

Отчет по лабораторной работе №4

Операционные системы

Кузьмин Егор Витальевич

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
4.1	Установка и настройка программного обеспечения	8
4.2	Общепринятые коммиты	11
4.3	Работа с репозиторием git	14
5	Выводы	16
	Список литературы	17

Список иллюстраций

4.1	Установка пакетов	8
4.2	Выполнение загрузки	8
4.3	Установка пакетов	9
4.4	Выполнение загрузки	9
4.5	Установка пакетов	9
4.6	Настройка	10
4.7	Настройка и установка	10
4.8	Установка добавочных программ	10
4.9	Установка добавочных программ	11
4.10	Создание репозитория	11
4.11	конфигурация формата коммитов	11
4.12	Работа с коммитами	12
4.13	Инициализация	12
4.14	Работа с ветками, отправка в репозиторий	13
4.15	Создание релиза и сопутствующие действия	13
4.16	Отправка данных	13
4.17	Создание и слияние веток	14
4.18	Создание релиза	14
4.19	Работа с журналом изменений	14
4.20	Работа с ветками	15
4.21	Отправка данных, создание релиза	15

Список таблиц

1 Цель работы

Цель данной лабораторной работы - получение навыков правильной работы с репозиториями git.

2 Задание

0. Ознакомиться с заданием и дополнительным программным обеспечением.
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

Данная модель отлично подходит для организации рабочего процесса на основе релизов.

Работа по модели Gitflow включает создание отдельной ветки для исправлений ошибок в р

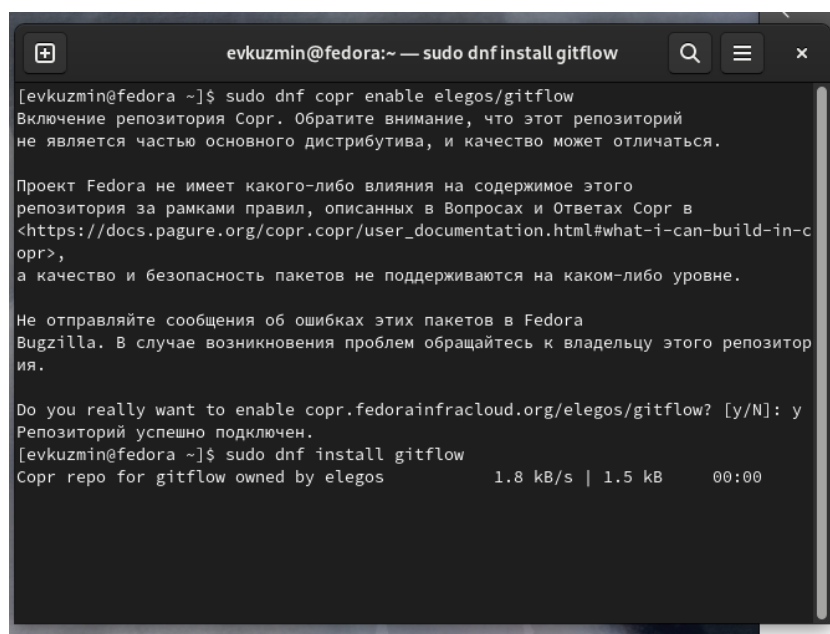
Последовательность действий при работе по модели Gitflow:

1. Из ветки master создаётся ветка develop.
2. Из ветки develop создаётся ветка release.
3. Из ветки develop создаются ветки feature.
4. Когда работа над веткой feature завершена, она сливается с веткой develop.
5. Когда работа над веткой релиза release завершена, она сливается в ветки develop и m
6. Если в master обнаружена проблема, из master создаётся ветка hotfix.
7. Когда работа над веткой исправления hotfix завершена, она сливается в ветки develop

4 Выполнение лабораторной работы

4.1 Установка и настройка программного обеспечения

Устанавливаю git-flow (рис. 1).



```
evkuzmin@fedora:~ — sudo dnf install gitflow

[evkuzmin@fedora ~]$ sudo dnf copr enable elegos/gitflow
Включение репозитория Copr. Обратите внимание, что этот репозиторий
не является частью основного дистрибутива, и качество может отличаться.

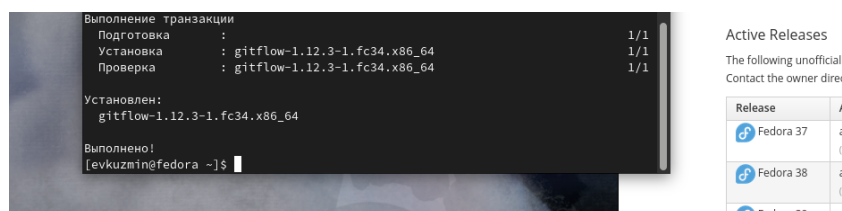
Проект Fedora не имеет какого-либо влияния на содержимое этого
репозитория за рамками правил, описанных в Вопросах и Ответах Copr в
<https://docs.pagure.org/copr.copr/user_documentation.html#what-i-can-build-in-copr>,
а качество и безопасность пакетов не поддерживаются на каком-либо уровне.

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
Репозиторий успешно подключен.
[evkuzmin@fedora ~]$ sudo dnf install gitflow
Copr repo for gitflow owned by elegos          1.8 kB/s | 1.5 kB      00:00
```

Рис. 4.1: Установка пакетов

Видим сообщение об успешности загрузки (рис. 2)



```
Выполнение транзакции
Подготовка      : 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!
[evkuzmin@fedora ~]$
```

Active Releases

The following unofficial
Contact the owner direc

Release	
Fedora 37	
Fedora 38	
Fedora 39	

Рис. 4.2: Выполнение загрузки

Устанавливаю Node.js (рис. 3).

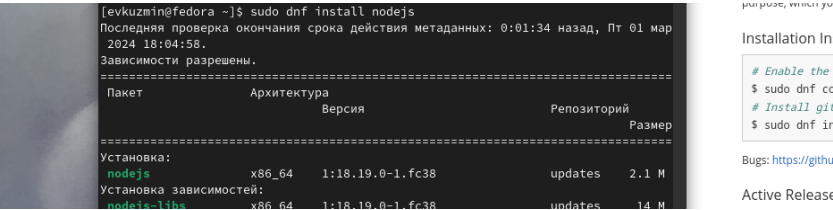


Рис. 4.3: Установка пакетов

Видим сообщение об успешности загрузки (рис. 4)

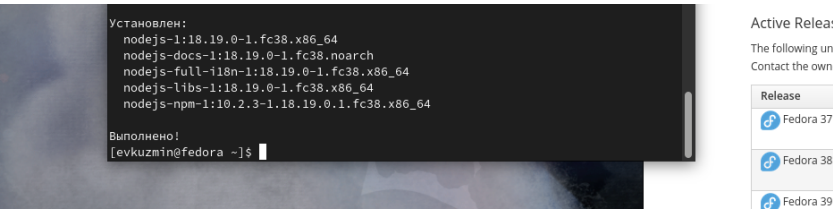


Рис. 4.4: Выполнение загрузки

Устанавливаю npm (рис. 5)

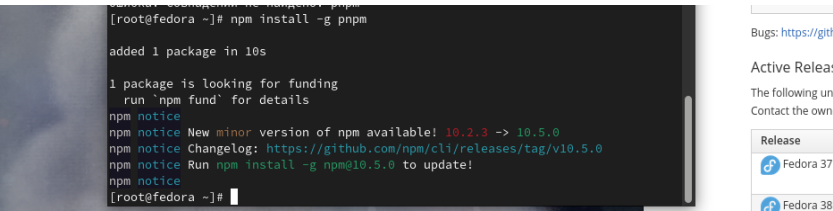
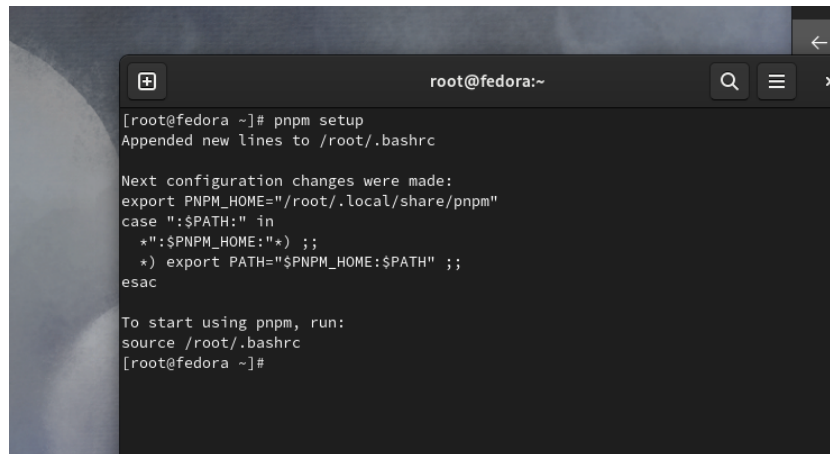


Рис. 4.5: Установка пакетов

Начинаю настройку Node.js (рис. 6)

A terminal window titled 'root@fedora:~' showing the output of the 'pnpm setup' command. The output indicates that new lines have been appended to the root .bashrc file and lists the configuration changes made to the environment variables. It also provides instructions on how to start using pnpm by sourcing the .bashrc file.

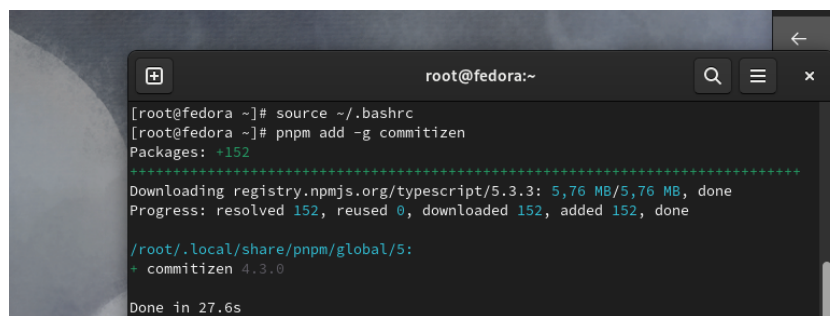
```
[root@fedora ~]# pnpm setup
Appended new lines to /root/.bashrc

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@fedora ~]#
```

Рис. 4.6: Настройка

Завершаю настройку Node.js и устанавливаю программу для помощи в форматировании коммитов (рис. 7)

A terminal window titled 'root@fedora:~' showing the installation of the 'commitizen' package globally using pnpm. The output shows the progress of downloading and installing the package, including the number of packages resolved, reused, downloaded, and added.

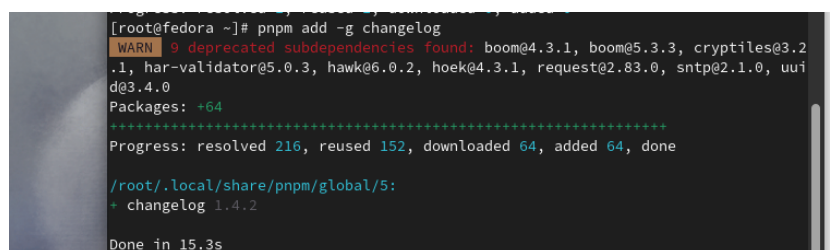
```
[root@fedora ~]# source ~/.bashrc
[root@fedora ~]# pnpm add -g commitizen
Packages: +152
+-----+
Downloading registry.npmjs.org/typescript/5.3.3: 5,76 MB/5,76 MB, done
Progress: resolved 152, reused 0, downloaded 152, added 152, done

/root/.local/share/pnpm/global/5:
+ commitizen 4.3.0

Done in 27.6s
```

Рис. 4.7: Настройка и установка

Добавляем программу для помощи в создании логов (рис. 8)

A terminal window titled 'root@fedora:~' showing the installation of the 'changelog' package globally using pnpm. The output shows a warning about deprecated subdependencies, followed by the progress of downloading and installing the package.

```
[root@fedora ~]# pnpm add -g changelog
WARN 9 deprecated subdependencies found: boom@4.3.1, boom@5.3.3, cryptiles@3.2
.1, har-validator@5.0.3, hawk@6.0.2, hoek@4.3.1, request@2.83.0, sntp@2.1.0, uui
d@3.4.0
Packages: +64
+-----+
Progress: resolved 216, reused 152, downloaded 64, added 64, done

/root/.local/share/pnpm/global/5:
+ changelog 1.4.2

Done in 15.3s
```

Рис. 4.8: Установка добавочных программ

Устанавливаем пакет “стандарт” (рис. 9)

```
[root@fedora ~]# pnpm add -g standart changelog
WARN 10 deprecated subdependencies found: boom@4.3.1, boom@5.3.3, circular-js
on@0.3.3, cryptiles@3.2.1, har-validator@5.0.3, hawk@6.0.2, hoek@4.3.1, request
@2.83.0, sntp@2.1.0, uuid@3.4.0
Packages: +146
Progress: resolved 362, reused 216, downloaded 146, added 146, done
.pnpm/es5-ext@0.10.64/node_modules/es5-ext: Running postinstall script, done in
493ms

/root/.local/share/pnpm/global/5:
+ standart 6.1.0

Done in 24.4s
```

Рис. 4.9: Установка добавочных программ

4.2 Общепринятые коммиты

Создаю репозиторий (рис. 10).

```
[evkuzmin@fedora git-extended]$ gh repo create git-extended --public
✓ Created repository evkuzmin1/git-extended on GitHub
[evkuzmin@fedora git-extended]$
```

Рис. 4.10: Создание репозитория

Делаю первый коммит и выкладываю его на гитхаб.

Конфигурирую пакеты Node.js - формат коммитов (рис. 11)

```
1 {
2   "name": "git-extended",
3   "version": "1.0.0",
4   "description": "Git repo for educational purposes",
5   "main": "index.js",
6   "repository": "github.com:evkuzmin1/git-extended.git",
7   "author": "Kuzmin Egor <132236046@pfur.ru>",
8   "license": "CC-BY-4.0",
9   "config": {
10    "commitizen": {
11      "path": "cz-conventional-changelog"
12    }
13  }
14 }
```

Рис. 4.11: конфигурация формата коммитов

Добавляю новые файлы, выполняю коммит (рис. 12)

```
evkuzmin@fedora:~/work/git-extended/git-extended
[evkuzmin@fedora git-extended]$ git cz
cz-cli@4.3.0, cz-conventional-changelog@3.3.0

? Select the type of change that you're committing: chore:   Other changes that
don't modify src or test files
? 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 93 chars):
(3) idk
? 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 5048a51] chore: idk
1 file changed, 14 insertions(+)
 create mode 100644 package.json
[evkuzmin@fedora 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), 1.14 КиБ | 583.00 КиБ/с, готово.
Всего 3 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To https://github.com/evkuzmin1/git-extended
 60356b6..5048a51  main -> main
[evkuzmin@fedora git-extended]$
```

Рис. 4.12: Работа с коммитами

Инициализирую git-flow (рис. 13)

```
[evkuzmin@fedora git-extended]$ git flow init

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

Which branch should be used for integration of the "next release"?
- main
Branch name for "next release" development: [] develop
Fatal: Production and integration branches should differ.
[evkuzmin@fedora git-extended]$ git flow init

Which branch should be used for integration of the "next release"?
- main
Branch name for "next release" development: [] main

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? [] []
Hooks and filters directory? [/home/evkuzmin/work/git-extended/git-extended/.git/hooks]
[evkuzmin@fedora git-extended]$ git flow init
Already initialized for gitflow.
To force reinitialization, use: git flow init -f
[evkuzmin@fedora git-extended]$
```

Рис. 4.13: Инициализация

Проверяю ветку и загружаю весь репозиторий в хранилище. Далее устанавливаю внешнюю ветку как вышестоящую (рис. 14)

```

[evkuzmin@fedora git-extended]$ git branch
* develop
  main
[evkuzmin@fedora git-extended]$ git push --all
Всего 0 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/evkuzmin1/git-extended/pull/new/develop
remote:
To https://github.com/evkuzmin1/git-extended
 * [new branch]      develop -> develop
[evkuzmin@fedora git-extended]$ git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.

```

Рис. 4.14: Работа с ветками, отправка в репозиторий

Создаю релиз с версией 1.0.0, создаю журнал изменений, добавляю журнал изменений в индекс. Заливаю релизную ветку в основную (рис. 15)

```

[evkuzmin@fedora 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 'main'
- 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'

[evkuzmin@fedora git-extended]$

```

Рис. 4.15: Создание релиза и сопутствующие действия

Отправляю все данные в git. Создаю на нём релиз (рис. 16)

```

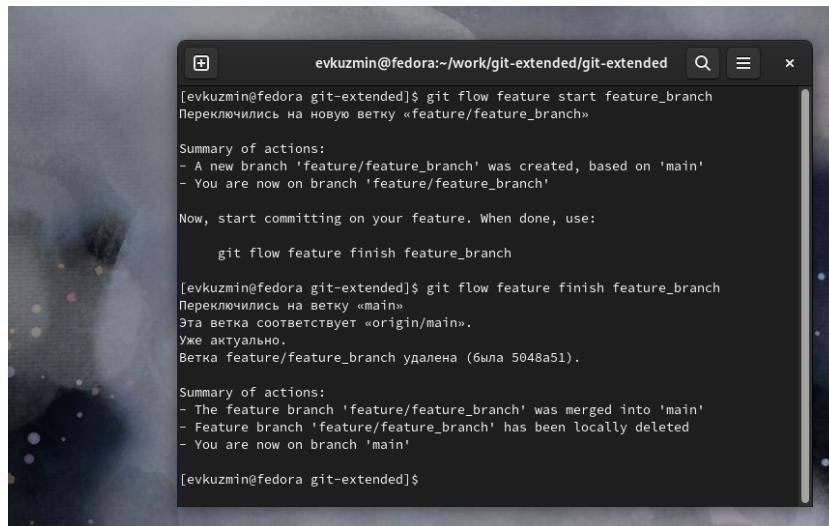
evkuzmin@fedora:~/work/git-extended/git-extended
[evkuzmin@fedora git-extended]$ git push --all
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
Сжатие объектов: 100% (4/4), готово.
Запись объектов: 100% (4/4), 1.96 КиБ | 1.96 МБ/с, готово.
Всего 4 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To https://github.com/evkuzmin1/git-extended
 5048a51..bddeff7 develop -> develop
 * [new branch]      release/1.0.0 -> release/1.0.0
[evkuzmin@fedora git-extended]$ git push --tags
Everything up-to-date
[evkuzmin@fedora git-extended]$ gh release create v1.0.0 -F CHANGELOG.md
https://github.com/evkuzmin1/git-extended/releases/tag/v1.0.0
[evkuzmin@fedora git-extended]$

```

Рис. 4.16: Отправка данных

4.3 Работа с репозиторием git

Создаю ветку для новой функциональности. Далее нам потребуется объединить эти ветки (рис. 17)



```
evkuzmin@fedora:~/work/git-extended/git-extended
[evkuzmin@fedora 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 'main'
- You are now on branch 'feature/feature_branch'

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

    git flow feature finish feature_branch

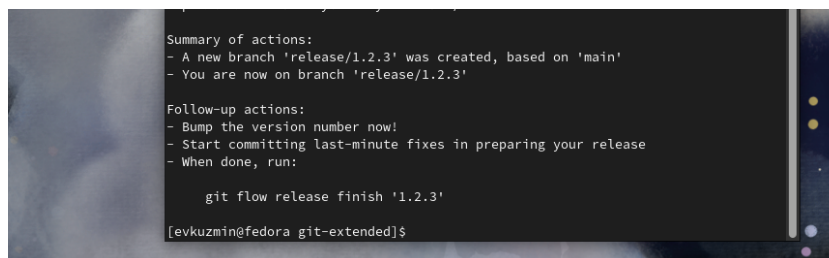
[evkuzmin@fedora git-extended]$ git flow feature finish feature_branch
Переключились на ветку «main»
Эта ветка соответствует «origin/main».
Уже актуально.
Ветка feature/feature_branch удалена (была 5048a51).

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

[evkuzmin@fedora git-extended]$
```

Рис. 4.17: Создание и слияние веток

Создаю релиз с версией 1.2.3 (рис. 18)



```
Summary of actions:
- A new branch 'release/1.2.3' was created, based on 'main'
- You are now on branch 'release/1.2.3'

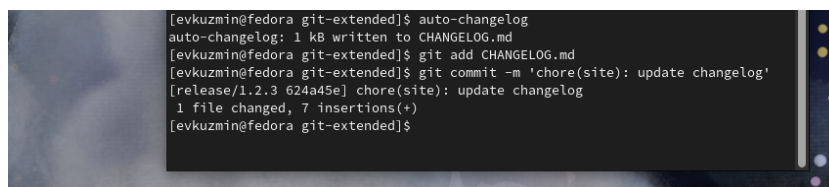
Follow-up actions:
- Bump the version number now!
- Start committing last-minute fixes in preparing your release
- When done, run:

    git flow release finish '1.2.3'

[evkuzmin@fedora git-extended]$
```

Рис. 4.18: Создание релиза

Создаю журнал изменений, добавляю журнал изменений в индекс (рис. 19)



```
[evkuzmin@fedora git-extended]$ auto-changelog
auto-changelog: 1 KB written to CHANGELOG.md
[evkuzmin@fedora git-extended]$ git add CHANGELOG.md
[evkuzmin@fedora git-extended]$ git commit -m 'chore(site): update changelog'
[release/1.2.3 624a45e] chore(site): update changelog
1 file changed, 7 insertions(+)
[evkuzmin@fedora git-extended]$
```

Рис. 4.19: Работа с журналом изменений

Затем перемещаю релизную ветку в основную ветку (рис. 20)

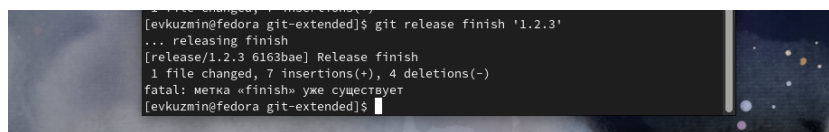
A terminal window with a dark background and a space-themed wallpaper on the right. The terminal shows the command 'git release finish '1.2.3'' being executed. The output includes '... releasing finish', '[release/1.2.3 6163bae] Release finish', and '1 file changed, 7 insertions(+), 4 deletions(-)'. A fatal error message follows: 'fatal: метка «finish» уже существует'. The prompt returns to '[evkuzmin@fedora git-extended]\$'.

Рис. 4.20: Работа с ветками

Отправляю все данные на github и сразу же создаю релиз на github с комментарием из журнала изменений (рис. 21)

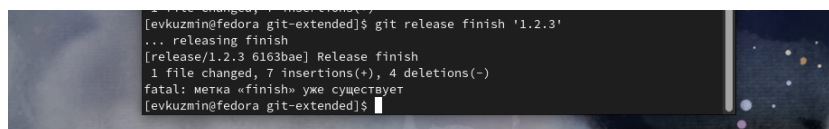
A terminal window with a dark background and a space-themed wallpaper on the right. The terminal shows the command 'git release finish '1.2.3'' being executed. The output includes '... releasing finish', '[release/1.2.3 6163bae] Release finish', and '1 file changed, 7 insertions(+), 4 deletions(-)'. A fatal error message follows: 'fatal: метка «finish» уже существует'. The prompt returns to '[evkuzmin@fedora git-extended]\$'.

Рис. 4.21: Отправка данных, создание релиза

5 Выводы

Я приобрел практический опыт работы репозиториями git на продвинутом уровне, научился доустанавливать нужные пакеты и настраивать их для определенных задач.

Список литературы

Архитектура компьютеров и ОС/Электронный ресурс