

# **Лабораторная работа №4**

**Операционные системы**

Башиянц Александра Кареновна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
3.1	Установка git-flow . . . . .	6
3.2	Установка Node.js . . . . .	6
3.3	Настройка Node.js . . . . .	7
3.4	Общепринятые коммиты . . . . .	7
3.5	Создание репозитория git . . . . .	8
<b>4</b>	<b>Выводы</b>	<b>13</b>

# Список иллюстраций

3.1	Установка сопр	6
3.2	Установка gitflow	6
3.3	Установка Node.js	7
3.4	Установка npnm	7
3.5	setup	7
3.6	commitizen	8
3.7	standard-changelog	8
3.8	Создание репозитория	9
3.9	Работа с репозиторием	9
3.10	package.json	10
3.11	git cz	10
3.12	Инициализация git-flow	10
3.13	git branch	11
3.14	Релиз 1.0.0	11
3.15	Настройка релиза	12
3.16	git push -all	12
3.17	Создание релиза	12

# 1 Цель работы

Цель данной работы — приобрести практические навыки правильной работы с репозиториями git.

## 2 Задание

В этой лабораторной работе необходимо изучить правильную работу с репозиториями git.

Необходимо научиться:

- Работать с git flow;
- Создавать ветки;
- Работать с журналом изменений;
- Создавать релизы

## 3 Выполнение лабораторной работы

### 3.1 Установка git-flow

Установим copr (рис. 3.1) и gitflow (рис. 3.2).

```
akbashiyanc@akbashiyanc:~$ dnf copr enable elegos/gitflow
https://copr.fedorainfracloud.org/api_3/rpmre 100% | 303.0 B/s | 652.0 B | 00m02s
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
```

Рис. 3.1: Установка copr

```
akbashiyanc@akbashiyanc:~$ sudo dnf install gitflow
Updating and loading repositories:
Copr repo for gitflow owned by elegos 100% | 2.9 KiB/s | 2.4 KiB | 00m01s
Repositories loaded.
Package Arch Version Repository Size
Installing:
gitflow x86_64 1.12.3-1.fc34 copr:copr.fedorain 261.7 KiB

Transaction Summary:
Installing: 1 package
```

Рис. 3.2: Установка gitflow

### 3.2 Установка Node.js

Установим Node.js (рис. 3.3).

```
akbashiyanc@akbashiyanc:~$ sudo dnf install nodejs
Updating and loading repositories:
Repositories loaded.
Package Arch Version Repository Size
Upgrading:
libuv x86_64 1:1.50.0-1.fc41 updates 568.4 KiB
replacing libuv x86_64 1:1.49.0-1.fc41 anaconda 563.6 KiB
Installing:
nodejs x86_64 1:22.14.0-2.fc41 updates 149.5 KiB
Installing dependencies:
nodejs-libs x86_64 1:22.14.0-2.fc41 updates 75.9 MiB
Installing weak dependencies:
```

Рис. 3.3: Установка Node.js

Установим pnpm (рис. 3.4).

```
akbashiyanc@akbashiyanc:~$ sudo dnf install pnpm
[sudo] пароль для akbashiyanc:
Updating and loading repositories:
```

Рис. 3.4: Установка pnpm

### 3.3 Настройка Node.js

Для работы с Node.js добавим каталог с исполняемыми файлами, устанавливаемыми yarn, в переменную PATH (рис. 3.5).

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

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

To start using pnpm, run:
source /home/akbashiyanc/.bashrc
```

Рис. 3.5: setup

### 3.4 Общепринятые коммиты

Выполним команду для помощи в форматировании коммитов (рис. 3.6).

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

Update available! 9.13.0 → 10.5.2.
Changelog: https://github.com/pnpm/pnpm/releases/tag/v10.5.2
Run "pnpm self-update" to update.

Follow @pnpmjs for updates: https://x.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/akbashiyanc/.local/share/pnpm/global/5:
+ commitizen 4.3.1
```

Рис. 3.6: commitizen

Выполним команду для помощи в создании логов (рис. 3.7).

```
Done in 9.9s
akbashiyanc@akbashiyanc:~$ 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/akbashiyanc/.local/share/pnpm/global/5:
+ standard-changelog 6.0.0

Done in 4.8s
akbashiyanc@akbashiyanc:~$
```

Рис. 3.7: standard-changelog

## 3.5 Создание репозитория git

Создадим репозиторий на GitHub (рис. 3.8).



## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).


Required fields are marked with an asterisk (\*).

### Repository template

No template ▾

Start your repository with a template repository's contents.

Owner \*

 AlexandraBashiyants ▾

Repository name \*

git-extended

✔ git-extended is available.

Рис. 3.8: Создание репозитория

Настроим репозиторий и сделаем первый коммит (рис. 3.9).

```
akbashiyanc@akbashiyanc:~$ echo "# git-extended" >> README.md
akbashiyanc@akbashiyanc:~$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Инициализирован пустой репозиторий Git в /home/akbashiyanc/.git/
akbashiyanc@akbashiyanc:~$ git add README.md
akbashiyanc@akbashiyanc:~$ git commit -m "first commit"
[master (корневой коммит) 82006b7] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
akbashiyanc@akbashiyanc:~$ git remote add origin git@github.com:AlexandraBashiyants/
-extended.git
akbashiyanc@akbashiyanc:~$ git push -u origin main
error: src refspec main ничему не соответствует
error: не удалось отправить некоторые ссылки в «github.com:AlexandraBashiyants/git-e
nded.git»
akbashiyanc@akbashiyanc:~$ git branch -M main
akbashiyanc@akbashiyanc:~$ git push -u origin main
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 883 байта | 126.00 КиБ/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:AlexandraBashiyants/git-extended.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Рис. 3.9: Работа с репозиторием

Скорректируем файл package.json (рис. 3.10).

```
mc [akbashiyanc@akbashiyanc]:~ — /usr/bin/mc -P /var/tmp/mc-akbashiy...
/home/akbashiyanc/package.json 403/403
{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:AlexandraBashiyants/git-extended.git",
  "author": "Alexandra Bashiyants 1132246777@pfur.ru>",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 3.10: package.json

Выполним коммит (рис. 3.11).

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

? Select the type of change that you're committing: chore:   Other changes t
modify src or test files
? What is the scope of this change (e.g. component or file name): (press ente
? Write a short, imperative tense description of the change (max 93 chars):
(7) add new
? 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 6c2321b] chore: add new
1 file changed, 14 insertions(+)
```

Рис. 3.11: git cz

Инициализируем git-flow (рис. 3.12).

```
akbashiyanc@akbashiyanc:~/work/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/] v
Bugfix branches? [bugfix/] v
Release branches? [release/] v
Hotfix branches? [hotfix/] v
Support branches? [support/] v
Version tag prefix? [] v
Hooks and filters directory? [/home/akbashiyanc/work/git-extended/.git/hooks]
```

Рис. 3.12: Инициализация git-flow

Проверим, что Вы на ветке develop и загрузим весь репозиторий в хранилище(рис. 3.13).

```
akbashiyanc@akbashiyanc:~/work/git-extended$ git branch
* develop
  main
akbashiyanc@akbashiyanc:~/work/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/AlexandraBashiyants/git-extended/pull/new/develop
remote:
To github.com:AlexandraBashiyants/git-extended.git
 * [new branch]      develop -> develop
```

Рис. 3.13: git branch

Установим внешнюю ветку как вышестоящую для этой ветки и Создадим релиз с версией 1.0.0 (рис. 3.14).

```
akbashiyanc@akbashiyanc:~/work/git-extended$ git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
akbashiyanc@akbashiyanc:~/work/git-extended$ git flow release start 1.0.0
Переключились на новую ветку «v1.0.0»

Summary of actions:
- A new branch 'v1.0.0' was created, based on 'develop'
- You are now on branch 'v1.0.0'

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. 3.14: Релиз 1.0.0

Создадим журнал изменений, Добавим журнал изменений в индекс, Зальём релизную ветку в основную ветку (рис. 3.15).

```

akbashiyanc@akbashiyanc:~/work/git-extended$ standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
akbashiyanc@akbashiyanc:~/work/git-extended$ git add CHANGELOG.md
akbashiyanc@akbashiyanc:~/work/git-extended$ git commit -am 'chore(site): add changelog'
[master 2f29c64] chore(site): add changelog
1 file changed, 4 insertions(+)
create mode 100644 CHANGELOG.md
akbashiyanc@akbashiyanc:~/work/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 | 4 ++++
1 file changed, 4 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)
fatal: нет описания метки?
Fatal: Tagging failed. Please run finish again to retry.

```

Рис. 3.15: Настройка релиза

Отправим данные на github (рис. 3.16).

```

akbashiyanc@akbashiyanc:~/work/git-extended$ git push --all
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (4/4), 1.83 КиБ | 1.83 МиБ/с, готово.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:AlexandraBashiyants/git-extended.git
6c2321b..f8b09d9  main -> main
* [new branch]      v1.0.0 -> v1.0.0
akbashiyanc@akbashiyanc:~/work/git-extended$ git push --tags
Everything up-to-date

```

Рис. 3.16: git push –all

Создадим релиз на github. Для этого будем использовать утилиты работы с github (рис. 3.17).

```

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

```

Рис. 3.17: Создание релиза

## **4 Выводы**

В этой лабораторной работе мы изучили работу git.