

# **Отчет по лабораторной работе 4**

**Операционные системы**

Пестова Ева Константиновна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>4</b>	<b>Выводы</b>	<b>17</b>

# Список иллюстраций

3.1	git-flow . . . . .	7
3.2	node.js . . . . .	8
3.3	pnpm . . . . .	8
3.4	запуск . . . . .	8
3.5	настройка . . . . .	9
3.6	новый репозиторий . . . . .	9
3.7	создание коммита . . . . .	9
3.8	создание коммита . . . . .	9
3.9	создание коммита . . . . .	10
3.10	конфигурация . . . . .	10
3.11	конфигурация . . . . .	10
3.12	добавление и выполнение . . . . .	11
3.13	отправка на github . . . . .	11
3.14	инициализация . . . . .	11
3.15	настройка . . . . .	12
3.16	журнал изменений . . . . .	12
3.17	ветки . . . . .	13
3.18	отправка данных . . . . .	13
3.19	создание релиза . . . . .	14
3.20	создание ветки . . . . .	14
3.21	настройки . . . . .	14
3.22	обновление package.json . . . . .	15
3.23	изменения . . . . .	15
3.24	загрузка на github . . . . .	15
3.25	загрузка на github . . . . .	16
3.26	релиз . . . . .	16

## **Список таблиц**

# 1 Цель работы

Получение навыков правильной работы с репозиториями git

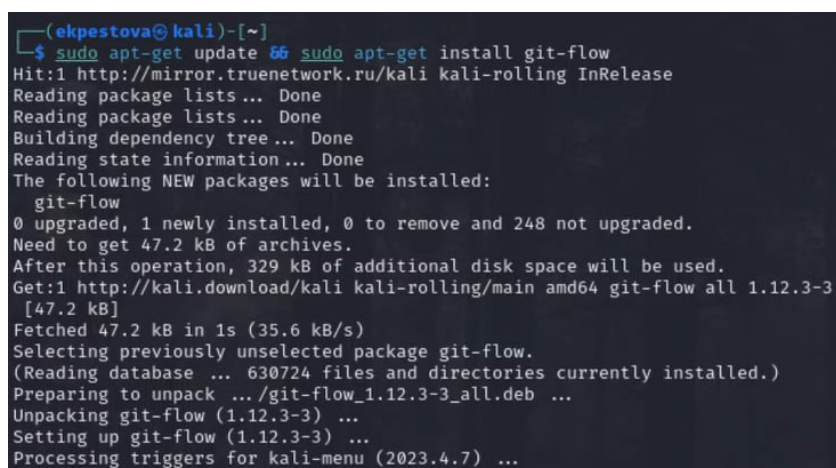
## 2 Задание

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

## 3 Выполнение лабораторной работы

### 1. Установка git-flow

Первым делом мы устанавливаем git-flow (рис. 3.1).



```
(ekpestova@kali)-[~]
$ sudo apt-get update && sudo apt-get install git-flow
Hit:1 http://mirror.truenetwork.ru/kali kali-rolling InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  git-flow
0 upgraded, 1 newly installed, 0 to remove and 248 not upgraded.
Need to get 47.2 kB of archives.
After this operation, 329 kB of additional disk space will be used.
Get:1 http://kali.download/kali kali-rolling/main amd64 git-flow all 1.12.3-3
[47.2 kB]
Fetched 47.2 kB in 1s (35.6 kB/s)
Selecting previously unselected package git-flow.
(Reading database ... 630724 files and directories currently installed.)
Preparing to unpack .../git-flow_1.12.3-3_all.deb ...
Unpacking git-flow (1.12.3-3) ...
Setting up git-flow (1.12.3-3) ...
Processing triggers for kali-menu (2023.4.7) ...
```

Рис. 3.1: git-flow

### 2. Установка node.js

Далее, я устанавливаю node.js и npm (рис. 3.2), (рис. 3.3).

```

(ekpestova@kali)-[~]
$ sudo apt-get update 66 sudo apt-get install nodejs
Hit:1 http://kali.download/kali kali-rolling InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libnode108 node-acorn node-cjs-module-lexer node-undici node-xtend
  nodejs-doc
Suggested packages:
  npm
The following NEW packages will be installed:
  libnode108 node-acorn node-cjs-module-lexer node-undici node-xtend nodejs
  nodejs-doc
0 upgraded, 7 newly installed, 0 to remove and 248 not upgraded.
Need to get 14.9 MB of archives.
After this operation, 69.1 MB of additional disk space will be used.
Do you want to continue? [Y/n]

```

Рис. 3.2: node.js

```

(ekpestova@kali)-[~]
$ wget -qO- https://get.pnpm.io/install.sh | sh -
=> Downloading pnpm binaries 8.15.4
WARN using --force I sure hope you know what you are doing
Copying pnpm CLI from /tmp/tmp.WFAGkWRBVx/pnpm to /home/ekpestova/.local/share/pnpm/pnpm
Appended new lines to /home/ekpestova/.zshrc

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

To start using pnpm, run:
source /home/ekpestova/.zshrc

(ekpestova@kali)-[~]
$ source /home/ekpestova/.zshrc

```

Рис. 3.3: pnpm

Запускаю (рис. 3.4).

```

(ekpestova@kali)-[~]
$ pnpm setup
No changes to the environment were made. Everything is already up to date.

```

Рис. 3.4: запуск

### 3. Общепринятые коммиты

Далее, я выполняю команду `source`, запускаю программы для помощи в форматировании коммитов и помощи в создании логов (рис. 3.5).



```
(ekpestova@kali)-[~]
$ source ~/.zshrc

(ekpestova@kali)-[~]
$ pnpm add -g commitizen
Already up to date
Progress: resolved 152, reused 152, downloaded 0, added 0, done
Done in 8.9s

(ekpestova@kali)-[~]
$ pnpm add -g standard-changelog
Packages: +56
Progress: resolved 208, reused 152, downloaded 56, added 56, done
/home/ekpestova/.local/share/pnpm/global/5:
+ standard-changelog 5.0.0
Done in 6.1s
```

Рис. 3.5: настройка

#### 4. Использование git

Создаю и клонирую репозиторий на github и называю его git-extended (рис. 3.6).

```
(ekpestova@kali)-[~]
$ gh repo create git-extended --public
✓ Created repository 1132236053/git-extended on GitHub
https://github.com/1132236053/git-extended

(ekpestova@kali)-[~]
$ git clone --recursive https://github.com/1132236053/git-extended
Cloning into 'git-extended' ...
warning: You appear to have cloned an empty repository.
```

Рис. 3.6: новый репозиторий

Делаю первый коммит и выкладываю на github (рис. 3.7), (рис. 3.8), (рис. 3.9).

```
(ekpestova@kali)-[~/git-extended]
$ git commit -m "First commit"
[main (root-commit) 04c8a02] First commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 First commit
```

Рис. 3.7: создание коммита

```
(ekpestova@kali)-[~/git-extended]
$ git remote add origin git@github.com:1132236053/git-extended.git
```

Рис. 3.8: создание коммита

```
(ekpestova@kali)-[~/git-extended]
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 872 bytes | 872.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:1132236053/git-extended.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Рис. 3.9: создание коммита

Следующим действием я немного меняю конфигурацию коммитов, меняя файл package.json (рис. 3.10), (рис. 3.11).

```
(ekpestova@kali)-[~/git-extended]
$ pnpm init
Wrote to /home/ekpestova/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"
}
```

Рис. 3.10: конфигурация

```
GNU nano 7.2 /home/ekpestova/git-extended/package.json
1  {
2    "name": "git-extended",
3    "version": "1.0.0",
4    "description": "Git repo for educational purposes",
5    "main": "index.js",
6    "repository": "git@github.com:1132236053/git-extended.git",
7    "author": "Eva Pestova <1132236053@pfur.ru>",
8    "license": "CC-BY-4.0",
9    "config": {
10      "commitizen": {
11        "path": "cz-conventional-changelog"
12      }
13    }
14  }
```

Рис. 3.11: конфигурация

Затем, добавляю новые файлы и выполняю коммит (рис. 3.12).

```

(ekpestova@kali)-[~/git-extended]
$ git add .

(ekpestova@kali)-[~/git-extended]
$ git cz
cz-cli@4.3.0, 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 91 chars):
(3) ///
? 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

```

Рис. 3.12: добавление и выполнение

Отправляю на github (рис. 3.13).

```

(ekpestova@kali)-[~/git-extended]
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.15 KiB | 1.15 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:1132236053/git-extended.git
04c8a02..ac24d5f  main -> main

```

Рис. 3.13: отправка на github

Инициализирую git-flow, префикс для ярлыков устанавливаю в v (рис. 3.14).

```

(ekpestova@kali)-[~/git-extended]
$ git flow init -f

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

Which branch should be used for integration of the "next release"?
- develop
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/ekpestova/git-extended/.git/hooks]

```

Рис. 3.14: инициализация

Проверяю, что я на ветке develop, загружаю весь репозиторий в хранилище, устанавливаю внешнюю ветку как вышестоящую для этой ветки (рис. 3.15).

```
(ekpestova@kali)-[~/git-extended]
$ git branch
* develop
  main

(ekpestova@kali)-[~/git-extended]
$ git push --all
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/1132236053/git-extended/pull/new/develop
remote:
To github.com:1132236053/git-extended.git
 * [new branch]      develop -> develop

(ekpestova@kali)-[~/git-extended]
$ git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
```

Рис. 3.15: настройка

Затем я создаю релиз с версией 1.0.0, создаю журнал изменений, добавляю журнал изменений в индекс (рис. 3.16).

```
(ekpestova@kali)-[~/git-extended]
$ git flow release start 1.0.0
Switched to a new branch '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'

(ekpestova@kali)-[~/git-extended]
$ standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md

(ekpestova@kali)-[~/git-extended]
$ git add CHANGELOG.md

(ekpestova@kali)-[~/git-extended]
$ git commit -am 'chore(site): add changelog'
[release/1.0.0 b66dc13] chore(site): add changelog
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md
```

Рис. 3.16: журнал изменений

Заливаю релизную ветку в основную ветку (рис. 3.17).

```
(ekpestova@kali)-[~/git-extended]
$ git flow release finish 1.0.0
Branches 'main' and 'origin/main' have diverged.
And local branch 'main' is ahead of 'origin/main'.
Already on 'main'
Your branch is ahead of 'origin/main' by 2 commits.
(use "git push" to publish your local commits)
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.
Merge made by the 'ort' strategy.
 CHANGELOG.md | 9 ++++++++
 1 file changed, 9 insertions(+)
 create mode 100644 CHANGELOG.md
Deleted branch release/1.0.0 (was b66dc13).

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
- You are now on branch 'develop'
```

Рис. 3.17: ветки

Отправляю данные на github (рис. 3.18).

```
(ekpestova@kali)-[~/git-extended]
$ git push --all
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 2.83 KiB | 2.83 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:1132236053/git-extended.git
   ac24d5f..5833591  develop → develop
   ac24d5f..fba4ae9  main → main

(ekpestova@kali)-[~/git-extended]
$ git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 173 bytes | 173.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:1132236053/git-extended.git
 * [new tag]         v1.0.0 → v1.0.0
```

Рис. 3.18: отправка данных

Создаю релиз на github. Для этого буду использовать утилиты работы с github (рис. 3.19).



```
(ekpestova@kali)~/git-extended
$ gh release create v1.0.0 -F CHANGELOG.md
https://github.com/1132236053/git-extended/releases/tag/v1.0.0
```

Рис. 3.19: создание релиза

## 5. Работа с репозиторием git

Создаю ветку для новой функциональности (рис. 3.20).

```
(ekpestova@kali)~/git-extended
$ git flow feature start feature_branch
Switched to a new branch '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. 3.20: создание ветки

После этого я объединяю ветку feature\_branch с develop и создаю релиз с версией 1.2.3 (рис. 3.21).

```
(ekpestova@kali)~/git-extended
$ git flow feature finish feature_branch
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.
Already up to date.
Deleted branch feature/feature_branch (was 5833591).

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'

(ekpestova@kali)~/git-extended
$ git flow release start 1.2.3
Switched to a new branch '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'
```

Рис. 3.21: настройки

Обновляю номер версии в файле package.json (рис. 3.22).

```
GNU nano 7.2 /home/ekpestova/git-extended/package.json *
{
  "name": "git-extended",
  "version": "1.2.3",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:1132236053/git-extended.git",
  "author": "Eva Pestova <1132236053@pfur.ru>",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 3.22: обновление package.json

Далее, создаю журнал изменений, добавляю журнал изменений в индекс и заливаю релизную ветку в основную ветку (рис. 3.23).

```
(ekpestova@kali)-[~/git-extended]
$ standard-changelog
✓ output changes to CHANGELOG.md

(ekpestova@kali)-[~/git-extended]
$ git add CHANGELOG.md

(ekpestova@kali)-[~/git-extended]
$ git commit -am 'chore(site): update changelog'
[release/1.2.3 49a340e] chore(site): update changelog
2 files changed, 5 insertions(+), 1 deletion(-)

(ekpestova@kali)-[~/git-extended]
$ git flow release finish 1.2.3
```

Рис. 3.23: изменения

Отправляю данные на github (рис. 3.24), (рис. 3.25)

```
(ekpestova@kali)-[~/git-extended]
$ git push --all
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 2.81 KiB | 2.81 MiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:1132236053/git-extended.git
```

Рис. 3.24: загрузка на github

```
(ekpestova@kali) - [~/git-extended]
$ git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 173 bytes | 173.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:1132236053/git-extended.git
 * [new tag]          v1.2.3 -> v1.2.3
```

Рис. 3.25: загрузка на github

Создаю релиз на github с комментарием из журнала изменений (рис. 3.26).

```
(ekpestova@kali) - [~/git-extended]
$ gh release create v1.2.3 -F CHANGELOG.md
https://github.com/1132236053/git-extended/releases/tag/v1.2.3
```

Рис. 3.26: релиз



## 4 Выводы

Мне удалось получить навыки правильной работы с репозиториями git