

Лабораторная работа №4

Операционные системы

Панина Ж. В.

07 марта 2025

Российский университет дружбы народов, Москва, Россия

Информация

- Панина Жанна Валерьевна
- НКАбд-02-24, студ. билет № 1132246710
- студент направления “Компьютерные и информационные науки”
- Российский университет дружбы народов
- 1132246710@pfur.ru
- https://github.com/zvpanina/study_2024-2025_os-intro

Вводная часть

В современной разработке программного обеспечения использование систем контроля версий, таких как Git, является неотъемлемой частью процесса. Однако, для эффективной работы в команде и поддержания порядка в кодовой базе, важно не только владеть базовыми навыками работы с Git, но и уметь применять более продвинутые методологии, такие как Git-flow и Conventional Commits. Эти подходы позволяют упростить управление ветками, улучшить читаемость истории коммитов и облегчить процесс выпуска версий. Освоение этих инструментов особенно актуально для студентов и начинающих разработчиков, так как они широко используются в индустрии.

Объект исследования:

- Процесс управления версиями в проекте с использованием системы контроля версий Git
- Применение методологий Git-flow и Conventional Commits для организации рабочего процесса.

Предмет исследования:

- Методология Git-flow и её применение для управления ветками в Git.
- Стандарт Conventional Commits и его использование для создания структурированной и понятной истории коммитов.
- Практические навыки работы с Git, включая создание репозитория, управление ветками, слияние изменений и разрешение конфликтов.

Цель работы - получение навыков правильной работы с репозиториями git.

Задачи:

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

- Локальный репозиторий Git.
- Документация по Git, Git-flow и Conventional Commits.
- Инструменты для работы с Git (GitHub).

Теоретическое введение

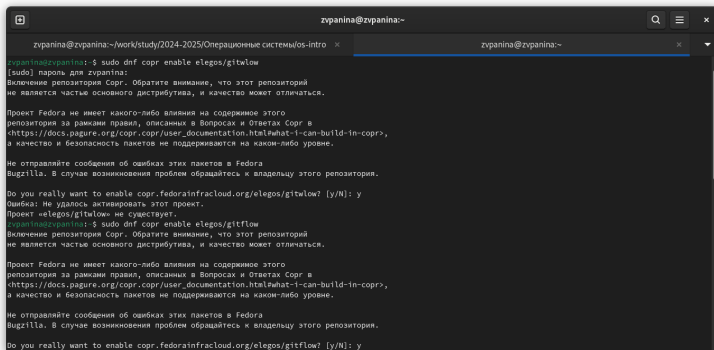
Gitflow Workflow опубликована и популяризована Винсентом Дриссенем. Gitflow Workflow предполагает выстраивание строгой модели ветвления с учётом выпуска проекта. Данная модель отлично подходит для организации рабочего процесса на основе релизов. Работа по модели Gitflow включает создание отдельной ветки 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 пререлизных и билд-метаданных возможны как дополнения к МАЖОРНАЯ.МИНОРНАЯ.ПАТЧ формату.

Выполнение лабораторной работы

Установка git-flow

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



```
zvpanina@zvpanina:~  
zvpanina@zvpanina:~/work/study/2024-2025/Операционные системы/os-intro  
zvpanina@zvpanina:~$ sudo dnf copr enable elegos/gitflow  
[sudo] пароль для zvpanina:  
Включение репозитория Copr. Обратите внимание, что этот репозиторий  
не является частью основного дистрибутива, и качество может отличаться.  
  
Проект Fedora не имеет никакого 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  
Ошибка: Не удалось активировать этот проект.  
Проект «elegos/gitflow» не существует.  
zvpanina@zvpanina:~$ 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 никакого 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. 1: Команда `dnf copr enable elegos/gitflow`

2. Выполняю команду `dnf install gitflow`.

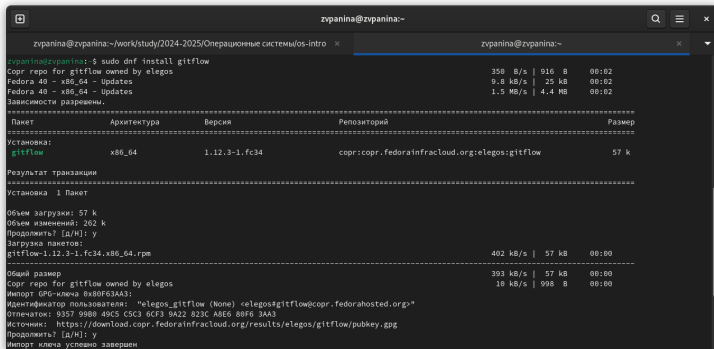
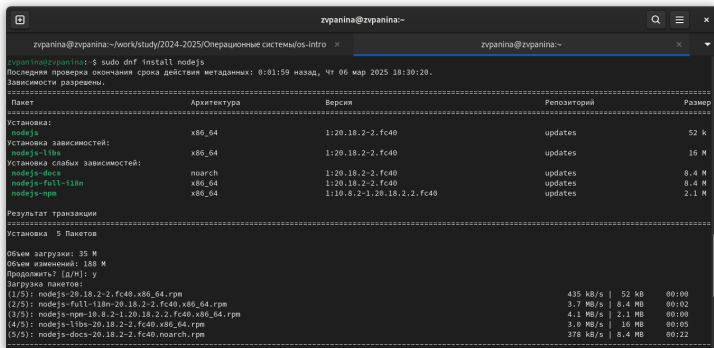


Рис. 2: Команда `dnf install gitflow`

Установка Node.js

1. Выполняю команду `dnf install nodejs` в режиме суперпользователя.



```
zvpanina@zvpanina:~  
zvpanina@zvpanina:~/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-libs	x86_64	1:20.18.2-2.fc40	updates	8.4 M
nodejs-npm	x86_64	1:10.8.2-1,20.18.2.2.fc40	updates	2.1 M

```
=====
```

Результат транзакции
Установка 5 Пакетов

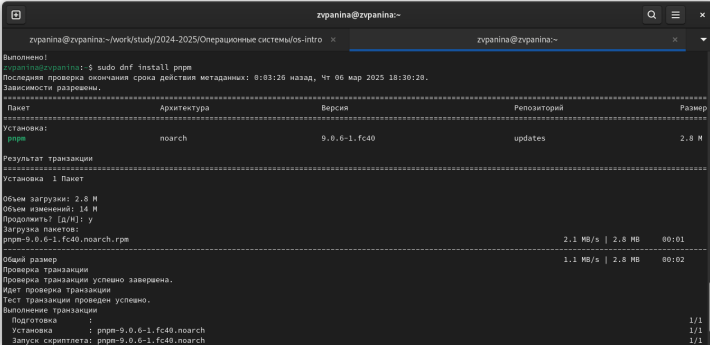
Объем загрузки: 35 М
Объем изменений: 188 М
Продолжить? [д/н]: y
Загрузка пакетов:

Пакет	Скорость	Объем	Время
(1/5): nodejs-20.18.2-2.fc40.x86_64.rpm	435 kB/s	52 kB	00:00
(2/5): nodejs-full-libs-20.18.2-2.fc40.x86_64.rpm	3.7 MB/s	8.4 MB	00:02
(3/5): nodejs-npm-10.8.2-1,20.18.2.2.fc40.x86_64.rpm	4.1 MB/s	2.1 MB	00:00
(4/5): nodejs-libs-20.18.2-2.fc40.x86_64.rpm	3.0 MB/s	16 MB	00:05
(5/5): nodejs-docs-20.18.2-2.fc40.noarch.rpm	378 kB/s	8.4 MB	00:22

```
=====
```

Рис. 3: Команда `dnf install nodejs`

2. Выполняю команду dnf install npm.



```
zvpanina@zvpanina:~  
zvpanina@zvpanina:~/work/study/2024-2025/Операционные системы/os-intro  
Выполнено!  
zvpanina@zvpanina:~$ sudo dnf install npm  
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: npm	noarch	9.0.6-1.fc40	updates	2.8 М

```
=====
```

Результат транзакции

Установка 1 Пакет

Объем загрузки: 2.8 М
Объем изменений: 14 М
Продолжить? [д/н]: y
Загрузка пакетов:
npm-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	:		1/1
Установка	:	npm-9.0.6-1.fc40.noarch	1/1
Запуск скрипглета:	:	npm-9.0.6-1.fc40.noarch	1/1

Рис. 4: Команда dnf install npm

Настройка Node.js

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

```
Выполнено!  
zvpanina@zvpanina:~$ pnpm setup  
Appended new lines to /home/zvpanina/.bashrc  
  
Next configuration changes were made:  
export PNPM_HOME="/home/zvpanina/.local/share/pnpm"  
case ":$PATH:" in  
  *)$PNPM_HOME:*) ;;  
  *) export PATH="$PNPM_HOME:$PATH" ;;  
esac  
  
To start using pnpm, run:  
source /home/zvpanina/.bashrc  
zvpanina@zvpanina:~$ source ~/.bashrc  
zvpanina@zvpanina:~$
```

Рис. 5: Настройка Node.js

Общепринятые коммиты

1. Данная программа используется для помощи в форматировании коммитов.

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

Update available! 9.0.6 → 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/zvanina/.local/share/pnpm/global/5:
+ commitizen 4.3.1

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

Рис. 6: Программа commitizen

2. Данная программа используется для помощи в создании логов.

```
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. 7: Программа standard-changelog

Создание репозитория git

1. Создаю репозиторий на GitHub и называю его git-extended.

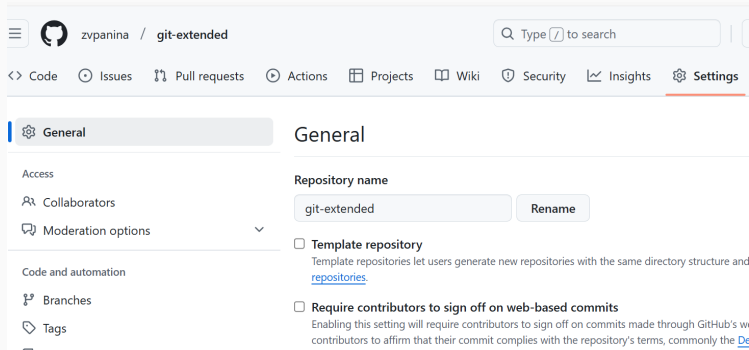


Рис. 8: Создание репозитория

2. Клонировать репозиторий, переходить 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.

```
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 ничему не соответствует
error: не удалось отправить некоторые ссылки в «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. 9: Первый коммит

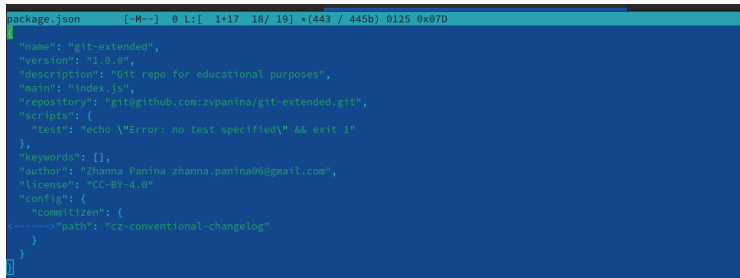
3. Создаю конфигурацию для пакетов Node.js и открываю файл git-extended в mc.

```
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. 10: Конфигурация общепринятых коммитов

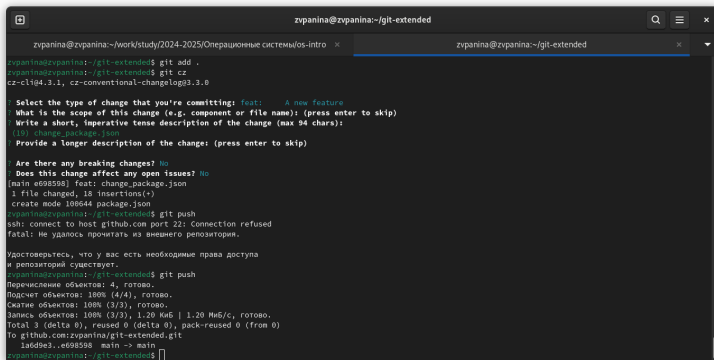
4. Заполняю несколько параметров пакета (название, лицензия, конфигурация пакета коммитов), чтобы файл выглядел вот так:

A screenshot of a code editor with a dark blue background. The title bar at the top reads "package.json" followed by a status bar showing "[-M--] 0 L: [1+17 18/ 19] *(443 / 445b) 0125 0x07D". The code is a JSON object for a package named "git-extended".

```
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:zvpanina/git-extended.git",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "keywords": [],
  "author": "Zhanna Panina zhanna.panina06@gmail.com",
  "license": "CC-BY-4.0"
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 11: Заполнение параметров пакета

5. С помощью следующих команд добавляю новые файлы, выполняю коммит, выкладываю на GitHub.



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

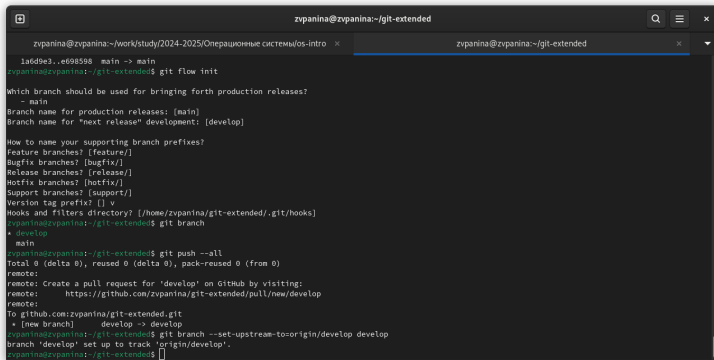
? Select the type of change that you're committing: feat:      A new feature
? What is the scope of this change (e.g. component or file name): (press enter to skip)
? Write a short, imperative tense description of the change (max 94 chars):
  (19) change_package.json
? Provide a longer description of the change: (press enter to skip)

? Are there any breaking changes? No
? Does this change affect any open issues? No
[main e698598] feat: change_package.json
1 file changed, 18 insertions(+)
create mode 100644 package.json
zvpanina@zvpanina:~/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.
zvpanina@zvpanina:~/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 КиБ | 1.20 МБ/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:zvpanina/git-extended.git
   1a6d9e3..e698598  main -> main
zvpanina@zvpanina:~/git-extended$
```

Рис. 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.



```
zspanina@zspanina:~/git-extended
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. 13: Конфигурация git-flow

7. Создаю релиз с версией 1.0.0 , создаю журнал изменений.

```
zvpanina@zvpanina:~/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'

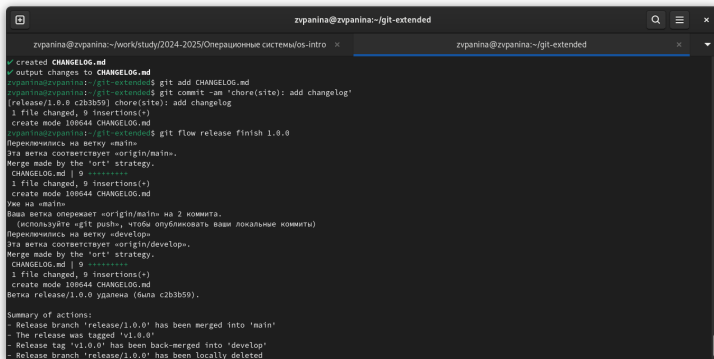
Follow-up actions:
- Bump the version number now!
- Start committing last-minute fixes in preparing your release
- When done, run:

    git flow release finish '1.0.0'

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

Рис. 14: Создание релиза и журнала изменений

После этой команды всплывает окно, где нужно написать сообщение для коммита.

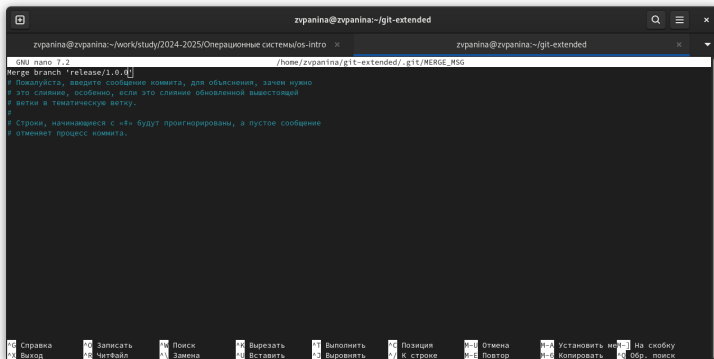


```
zvpanina@zvpanina:~/git-extended
zvpanina@zvpanina:~/work/study/2024-2025/Операционные системы/os-intro x zvpanina@zvpanina:~/git-extended
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
zvpanina@zvpanina:~/git-extended$ git add CHANGELOG.md
zvpanina@zvpanina:~/git-extended$ git commit -am 'chore(site): add changelog'
[release/1.0.0 c2b3b59] chore(site): add changelog
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md
zvpanina@zvpanina:~/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 (9d9a 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. 15: Редактирование коммита

8. Заливаю релизную ветку в основную ветку.



The screenshot shows a terminal window with a dark background. At the top, the title bar reads "zvpanina@zvpanina:~/git-extended". Below it, there are two tabs: "zvpanina@zvpanina:~/work/study/2024-2025/Операционные системы/os-intro" and "zvpanina@zvpanina:~/git-extended". The active tab is the second one. The terminal content shows the nano editor editing a file named "/home/zvpanina/git-extended/.git/MERGE_MSG". The text in the editor is as follows:

```
GNU nano 7.2 /home/zvpanina/git-extended/.git/MERGE_MSG
Merge branch 'release/1.0.0'
# Пожалуйста, введите сообщение коммита, для объяснения, зачем нужно
# это слияние, особенно, если это слияние обновленной выходящей
# ветки в тематическую ветку.
#
# Строки, начинающиеся с «#» будут проигнорированы, а пустое сообщение
# отменяет процесс коммита.
```

At the bottom of the terminal, there is a status bar with various keyboard shortcuts and their functions in Russian, such as "Справка", "Выход", "Записать", "ЧитФайл", "Поиск", "Замена", "Вырезать", "Вставить", "Выполнить", "Выровнять", "Позиция", "К строке", "Отмена", "Повтор", "Установить метку", "На скобку", "Копировать", and "Обр. поиск".

Рис. 16: Команда finish

9. Отправляю данные на github.

```
zvpanina@zvpanina:~/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), 2.80 КиБ | 2.80 МБ/с, готово.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:zvpanina/git-extended.git
   e698598..2eaab01  develop -> develop
   e698598..0901d5f  main -> main
zvpanina@zvpanina:~/git-extended$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 168 байтов | 168.00 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]         v1.0.0 -> v1.0.0
zvpanina@zvpanina:~/git-extended$
```

Рис. 17: Отправка данных на github

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

Работа с репозиторием git

1. Создаю ветку для новой функциональности. Объединяю ветку feature_branch с develop.

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

zvpanina@zvpanina:~/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'

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

Рис. 18: Разработка новой функциональности

2. Создаю релиз с версией 1.2.3 и открываю файл package_json в mc.

```
zvpanina@zvpanina:~/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'

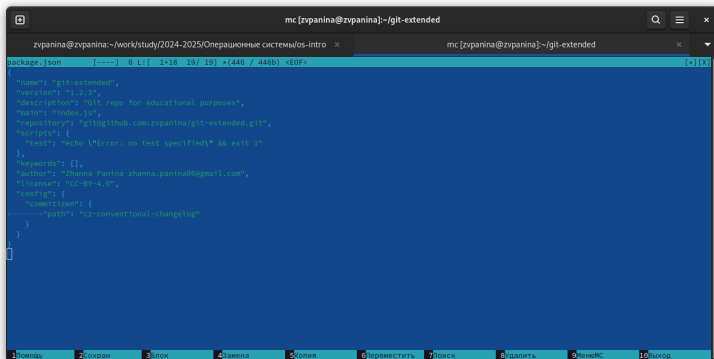
Follow-up actions:
- Bump the version number now!
- Start committing last-minute fixes in preparing your release
- When done, run:

    git flow release finish '1.2.3'

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

Рис. 19: Создание релиза git-flow

3. В файле устанавливаю номер версии в 1.2.3.



The screenshot shows a code editor window titled 'mc [zvpanina@zvpanina]:~/git-extended'. The editor displays the content of a 'package.json' file. The file is a JSON object with the following fields: 'name' (git-extended), 'version' (1.2.3), 'description' (git repo for educational purposes), 'main' (index.js), 'repository' (git@github.com:zvpanina/git-extended.git), 'scripts' (a test script that echoes an error message and exits with code 3), 'keywords' (an empty array), 'author' (Zhanna Panina), 'license' (CC-BY-4.0), and 'config' (a commitizen configuration with a path to 'cz-conventional-changelog'). The 'version' field is highlighted in blue. At the bottom of the editor, there is a toolbar with buttons for 'Помощь', 'Сохран', 'Блок', 'Замена', 'Копия', 'Перенести', 'Поиск', 'Удалить', 'Меню', and 'Выход'.

```
package.json [-----] 0 L: 1+18 19/ 19] + (446 / 446b) <EOF> [X]
{
  "name": "git-extended",
  "version": "1.2.3",
  "description": "git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:zvpanina/git-extended.git",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 3"
  },
  "keywords": [],
  "author": "Zhanna Panina <zhanna.panina@gmail.com>",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

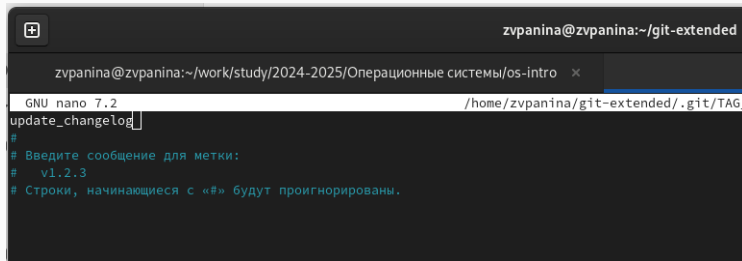
Рис. 20: Обновление номера версии в файле

4. Создаю журнал изменений, добавляю его в индекс.

```
zvpanina@zvpanina:~/git-extended$ standard-changelog
✓ output changes to CHANGELOG.md
zvpanina@zvpanina:~/git-extended$ git add CHANGELOG.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. 21: Создание журнала изменений

5. В всплывающем окне пишу сообщение для коммита.



The screenshot shows a terminal window with a dark background. At the top, the title bar reads "zvpanina@zvpanina:~/git-extended". Below it, a tab shows the current directory: "zvpanina@zvpanina:~/work/study/2024-2025/Операционные системы/os-intro". The main area of the terminal shows the GNU nano 7.2 editor editing the file "/home/zvpanina/git-extended/.git/TAG_update_changelog". The editor's status line at the bottom indicates the current mode: "# Введите сообщение для метки:" (Enter message for tag:). Below this, the text "v1.2.3" is entered, followed by a prompt "# Строки, начинающиеся с «#» будут проигнорированы." (Lines starting with «#» will be ignored.).

```
zvpanina@zvpanina:~/git-extended
zvpanina@zvpanina:~/work/study/2024-2025/Операционные системы/os-intro
GNU nano 7.2 /home/zvpanina/git-extended/.git/TAG_update_changelog
#
# Введите сообщение для метки:
#   v1.2.3
# Строки, начинающиеся с «#» будут проигнорированы.
```

Рис. 22: Редактирование коммита

6. Заливаю релизную ветку в основную ветку.

```
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.
  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 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. 23: Команда finish

7. Отправляю данные на github. Создаю релиз на github с комментарием из журнала изменений.

```
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 МиБ/с, готово.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com: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. 24: Создание релиза на github

В ходе выполнения лабораторной работы я приобрела навыки правильной работы с репозиториями git. Выполнила работу для тестового репозитория, преобразовала рабочий репозиторий в репозиторий с git-flow и conventional commits.

Рабочий процесс с Gitflow(электронный ресурс) URL:

<https://yamadharma.github.io/ru/post/2021/04/18/gitflow-workflow/>