

Отчёт по лабораторной работе №4

Операционные системы

Кудинец М. А.

Российский университет дружбы народов, Москва, Россия

Информация

- Кудинец Максим Антонович
- НКАбд-02-2024 № Студенческого билета: 1132246729
- Российский университет дружбы народов
- https://github.com/MaKudinets/study_2024-2025_os-intro

Цель работы

Получение навыков правильной работы с репозиториями git.

Задание

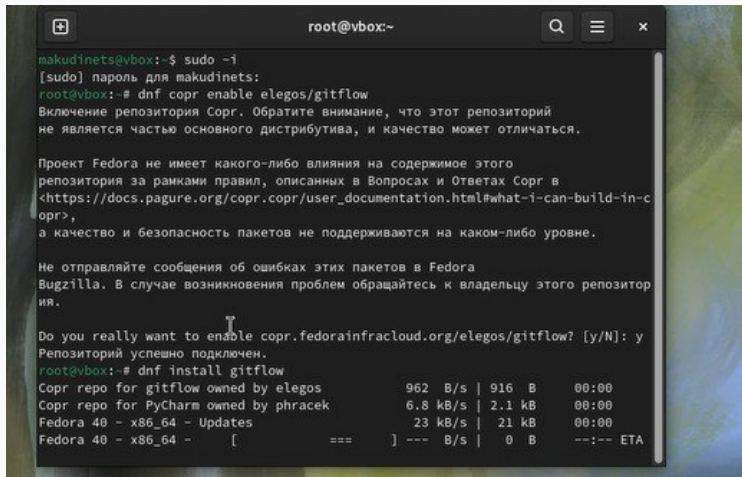
1. Выполнить работу для тестового репозитория.
2. Преобразовать рабочий репозиторий в репозиторий с git-flow и conventional commits.

Теоретическое введение

Рабочий процесс Gitflow Workflow. Будем описывать его с использованием пакета `git-flow`. Общая информация Gitflow Workflow опубликована и популяризована Винсентом Дриссенем. Gitflow Workflow предполагает выстраивание строгой модели ветвления с учётом выпуска проекта. Данная модель отлично подходит для организации рабочего процесса на основе релизов. Работа по модели Gitflow включает создание отдельной ветки для исправлений ошибок в рабочей среде. Последовательность действий при работе по модели Gitflow: Из ветки `master` создаётся ветка `develop`. Из ветки `develop` создаётся ветка `release`. Из ветки `develop` создаются ветки `feature`. Когда работа над веткой `feature` завершена, она сливается с веткой `develop`. Когда работа над веткой релиза `release` завершена, она сливается в ветки `develop` и `master`. Если в `master` обнаружена проблема, из `master` создаётся ветка `hotfix`. Когда работа над веткой исправления `hotfix` завершена, она сливается в ветки `develop` и `master`.

Выполнение лабораторной работы

1. Открываем терминал и входим в режим суперпользователя, устанавливаем gitflow.

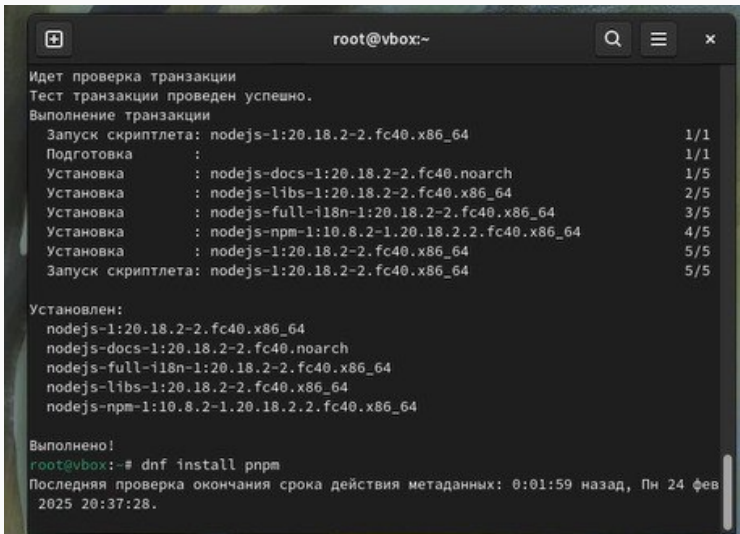


```
root@vbox:~  
makudinets@vbox:~$ sudo -i  
[sudo] пароль для makudinets:  
root@vbox:~# dnf copr enable elegos/gitflow  
Включение репозитория Copr. Обратите внимание, что этот репозиторий  
не является частью основного дистрибутива, и качество может отличаться.  
  
Проект Fedora не имеет какого-либо 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@vbox:~# dnf install gitflow  
Copr repo for gitflow owned by elegos          962 B/s | 916 B      00:00  
Copr repo for PyCharm owned by phracek        6.8 kB/s | 2.1 kB   00:00  
Fedora 40 - x86_64 - Updates                  23 kB/s | 21 kB     00:00  
Fedora 40 - x86_64 - [ === ] --- B/s | 0 B     ---:-- ETA
```

Рис. 1: Установка gitflow в режиме суперпользователя

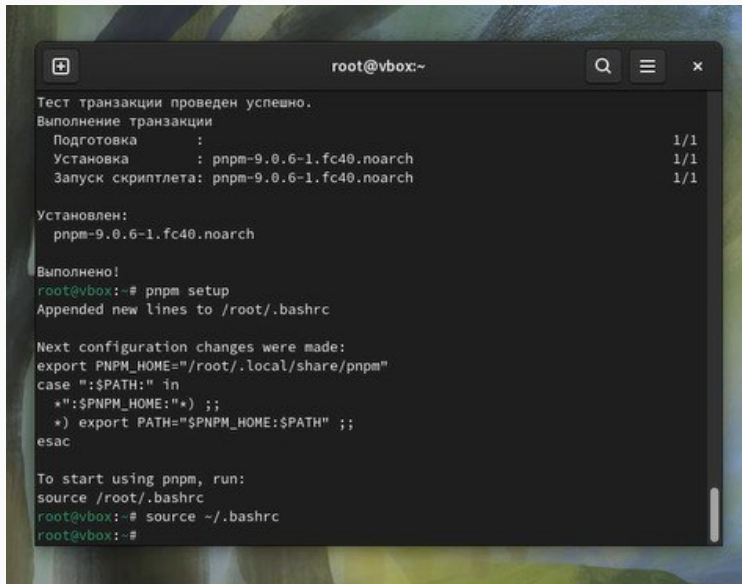
Установка и настройка node.js

2. Устанавливаем npm и nodejs.



```
root@vbox:~  
Идет проверка транзакции  
Тест транзакции проведен успешно.  
Выполнение транзакции  
Запуск скриптлета: nodejs-1:20.18.2-2.fc40.x86_64 1/1  
Подготовка : 1/1  
Установка : nodejs-docs-1:20.18.2-2.fc40.noarch 1/5  
Установка : nodejs-libs-1:20.18.2-2.fc40.x86_64 2/5  
Установка : nodejs-full-i18n-1:20.18.2-2.fc40.x86_64 3/5  
Установка : nodejs-npm-1:10.8.2-1.20.18.2.2.fc40.x86_64 4/5  
Установка : nodejs-1:20.18.2-2.fc40.x86_64 5/5  
Запуск скриптлета: nodejs-1:20.18.2-2.fc40.x86_64 5/5  
  
Установлен:  
nodejs-1:20.18.2-2.fc40.x86_64  
nodejs-docs-1:20.18.2-2.fc40.noarch  
nodejs-full-i18n-1:20.18.2-2.fc40.x86_64  
nodejs-libs-1:20.18.2-2.fc40.x86_64  
nodejs-npm-1:10.8.2-1.20.18.2.2.fc40.x86_64  
  
Выполнено!  
root@vbox:~# dnf install npm  
Последняя проверка окончания срока действия метаданных: 0:01:59 назад, Пн 24 фев  
2025 20:37:28.
```

3. Настраиваем nodejs.



A terminal window titled 'root@vbox:~' with search, menu, and close icons in the title bar. The terminal output shows the successful completion of a transaction to install pnpm. It lists the steps: preparation, installation of pnpm-9.0.6-1.fc40.noarch, and running the post-install script. Below this, it shows the command 'pnpm setup' being executed, which appends configuration to the .bashrc file. The configuration includes setting PNPM_HOME and updating the PATH. Finally, it shows the user sourcing the .bashrc file to apply the changes.

```
Тест транзакции проведен успешно.
Выполнение транзакции
  Подготовка      :                               1/1
  Установка       : pnpm-9.0.6-1.fc40.noarch       1/1
  Запуск скрипта  : pnpm-9.0.6-1.fc40.noarch       1/1

Установлен:
  pnpm-9.0.6-1.fc40.noarch

Выполнено!
root@vbox:~# 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@vbox:~# source ~/.bashrc
root@vbox:~#
```

Общепринятые коммиты.

4. Настраиваем commitizen и standard-changelog.

```
root@vbox:~# pnpm add -g commitizen

Update available! 9.0.6 → 10.4.1.
Changelog: https://github.com/pnpm/pnpm/releases/tag/v10.4.1
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

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

Done in 5.7s
root@vbox:~# pnpm add -g standard-changelog
```

Рис. 4: Настройка программ

5. Создаем репозиторий git, настраиваем его и делаем в него первый коммит.

```
makudinets@vbox:~ — git clone --recursive git@github.com:MaKudinets/...
makudinets@vbox:~$ git clone --recursive git@github.com:MaKudinets/git-extended.git
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 (from 0)
Получение объектов: 100% (3/3), готово.
makudinets@vbox:~$ git commit -m "first commit"
Текущая ветка: master

Начальный коммит

Неотслеживаемые файлы:
(используйте «git add <файл>...», чтобы добавить в то, что будет включено в коммит)
.bash_history
.bash_logout
.bash_profile
.bashrc
.cache/
.config/
.gitconfig
.gnupg/
.local/
.mozilla/
.ssh/
.texlive2023/
.vboxclient-clipboard-tty2-control.pid
.vboxclient-clipboard-tty2-service.pid
.vboxclient-draganddrop-tty2-control.pid
.vboxclient-draganddrop-tty2-service.pid
.vboxclient-hostversion-tty2-control.pid
.vboxclient-usb-lss-tty2-control.pid
```

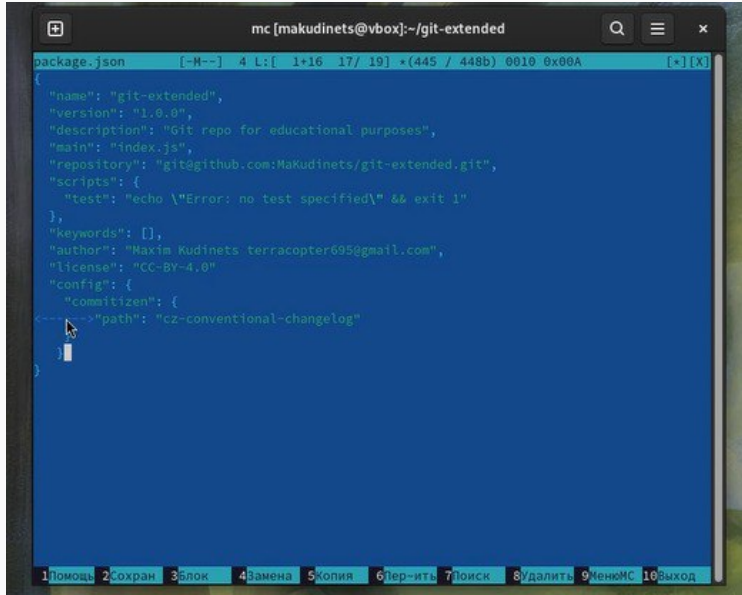
Рис. 5: Создание репозитория, первый коммит

6. Настраиваем пакет файлов nodejs. В файле `package.json` меняем необходимые данные.

```
makudinets@vbox:~/git-extended$ git push -u
branch 'main' set up to track 'origin/main'.
Everything up-to-date
makudinets@vbox:~/git-extended$ cd ~
makudinets@vbox:~$ pnpm init
Wrote to /home/makudinets/package.json

{
  "name": "makudinets",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
makudinets@vbox:~$
```

Рис. 6: Настройка пакета



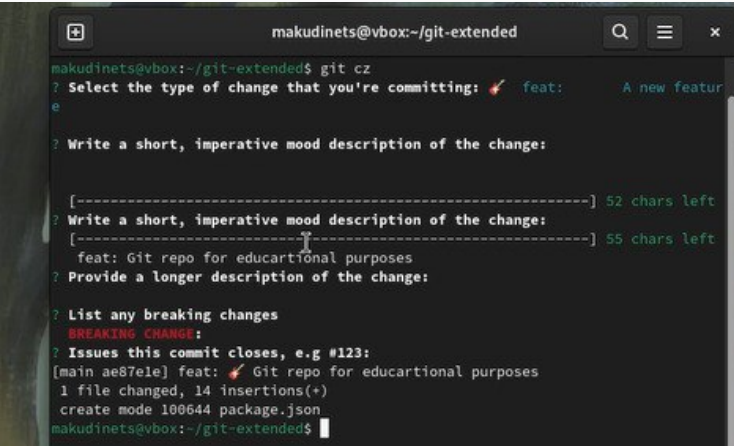
The image shows a code editor window titled "mc [makudinets@vbox]:~/git-extended". The editor displays the content of a file named "package.json". The file content is as follows:

```
package.json  [-M--]  4 L: [ 1+16 17/ 19] *(445 / 448b) 0010 0x00A [*] [X]
{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:MaKudinets/git-extended.git",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "keywords": [],
  "author": "Maxim Kudinets terracooper695@gmail.com",
  "license": "CC-BY-4.0"
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

A mouse cursor is positioned over the "path" property in the "commitizen" configuration object. At the bottom of the editor, there is a toolbar with the following buttons: 1 Помощь, 2 Сохран, 3 Блок, 4 Замена, 5 Копия, 6 Вер-ить, 7 Поиск, 8 Удалить, 9 МенюМС, 10 Выход.

Рис. 7: Изменения файла

7. Выполняем коммит. Выкладываем на github.



```
makudinets@vbox:~/git-extended$ git cz
? Select the type of change that you're committing: feat: A new featur
e

? Write a short, imperative mood description of the change:

[-----] 52 chars left
? Write a short, imperative mood description of the change:
[-----] 55 chars left
feat: Git repo for educartional purposes
? Provide a longer description of the change:

? List any breaking changes
BREAKING CHANGE:
? Issues this commit closes, e.g #123:
[main ae87ele] feat: Git repo for educartional purposes
1 file changed, 14 insertions(+)
create mode 100644 package.json
makudinets@vbox:~/git-extended$
```

Рис. 8: Выполнение коммита

```
makudinets@vbox:~/git-extended$ git push
Перечисление объектов: 4, готово.
Подсчет объектов: 100% (4/4), готово.
При сжатии изменений используется до 10 потоков
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 1.17 КиБ | 1.17 МБ/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:MaKudinets/git-extended.git
   aab92fc..ae87e1e  main -> main
makudinets@vbox:~/git-extended$ git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
```

Рис. 9: Команда push

8. Инициализируем gitflow, проверяем, на какой ветке мы находимся в данный момент, после чего загружаем весь репозиторий в хранилище.

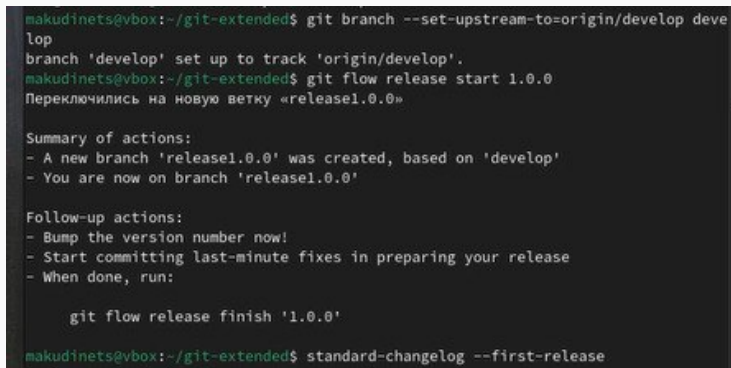
```
makudinets@vbox:~/git-extended$ git flow init

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

How to name your supporting branch prefixes?
Feature branches? [feature/] feature
Bugfix branches? [bugfix/] bugfix
Release branches? [release/] release
Hotfix branches? [hotfix/] hotfix
Support branches? [support/] support
Version tag prefix? [] v
Hooks and filters directory? [/home/makudinets/git-extended/.git/hooks]
makudinets@vbox:~/git-extended$ git branch
* develop
  main
makudinets@vbox:~/git-extended$
makudinets@vbox:~/git-extended$ git push --all
```

Рис. 10: Инициализация gitflow

9. Устанавливаем внешнюю ветку как вышестоящую для этой ветки. Создаем релиз с версией 1.0.0 и журнал изменений..



```
makudinets@vbox:~/git-extended$ git flow release --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
makudinets@vbox:~/git-extended$ git flow release start 1.0.0
Переключились на новую ветку «release1.0.0»

Summary of actions:
- A new branch 'release1.0.0' was created, based on 'develop'
- You are now on branch 'release1.0.0'

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'

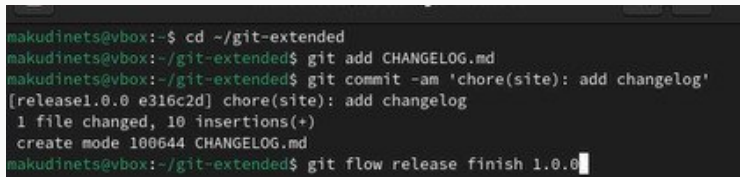
makudinets@vbox:~/git-extended$ standard-changelog --first-release
```

Рис. 11: Установка внешней ветки

```
makudinets@vbox:~/git-extended$ git flow release start 1.0.0
Fatal: There is an existing release branch '1.0.0'. Finish that one first.
makudinets@vbox:~/git-extended$ ls
CHANGELOG.md  package.json  README.md
makudinets@vbox:~/git-extended$ git add CHANGELOG.md'
```

Рис. 12: Создание релиза и журнала изменений

10. Заливаем релизную ветку в основную, добавляем журнал изменений в индекс, после чего заливаем релизную ветку в основную.



```
makudinets@vbox:~$ cd ~/git-extended
makudinets@vbox:~/git-extended$ git add CHANGELOG.md
makudinets@vbox:~/git-extended$ git commit -am 'chore(site): add changelog'
[release1.0.0 e316c2d] chore(site): add changelog
1 file changed, 10 insertions(+)
create mode 100644 CHANGELOG.md
makudinets@vbox:~/git-extended$ git flow release finish 1.0.0
```

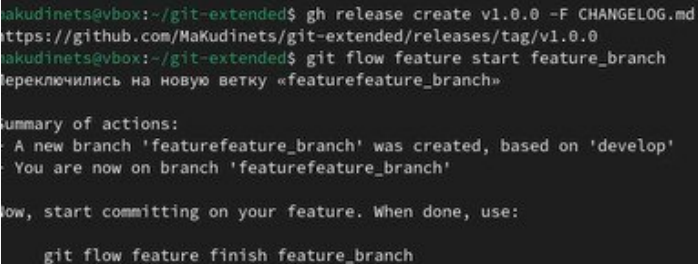
Рис. 13: Добавление релизной ветки в основную

11. Отправляем данные и теги на гитхаб.

```
makudinets@vbox:~/git-extended$ git push --all
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
При сжатии изменений используется до 10 потоков
Сжатие объектов: 100% (4/4), готово.
Запись объектов: 100% (4/4), 1.97 КиБ | 1.97 МиБ/с, готово.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:MaKudinets/git-extended.git
   ae87ele..5c88867  main -> main
   * [new branch]      releasel.0.0 -> releasel.0.0
makudinets@vbox:~/git-extended$ git push --tags
```

Рис. 14: Команды push -all и push -tags

12. Создаем релиз на гитхабе. Создаем ветку для новой функциональности.

A terminal window with a dark background and light green text. It shows a sequence of commands and their outputs. The first command is 'gh release create v1.0.0 -F CHANGELOG.md' which outputs a GitHub release URL. The second command is 'git flow feature start feature_branch' which outputs a message about switching to a new branch. The third part shows a 'Summary of actions:' followed by two bullet points: 'A new branch 'featurefeature_branch' was created, based on 'develop'' and 'You are now on branch 'featurefeature_branch''. The final part shows the command 'git flow feature finish feature_branch' without an output.

```
makudinets@vbox:~/git-extended$ gh release create v1.0.0 -F CHANGELOG.md
https://github.com/MaKudinets/git-extended/releases/tag/v1.0.0
makudinets@vbox:~/git-extended$ git flow feature start feature_branch
Переключились на новую ветку «featurefeature_branch»

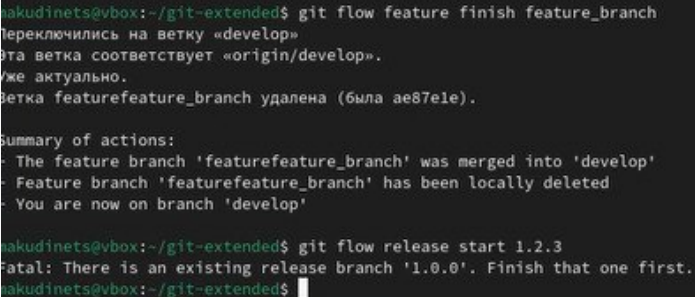
Summary of actions:
- A new branch 'featurefeature_branch' was created, based on 'develop'
- You are now on branch 'featurefeature_branch'

Now, start committing on your feature. When done, use:

    git flow feature finish feature_branch
```

Рис. 15: Создание новой ветки

13. Объединяем новую ветку с develop. Создаём релиз с версией 1.2.3.

A terminal window with a dark background and light green text. The prompt is 'aakudinets@vbox:~/git-extended\$'. The first command is 'git flow feature finish feature_branch'. The output shows a message in Russian: 'Переключились на ветку «develop»', 'Эта ветка соответствует «origin/develop».', 'Все актуально.', and 'Ветка featurefeature_branch удалена (была ae87ele)'. Then a 'Summary of actions:' is shown with three bullet points: '- The feature branch 'featurefeature_branch' was merged into 'develop'', '- Feature branch 'featurefeature_branch' has been locally deleted', and '- You are now on branch 'develop''. The second command is 'git flow release start 1.2.3'. The output is an error: 'fatal: There is an existing release branch '1.0.0'. Finish that one first.' The prompt returns to 'aakudinets@vbox:~/git-extended\$' with a cursor.

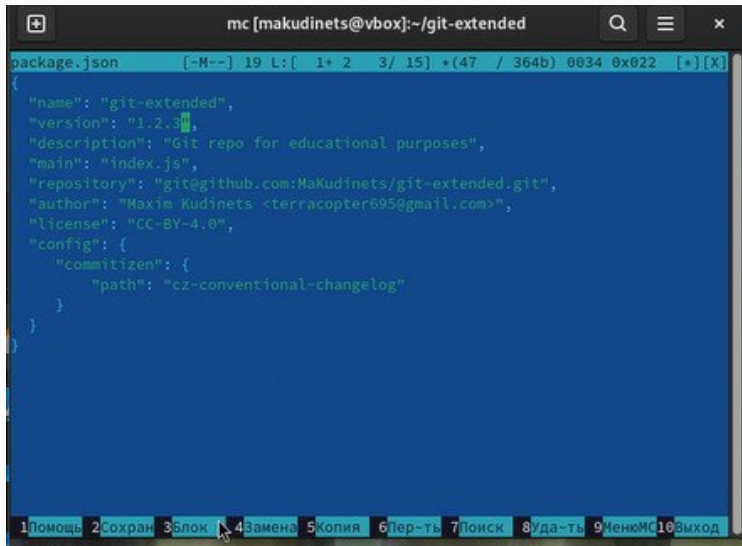
```
aakudinets@vbox:~/git-extended$ git flow feature finish feature_branch
Переключились на ветку «develop»
Эта ветка соответствует «origin/develop».
Все актуально.
Ветка featurefeature_branch удалена (была ae87ele).

Summary of actions:
- The feature branch 'featurefeature_branch' was merged into 'develop'
- Feature branch 'featurefeature_branch' has been locally deleted
- You are now on branch 'develop'

aakudinets@vbox:~/git-extended$ git flow release start 1.2.3
fatal: There is an existing release branch '1.0.0'. Finish that one first.
aakudinets@vbox:~/git-extended$
```

Рис. 16: Объединение веток, создание релиза с более новой версией

14. Изменяем номер версии в файле package.json.



The screenshot shows a terminal window with the title bar "mc [makudinets@vbox]:~/git-extended". The file "package.json" is open, and its content is displayed on a blue background. The "version" field is currently "1.0.0" and is being edited to "1.2.3". The terminal window has a status bar at the bottom with menu items: 1Помощь, 2Сохран, 3Блок, 4Замена, 5Копия, 6Пер-ть, 7Поиск, 8Уда-ть, 9МенюМС, 10Выход.

```
package.json [-M--] 19 L:[ 1+ 2 3/ 15] *(47 / 364b) 0034 0x022 [*][X]
{
  "name": "git-extended",
  "version": "1.2.3",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:MaKudinets/git-extended.git",
  "author": "Maxim Kudinets <terracooper695@gmail.com>",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 17: Изменение номера версии с 1.0.0 на 1.2.3

15. Заливаем релизную ветку в основную. Отправляем данные на гитхаб.

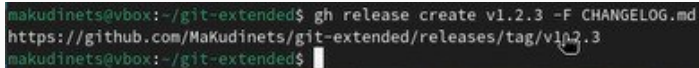
```
akudinets@vbox:~/git-extended$ git flow release finish 1.2.3
ranches 'release1.2.3' and 'origin/release1.2.3' have diverged.
nd local branch 'release1.2.3' is ahead of 'origin/release1.2.3'.
ranches 'main' and 'origin/main' have diverged.
nd local branch 'main' is ahead of 'origin/main'.
же на «main»
аша ветка опережает «origin/main» на 4 коммита.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
ерекнулись на ветку «develop»
та ветка соответствует «origin/develop».
erge made by the 'ort' strategy.
CHANGELOG.md | 8 +++++++
package.json | 2 +-
2 files changed, 9 insertions(+), 1 deletion(-)
```

Рис. 18: Добавление релизной ветки в основную


```
kudinets@vbox:~/git-extended$ git push --all
счисление объектов: 12, готово.
счет объектов: 100% (12/12), готово.
и сжатии изменений используется до 10 потоков
ание объектов: 100% (9/9), готово.
пись объектов: 100% (9/9), 3.93 КиБ | 3.93 МиБ/с, готово.
tal 9 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
note: Resolving deltas: 100% (2/2), completed with 1 local object.
github.com:MaKudinets/git-extended.git
714dfcc..68c887f develop -> develop
5c88867..7e6e62d main -> main
kudinets@vbox:~/git-extended$ git push --tags
```

Рис. 19: Отправка данных на гитхаб

16. Создаём релиз на гитхабе с комментарием из журнала изменений.

A terminal window with a dark background and green text. The prompt is 'makudinets@vbox:~/git-extended\$'. The command entered is 'gh release create v1.2.3 -F CHANGELOG.md'. The output is 'https://github.com/MaKudinets/git-extended/releases/tag/v1.2.3'. The prompt is repeated on the next line with a cursor.

```
makudinets@vbox:~/git-extended$ gh release create v1.2.3 -F CHANGELOG.md
https://github.com/MaKudinets/git-extended/releases/tag/v1.2.3
makudinets@vbox:~/git-extended$
```

Рис. 20: Создание релиза на гитхабе с необходимым комментарием

В процессе выполнения лабораторной работы я приобрел навыки правильной работы с репозиториями git.