3 Теоретическое введение

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

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

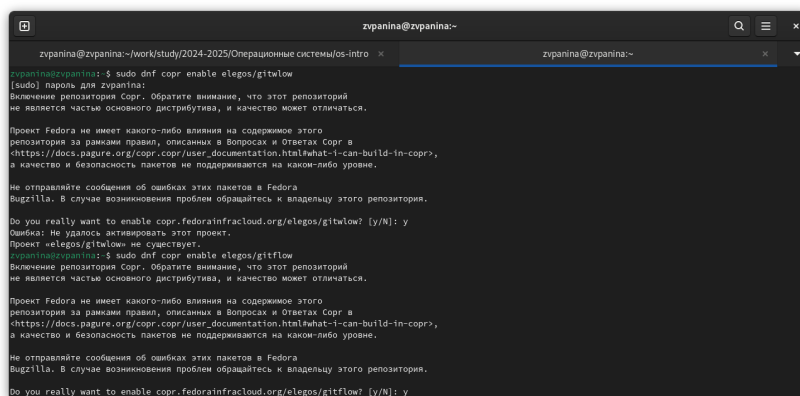
Спецификация Conventional Commits: Соглашение о том, как нужно писать сообщения commit'ов. Совместимо с SemVer. Даже вернее сказать, сильно связано с семантическим версионированием. Регламентирует структуру и основные типы коммитов.

4 Выполнение лабораторной работы

4.1 Установка программного обеспечения

4.1.1 Установка git-flow

Устанавливаю git-flow из коллекции репозитория Copr 1. Выполняю команду `dnf copr enable elegos/gitflow` в режиме суперпользователя (рис. 4.1).



```
zypanina@zypanina: ~/work/study/2024-2025/Операционные системы/os-intro
zypanina@zypanina:~$ sudo dnf copr enable elegos/gitflow
[sudo] пароль для zypanina:
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
Ошибки: Не удалось активировать этот проект.
Проект «elegos/gitflow» не существует.
zypanina@zypanina:~$ sudo 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
```

Рис. 4.1: Команда `dnf copr enable elegos/gitflow`

2. Выполняю команду `dnf install gitflow` (рис. 4.2).


```
zypanina@zypanina:~/work/study/2024-2025/Операционные системы/os-intro$ sudo dnf install gitflow
Copr repo for gitflow owned by elegos
Fedora 40 - x86_64 - Updates
Fedora 40 - x86_64 - Updates
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
=====
Общий размер
Copr repo for gitflow owned by elegos
Импорт GPG-ключа 0x80F63AA3:
Идентификатор пользователя: "elegos.gitflow (None) <elegos.gitflow@copr.fedorahosted.org>"
Отпечаток: 9357 1980 49C5 C5C3 6CF3 9A22 823C AB66 88F6 3AA3
Источник: https://download.copr.fedorainfracloud.org/results/elegos/gitflow/pubkey.gpg
Продолжить? [д/н]: y
Импорт ключа успешно завершен
```

Рис. 4.2: Команда dnf install gitflow

4.1.2 Установка Node.js

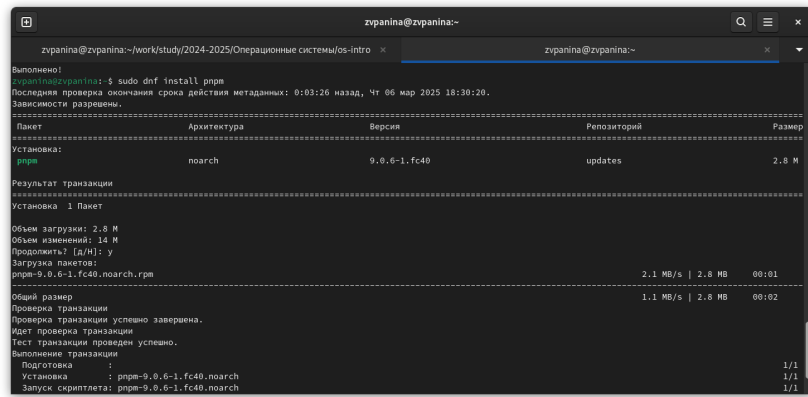
1. Выполняю команду dnf install nodejs в режиме суперпользователя (рис. 4.3).

```
zypanina@zypanina:~/work/study/2024-2025/Операционные системы/os-intro$ sudo dnf install nodejs
Последняя проверка окончания срока действия метаданных: 0:01:59 назад, Чт 06 мар 2025 18:30:20.
Зависимости разрешены.
=====
Пакет      Архитектура  Версия      Репозиторий      Размер
-----
Установка:
nodejs     x86_64       1:20.18.2-2.fc40  updates          52 k
Установка зависимостей:
nodejs-libs x86_64       1:20.18.2-2.fc40  updates          16 M
Установка слабых зависимостей:
nodejs-docs noarch       1:20.18.2-2.fc40  updates          8.4 M
nodejs-full-libm x86_64       1:20.18.2-2.fc40  updates          8.4 M
nodejs-ppc64le x86_64       1:10.8.2-1.20.18.2.2.fc40  updates          2.1 M
=====
Результат транзакции
=====
Установка 5 Пакетов

Объем загрузки: 35 M
Объем изменений: 188 M
Продолжить? [д/н]: y
Загрузка пакетов:
(1/5): nodejs-20.18.2-2.fc40.x86_64.rpm
(2/5): nodejs-full-libm-20.18.2-2.fc40.x86_64.rpm
(3/5): nodejs-ppc64le-10.8.2-1.20.18.2.2.fc40.x86_64.rpm
(4/5): nodejs-libs-20.18.2-2.fc40.x86_64.rpm
(5/5): nodejs-docs-20.18.2-2.fc40.noarch.rpm
=====
378 kB/s | 8.4 MB 00:22
```

Рис. 4.3: Команда dnf install nodejs

2. Выполняю команду dnf install npm (рис. 4.4).

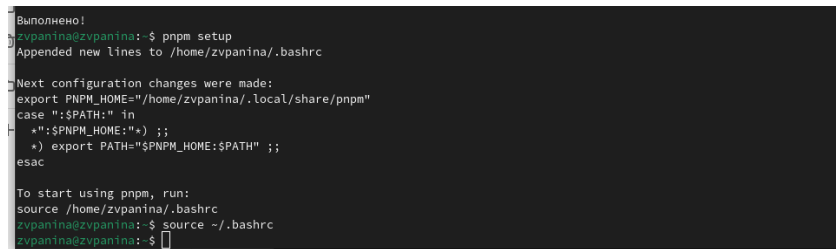


```
zypanina@zypanina:~$ sudo dnf install pnpm
Последняя проверка окончания срока действия метаданных: 0:03:26 назад, Чт 06 мар 2025 18:30:20.
Зависимости разрешены.
=====
Пакет      Архитектура  Версия  Репозиторий  Размер
-----
Установка: pnpm      noarch    9.0.6-1.fc40  updates      2.8 М
=====
Результат транзакции
Установка 1 Пакет
=====
Объем загрузки: 2.8 М
Объем изменений: 1.1 М
Продолжить [d/n]: y
Загрузка пакетов:
pnpm-9.0.6-1.fc40.noarch.rpm                                2.1 MB/s | 2.8 MB  00:01
-----
Общий размер                                1.1 MB/s | 2.8 MB  00:02
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверка транзакции
Тест транзакции проведен успешно.
Выполнение транзакции
Подготовка
Установка      : pnpm-9.0.6-1.fc40.noarch      1/1
Запуск скриплетов: pnpm-9.0.6-1.fc40.noarch      1/1
```

Рис. 4.4: Команда dnf install pnpm

4.1.3 Настройка Node.js

Для работы с Node.js добавляю каталог с исполняемыми файлами, устанавливаемыми в yarn, в переменную PATH (рис. 4.5).



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

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

To start using pnpm, run:
source /home/zypanina/.bashrc
zypanina@zypanina:~$ source ~/.bashrc
zypanina@zypanina:~$
```

Рис. 4.5: Настройка Node.js

4.1.4 Общепринятые коммиты

1. Данная программа используется для помощи в форматировании коммитов (рис. 4.6).

```
zvpanina@zvpanina:~$ pnpm add -g commitizen

Update available! 0.0.0 -> 10.5.2.
Changelog: https://github.com/pnpm/pnpm/releases/tag/v10.5.2
Run "pnpm add -g pnpm" to update.

Follow @pnpmjs for updates: https://twitter.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/zvpanina/.local/share/pnpm/global/5:
+ commitizen 4.3.1

Done in 9.7s
zvpanina@zvpanina:~$
```

Рис. 4.6: Программа commitizen

2. Данная программа используется для помощи в создании логов (рис. 4.7).

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

Done in 4.1s
zvpanina@zvpanina:~$
```

Рис. 4.7: Программа standard-changelog

4.2 Практический сценарий использования git

4.2.1 Создание репозитория git

1. Создаю репозиторий на GitHub и называю его git-extended (рис. 4.8).

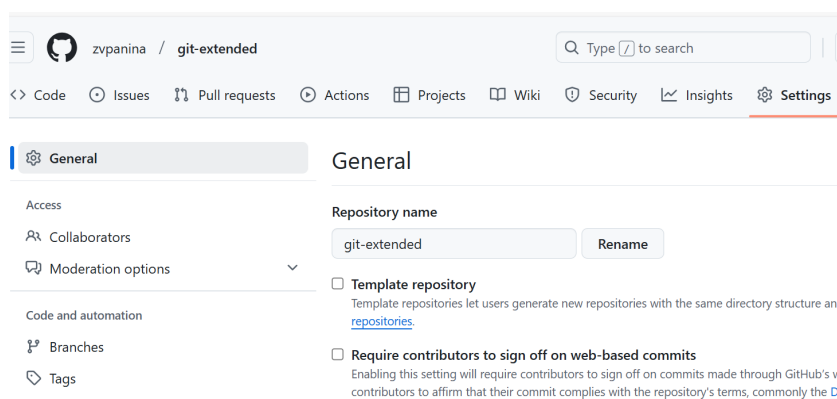


Рис. 4.8: Создание репозитория

2. Клонировать репозиторий, перейти в него и создать файл README.md, чтобы закоммитить его. С помощью команд `git add.`, `git commit`, `git push` делаю первый коммит и выкладываю его на GitHub (рис. 4.9).

```
Клонирование в «git-extended»...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Получение объектов: 100% (5/5), готово.
zvpanina@zvpanina:~$ cd ~/git-extended
zvpanina@zvpanina:~/git-extended$ touch README.md
zvpanina@zvpanina:~/git-extended$ git add .
zvpanina@zvpanina:~/git-extended$ git commit -m "first commit"
[main 8924e39] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
zvpanina@zvpanina:~/git-extended$ git remote add origin git@github.com:zvpanina/git-extended.git
error: внешний репозиторий origin уже существует
zvpanina@zvpanina:~/git-extended$ git remote -v
origin  git@github.com:zvpanina/git-extended.git (fetch)
origin  git@github.com:zvpanina/git-extended.git (push)
zvpanina@zvpanina:~/git-extended$ git push -u origin master
error: src refspec master ничего не соответствует
error: не удалось отправить некоторые объекты в git@github.com:zvpanina/git-extended.git
zvpanina@zvpanina:~/git-extended$ git push -u origin
ssh: connect to host github.com port 22: Connection refused
fatal: Не удалось прочитать из внешнего репозитория.

Удостоверьтесь, что у вас есть необходимые права доступа
и репозиторий существует.
zvpanina@zvpanina:~/git-extended$ git push -u origin
Перечисление объектов: 4, готово.
Подсчет объектов: 100% (4/4), готово.
```

Рис. 4.9: Первый коммит

3. Создаю конфигурацию для пакетов Node.js и открываю файл `git-extended` в `mc` (рис. 4.10).

```
zvpanina@zvpanina:~/git-extended$ npm init
Wrote to /home/zvpanina/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"
}
zvpanina@zvpanina:~/git-extended$ mc
```

Рис. 4.10: Конфигурация общепринятых коммитов

4. Заполняю несколько параметров пакета (название, лицензия, конфигурация пакета коммитов), чтобы файл выглядел вот так (рис. 4.11).

```

package.json  [-M--]  0 L: [ 1+17 18/ 19] *(443 / 445b) 0125 0x07D
{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:zvpalina/git-extended.git",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "keywords": [],
  "author": "Zhanna Panina zhanna.panina@gmail.com",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}

```

Рис. 4.11: Заполнение параметров пакета

5. С помощью следующих команд добавляю новые файлы, выполняю коммит, выкладываю на GitHub (рис. 4.12).

```

zvpalina@zvpalina:~/git-extended
zvpalina@zvpalina:~/git-extended$ git add .
zvpalina@zvpalina:~/git-extended$ git cz
cz-cli@4.3.1, cz-conventional-changelog@3.3.0

Select the type of change 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):
(1) change 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
main e0d5981 feat: change package.json
1 file changed, 18 insertions(+)
create mode 100644 package.json
zvpalina@zvpalina:~/git-extended$ git push
ssh: connect to host github.com port 22: Connection refused
fatal: Не удалось прочитать из внешнего репозитория.

Удостоверьтесь, что у вас есть необходимые права доступа
и репозиторий существует.
zvpalina@zvpalina:~/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), 1.20 KiB | 1.20 MiB/s, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:zvpalina/git-extended.git
 1a6d9e3..e698598 main -> main
zvpalina@zvpalina:~/git-extended$

```

Рис. 4.12: Отправка файлов на GitHub

6. Инициализирую git-flow (Префикс для ярлыков устанавливаю в v); проверяю, что я на ветке develop; загружаю весь репозиторий в хранилище; устанавливаю внешнюю ветку как вышестоящую для этой ветки (рис. 4.13).

```
zspanina@zspanina:~/work/study/2024-2025/Операционные системы/os-intro x zspanina@zspanina:~/git-extended x
1a6d9e3..e698598 main -> main
zspanina@zspanina:~/git-extended$ git flow init

Which branch should be used for bringing forth production releases?
  main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? [] v
Hooks and filters directory? [/home/zspanina/git-extended/.git/hooks]
zspanina@zspanina:~/git-extended$ git branch
* develop
+ main
zspanina@zspanina:~/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/zspanina/git-extended/pull/new/develop
remote:
to github.com:zspanina/git-extended git
+ (new branch) develop -> develop
zspanina@zspanina:~/git-extended$ git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
zspanina@zspanina:~/git-extended$
```

Рис. 4.13: Конфигурация git-flow

7. Создаю релиз с версией 1.0.0 , создаю журнал изменений (рис. 4.14).

```
zspanina@zspanina:~/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'

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

Рис. 4.14: Создание релиза и журнала изменений

После этой команды всплывает окно, где нужно написать сообщение для коммита (рис. 4.15).

```
zspanina@zspanina:~/work/study/2024-2025/Операционные системы/os-intro x zspanina@zspanina:~/git-extended x
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
zspanina@zspanina:~/git-extended$ git add CHANGELOG.md
zspanina@zspanina:~/git-extended$ git commit -m 'chore(site): add changelog'
[release/1.0.0 c2b3b59] chore(site): add changelog
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md
zspanina@zspanina:~/git-extended$ git flow release finish 1.0.0

Переключились на ветку «main».
37a ветка counterpart «origin/main».
Merge made by the 'ort' strategy.
CHANGELOG.md | 9 ++++++++
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md
You are on 'main'.
Ваша ветка опережает «origin/main» на 2 коммита.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
Переключились на ветку «develop».
37a ветка counterpart «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 (6bba c2b3b59).

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
```

Рис. 4.15: Редактирование коммита

8. Заливаю релизную ветку в основную ветку (рис. 4.16).

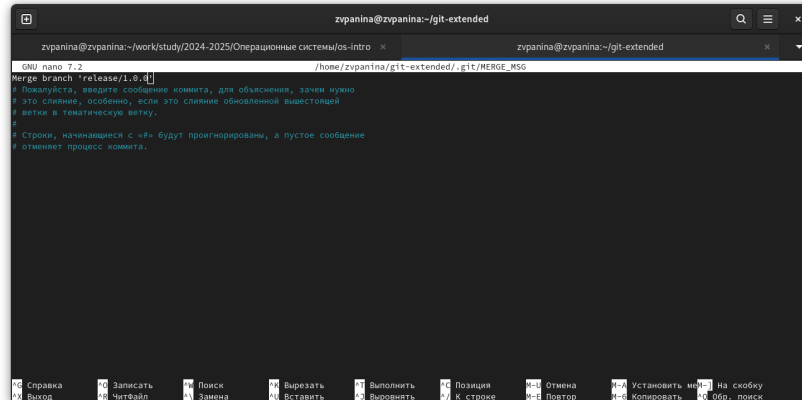


Рис. 4.16: Команда finish

9. Отправляю данные на github (рис. 4.17).

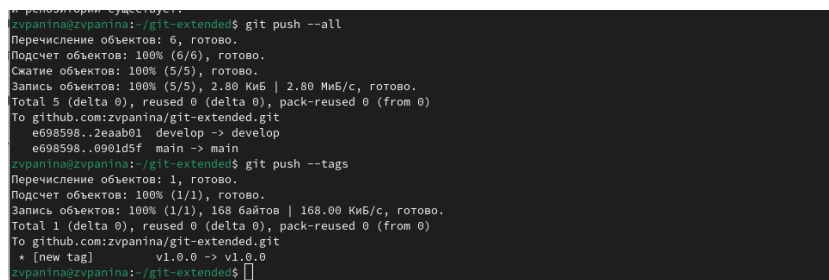


Рис. 4.17: Отправка данных на github

10. Создаю релиз на github. Для этого использую утилиты работы с github (Первая команда на рисунке 4.18) .

4.2.2 Работа с репозиторием git

1. Создаю ветку для новой функциональности. Объединяю ветку feature_branch с develop (рис. 4.18).

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

zspanina@zspanina:~/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 2eaab01).

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'

zspanina@zspanina:~/git-extended$
```

Рис. 4.18: Разработка новой функциональности

2. Создаю релиз с версией 1.2.3 и открываю файл package_json в mc (рис. 4.19).

```
zspanina@zspanina:~/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'

zspanina@zspanina:~/git-extended$ mc
zspanina@zspanina:~/git-extended$
```

Рис. 4.19: Создание релиза git-flow

3. В файле устанавливаю номер версии в 1.2.3 (рис. 4.20).

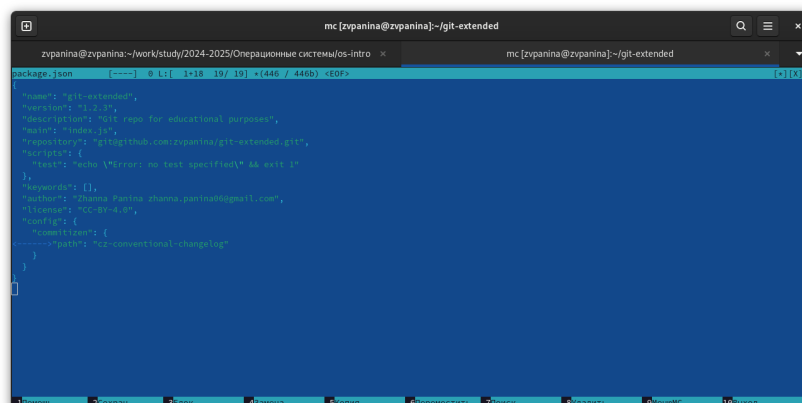


Рис. 4.20: Обновление номера версии в файле

4. Создаю журнал изменений, добавляю его в индекс (рис. 4.21).


```

zvpanina@zvpanina:~/git-extended$ standard-changelog
✓ output changes to CHANGELLOG.md
zvpanina@zvpanina:~/git-extended$ git add CHANGELLOG.md
zvpanina@zvpanina:~/git-extended$ git commit -am 'chore(site): update changelog'
[release/1.2.3 87edcf5] chore(site): update changelog
2 files changed, 5 insertions(+), 1 deletion(-)
zvpanina@zvpanina:~/git-extended$

```

Рис. 4.21: Создание журнала изменений

5. В всплывающем окне пишу сообщение для коммита (рис. 4.22).

The screenshot shows a terminal window with a nano editor open. The editor is editing a file named 'update_changelog'. The content of the file is:


```

#
# Введите сообщение для метки:
# v1.2.3
# Строки, начинающиеся с «#» будут проигнорированы.
  
```

 The window title is 'zvpanina@zvpanina:~/git-extended'.

Рис. 4.22: Редактирование коммита

6. Заливаю релизную ветку в основную ветку (рис. 4.23).

```

zvpanina@zvpanina:~/git-extended$ git flow release finish 1.2.3
Переключились на ветку «main»
Эта ветка соответствует «origin/main».
Merge made by the 'ort' strategy.
 CHANGELLOG.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.
 CHANGELLOG.md | 4 ++++
 package.json | 2 +-
 2 files changed, 5 insertions(+), 1 deletion(-)
Ветка release/1.2.3 удалена (была 87edcf5).

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'

zvpanina@zvpanina:~/git-extended$

```

Рис. 4.23: Команда finish

7. Отправляю данные на github. Создаю релиз на github с комментарием из журнала изменений (рис. 4.24).

```

zvpanina@zvpanina:~/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), 2.78 КиБ | 2.78 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 github.com:zvpanina/git-extended.git
   2eaab01..2164ade  develop -> develop
   0901d5f..922161a  main -> main
zvpanina@zvpanina:~/git-extended$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 169 байтов | 169.00 КиБ/с, готово.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:zvpanina/git-extended.git
 * [new tag]           v1.2.3 -> v1.2.3
zvpanina@zvpanina:~/git-extended$ gh release create v1.2.3 -F CHANGELOG.md
https://github.com/zvpanina/git-extended/releases/tag/v1.2.3
zvpanina@zvpanina:~/git-extended$

```

Рис. 4.24: Создание релиза на github

5 Выводы

В ходе выполнения лабораторной работы я приобрела навыки правильной работы с репозиториями git.

Список литературы

Рабочий процесс с Gitflow(электронный ресурс) URL: <https://yamadharma.github.io/ru/post/2021/04/18/gitflow-workflow/>