

# **Лабораторная работа №4**

**Операционные системы**

Ермакова Анастасия Алексеевна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
4.1	Установка git-flow . . . . .	8
4.2	Установка Node.js . . . . .	8
4.3	Настройка Node.js . . . . .	9
4.4	Общепринятые коммиты . . . . .	9
4.5	Создание репозитория git . . . . .	10
4.6	Работа с репозиторием git . . . . .	15
<b>5</b>	<b>Выводы</b>	<b>18</b>
<b>6</b>	<b>Список литературы</b>	<b>19</b>

# Список иллюстраций

4.1	Установка git-flow . . . . .	8
4.2	Установка Node.js . . . . .	9
4.3	Настройка Node.js . . . . .	9
4.4	Настройка коммитов . . . . .	9
4.5	Настройка коммитов . . . . .	10
4.6	Создание репозитория . . . . .	10
4.7	Конфигурация коммитов . . . . .	10
4.8	Конфигурация коммитов . . . . .	11
4.9	Открытие файла . . . . .	11
4.10	Редактирование файла . . . . .	12
4.11	Добавление файлов . . . . .	12
4.12	Отправление файлов . . . . .	12
4.13	Инициализация git-flow . . . . .	13
4.14	Загрузка репозитория . . . . .	13
4.15	Работа с ветками . . . . .	13
4.16	Создание релиза . . . . .	14
4.17	Создание журнала . . . . .	14
4.18	Добавление журнала . . . . .	14
4.19	Добавление журнала . . . . .	14
4.20	Работа с ветками . . . . .	14
4.21	Отправка данных . . . . .	15
4.22	Создание релиза . . . . .	15
4.23	Создание ветки . . . . .	15
4.24	Объединение веток . . . . .	16
4.25	Создание релиза . . . . .	16
4.26	Создание журнала . . . . .	16
4.27	Добавление журнала в индекс . . . . .	16
4.28	Добавление журнала в индекс . . . . .	16
4.29	Работа с ветками . . . . .	16
4.30	Отправка данных . . . . .	17
4.31	Отправка данных . . . . .	17
4.32	Создание релиза . . . . .	17

## **Список таблиц**

# 1 Цель работы

Получение навыков правильной работы с репозиторием git.

## 2 Задание

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

## 3 Теоретическое введение

- Gitflow Workflow опубликована и популяризована Винсентом Дриссеном.
- 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.

## 4 Выполнение лабораторной работы

### 4.1 Установка git-flow

Установка из коллекции репозитория Copr (рис. 4.1).

```
[aaermakova@fedora ~]$ sudo -i
[sudo] пароль для aaermakova:
[root@fedora ~]# # Enable the copr repository
[root@fedora ~]# dnf copr enable elegos/gitflow
https://copr.fedorainfracloud.org/api_3/rpmrepo/elegos/gitflow/fedora-41/
Включение репозитория Copr. Обратите внимание, что этот репозиторий не является частью
основного дистрибутива, и его качество может отличаться.

Проект Fedora не имеет никакой власти над содержимым
за пределами правил, изложенных в Copr FAQ по адресу
<https://docs.pagure.org/copr.copr/user_documentation.html#what-i-can-build-in-copr>,
и пакеты не имеют никаких требований к качеству или уровню безопасности.

Пожалуйста, не публикуйте сообщения об ошибках, связанных с этими пакетами, в Fedora
Bugzilla. В случае возникновения проблем обращайтесь к владельцу этого репозитория.
Is this ok [y/N]: y
[root@fedora ~]# dnf install gitflow
Обновление и загрузка репозитория:
Copr repo for gitflow owned by elegos
Репозитории загружены.
Пакет "gitflow-1.12.3-1.fc34.x86_64" уже установлен.

Нечего делать.
```

Рис. 4.1: Установка git-flow

### 4.2 Установка Node.js

Устанавливаем Node.js, на котором базируется программное обеспечение для семантического весионирования и общепринятых коммитов (рис. 4.2).



```
[root@fedora ~]# dnf install nodejs
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет "nodejs-1:22.14.0-2.fc41.x86_64" уже установлен.

Нечего делать.
[root@fedora ~]# dnf install pnpm
Обновление и загрузка репозитория:
Репозитории загружены.
Пакет "pnpm-9.13.0-1.fc41.noarch" уже установлен.

Нечего делать.
```

Рис. 4.2: Установка Node.js

## 4.3 Настройка Node.js

Для работы добавим каталог с исполняемыми файлами. Запускаю (рис. 4.3).

```
[root@fedora ~]# pnpm setup
No changes to the environment were made. Everything is already up to date.
[root@fedora ~]# source ~/.bashrc
```

Рис. 4.3: Настройка Node.js

## 4.4 Общепринятые коммиты

Программа commitizen используется для помощи в форматировании коммитов (рис. 4.4).

```
[root@fedora ~]# pnpm add -g commitizen
WARN 2 deprecated subdependencies found: glob@7.2.3, inflight@1.0.6
Already up to date
Progress: resolved 151, reused 151, downloaded 0, added 0, done
Done in 5.5s
```

Рис. 4.4: Настройка коммитов

Программа standart-changelog используется для помощи в создании логов (рис. 4.5).

```
[root@fedora ~]# pnpm add -g standart-changelog
```

Рис. 4.5: Настройка коммитов

## 4.5 Создание репозитория git

Создаю репозиторий на гитхаб (рис. 4.6).

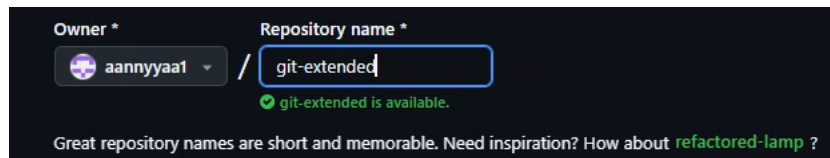


Рис. 4.6: Создание репозитория

Конфигурация общепринятых коммитов.(рис. 4.7).

```
[root@fedora ~]# pnpm init
Wrote to /root/package.json

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

Рис. 4.7: Конфигурация коммитов

Заполняю несколько параметров пакета (рис. 4.8).

```

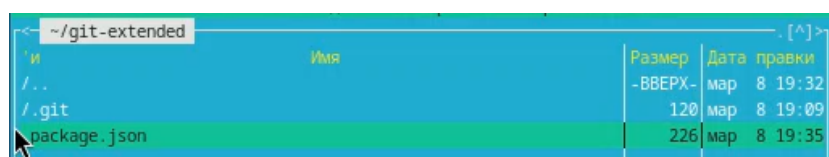
[root@fedora ~]# cd ~/git-extended
[root@fedora git-extended]# pnpm init
Wrote to /root/git-extended/package.json

{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
[root@fedora git-extended]# mc

```

Рис. 4.8: Конфигурация коммитов

Захожу в mc и редактирую файл (рис. 4.9).



Имя	Размер	Дата	правки
..	-ВВЕРХ-	мар	8 19:32
.git	120	мар	8 19:09
package.json	226	мар	8 19:35

Рис. 4.9: Открытие файла

Получившийся файл сохраняю и выхожу (рис. 4.10).

```
package.json  [-M--]  1 L: [ 1+13 14/ 14] *(373 / 373b) <EOF>
{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:aannyaa1/study_2024-2025_os-intro",
  "author": "Nastya Yeznakova <nayakaa6363@yandex.ru>",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 4.10: Редактирование файла

Добавляю новые файлы, выполняю коммит (рис. 4.11).

```
[root@fedora git-extended]# git add .
[root@fedora git-extended]# git cz
cz-cli@4.3.1, cz-conventional-changelog@3.3.0

? Select the type of change that you're committing: perf:  A code change that improves performance
? 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):
(3) lol
? Provide a longer description of the change: (press enter to skip)

? Are there any breaking changes? Yes
? A BREAKING CHANGE commit requires a body. Please enter a longer description of the commit itself:
lol lol lol lol lol
? Describe the breaking changes:
idk
? Does this change affect any open issues? No
[main (корневой коммит) 2b79c6f] perf: lol
1 file changed, 14 insertions(+)
create mode 100644 package.json
```

Рис. 4.11: Добавление файлов

Отправляю на гитхаб (рис. 4.12).

```
[root@fedora git-extended]# git push
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 1.10 КиБ | 1.10 МиБ/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/aannyaa1/git-extended.git
 * [new branch]      main -> main
```

Рис. 4.12: Отправление файлов

Инициализирую git-flow (рис. 4.13).

```
[root@fedora 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/
```

Рис. 4.13: Инициализация git-flow

Проверяю, что я в ветке develop и загружаю весь репозиторий в хранилище (рис. 4.14).

```
[root@fedora git-extended]# git branch
* develop
  main
[root@fedora 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/aannyaa1/git-extended/pull/new/develop
remote:
To https://github.com/aannyaa1/git-extended.git
 * [new branch]      develop -> develop
```

Рис. 4.14: Загрузка репозитория

Устанавливаю внешнюю ветку как вышестоящую для ветки (рис. 4.15).

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

Рис. 4.15: Работа с ветками

Создаю релиз с версией 1.0.0 (рис. 4.16).

```

[branch: develop] set up to track origin/develop :
[root@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 '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'

```

Рис. 4.16: Создание релиза

Создаю журнал изменений (рис. 4.17).

```

[root@fedora git-extended]# standart-changelog --first-release

```

Рис. 4.17: Создание журнала

Добавляю журнал изменений в индекс (рис. 4.18-4.19).

```

[root@fedora git-extended]# git add CHANGELOG.md

```

Рис. 4.18: Добавление журнала

```

[root@fedora git-extended]# git commit -am 'chore(site): add cangelog'
Текущая ветка: release/1.0.0

```

Рис. 4.19: Добавление журнала

Заливаю релизную ветку в основную ветку (рис. 4.20).

```

[root@fedora git-extended]# git flow release finish 1.0.0
Переключились на ветку «main»

```

Рис. 4.20: Работа с ветками

Отправляю данные на гитхаб (рис. 4.21).

```
[root@fedora 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 'release/1.0.0' on GitHub by visiting:
remote:   https://github.com/aannyaa1/git-extended/pull/new/release/1.0.0
remote:
To https://github.com/aannyaa1/git-extended.git
 * [new branch]      release/1.0.0 -> release/1.0.0
[root@fedora git-extended]# git push --tags
```

Рис. 4.21: Отправка данных

Создаю релиз на гитхаб (рис. 4.22).

```
[root@fedora git-extended]# gh release create v1.0.0 -F CHANGELOG.md
```

Рис. 4.22: Создание релиза

## 4.6 Работа с репозиторием git

Создаю ветку для новой функциональности (рис. 4.23).

```
[root@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 'develop'
- You are now on branch 'feature/feature_branch'

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

    git flow feature finish feature_branch
```

Рис. 4.23: Создание ветки

Объединяю ветку feature\_branch с develop (рис. 4.24).

```
[root@fedora 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 2b79c6f).

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'
```

Рис. 4.24: Объединение веток

Создаю релиз с версией 1.2.3 (рис. 4.25).

```
[root@fedora git-extended]# git flow release start 1.2.3
```

Рис. 4.25: СОздание релиза

Создаю журнал изменений (рис. 4.26).

```
[root@fedora git-extended]# standart-changelog
```

Рис. 4.26: СОздание журнала

Добавляю журнал изменений в индекс (рис. 4.27-4.28).

```
[root@fedora git-extended]# git add CHANGELOG.md
```

Рис. 4.27: Добавление журнала в индекс

```
[root@fedora git-extended]# git commit -am 'chore(site): update changelog'
[develop 59126e6] chore(site): update changelog
1 file changed, 1 insertion(+), 1 deletion(-)
```

Рис. 4.28: Добавление журнала в индекс

Заливаю релизную ветку в основную ветку (рис. 4.29).

```
[root@fedora git-extended]# git flow release finish 1.2.3
```

Рис. 4.29: Работа с ветками

Отправляю данные на гитхаб (рис. 4.30-4.31).



```
[root@fedora git-extended]# git push --all
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
При сжатии изменений используется до 2 потоков
```

Рис. 4.30: Отправка данных

```
[root@fedora git-extended]# git push --tags
```

Рис. 4.31: Отправка данных

Создаю релиз на гитхаб с комментарием из журнала изменений (рис. 4.32).

```
[root@fedora git-extended]# gh release create v1.2.3 -F CHANGELOG.md
```

Рис. 4.32: Создание релиза

## 5 Выводы

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

## **6 Список литературы**