

# **Лабораторная работа №4**

**Операционные системы**

Машков Илья Евгеньевич

# 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>
3.1	Установка программного обеспечения . . . . .	7
3.1.1	Установка git-flow . . . . .	7
3.2	Установка Node.js . . . . .	7
3.3	Настройка Node.js . . . . .	8
3.4	Общепринятые коммиты . . . . .	8
3.5	Конфигурация git flow . . . . .	11
3.6	Работа с репозиторием git . . . . .	15
<b>4</b>	<b>Выводы</b>	<b>18</b>
	<b>Список литературы</b>	<b>19</b>

## Список иллюстраций

3.1	Установка git-flow. . . . .	7
3.2	Установка node.js. . . . .	7
3.3	Установка npnm. . . . .	8
3.4	Настройка Node.js. . . . .	8
3.5	Добавление программы commitizen. . . . .	8
3.6	Добавление программы standart-changelog. . . . .	9
3.7	Клонирование репозитория. . . . .	9
3.8	Создание файла. . . . .	9
3.9	Инициализация репозитория на устройстве. . . . .	9
3.10	Подготовка к отправлению первого коммита. . . . .	9
3.11	Загрузка первого коммита на сервер. . . . .	10
3.12	Создание конфигурационного файла. . . . .	10
3.13	Конфигурация файла. . . . .	11
3.14	Добавление файла и выполнение коммита. . . . .	11
3.15	Загрузка файла на сервер. . . . .	11
3.16	Инициализация git flow. . . . .	12
3.17	Проверка. . . . .	12
3.18	Загрузка всего репозитория в хранилище. . . . .	12
3.19	Установка вышестоящей метки. . . . .	12
3.20	Создание релиза с версией 1.0.0. . . . .	13
3.21	Создание журнала изменений. . . . .	13
3.22	Добавление журнала изменений в индекс. . . . .	13
3.23	Добавление релизной ветки в основную. . . . .	14
3.24	Загрузка данных на сервер. . . . .	14
3.25	Создание релиза на github. . . . .	14
3.26	Создание ветки для новой функциональности. . . . .	15
3.27	Создание релиза с новой версией. . . . .	15
3.28	Обновление данных в файле package.json. . . . .	15
3.29	Создание журнала изменений. . . . .	16
3.30	Добавление журнала изменений в индекс. . . . .	16
3.31	Добавление релизной ветки в основную. . . . .	16
3.32	Отправка данных на сервер. . . . .	17
3.33	Создание релиза. . . . .	17

## **Список таблиц**

# 1 Цель работы

Получение навыков правильной работы с репозиториями git.

## 2 Задание

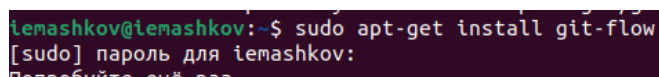
1. Выполнить работу для тестового репозитория.
2. Преобразовать рабочий репозиторий в репозиторий с git-flow и conventional commits.

## 3 Выполнение лабораторной работы

### 3.1 Установка программного обеспечения

#### 3.1.1 Установка git-flow

Для начала требуется установить **git-flow** (рис. [3.1]).

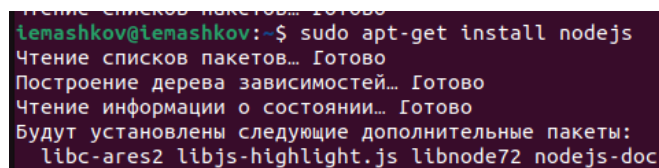


```
iemashkov@iemashkov:~$ sudo apt-get install git-flow
[sudo] пароль для iemashkov:
Подождите, пока...
```

Рис. 3.1: Установка git-flow.

### 3.2 Установка Node.js

Затем устанавливаю **node.js** (рис. [3.2]).



```
Чтение списков пакетов... Готово
Построение дерева зависимостей... Готово
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
  libc-ares2 libjs-highlight.js libnode72 nodejs-doc
```

Рис. 3.2: Установка node.js.

Также понадобится **rnpmt**, с которым возникли проблемы, поэтому я устанавливал его через **wget** (рис. [3.3]).

```

iemashkov@iemashkov:~$ wget -qO- https://get.pnpm.io/install.sh | sh -
==> Downloading pnpm binaries 8.15.4

```

Рис. 3.3: Установка pnpm.

### 3.3 Настройка Node.js

Далее провожу настройку Node.js. Запускаю **pnpm setup**, затем использую команду **source ~/.bashrc**, чтобы перелогиниться (рис. [3.4]).

```

iemashkov@iemashkov:~$ pnpm setup
No changes to the environment were made. Everything is already up to date.
iemashkov@iemashkov:~$ source ~/.bashrc

```

Рис. 3.4: Настройка Node.js.

### 3.4 Общепринятые коммиты

1. Добавляю с помощью команды **pnpm add -g commitizen**, чтобы эта программа помогала формировать коммиты (рис. [3.5]).

```

iemashkov@iemashkov:~$ source ~/.bashrc
iemashkov@iemashkov:~$ pnpm add -g commitizen
Packages: +152
+-----+
Downloading registry.npmjs.org/typescript/5.4.2: 5.82 MB/5.82 MB, done
Progress: resolved 152, reused 0, downloaded 152, added 152, done

/home/iemashkov/.local/share/pnpm/global/5:
+ commitizen 4.3.0

Done in 10.5s

```

Рис. 3.5: Добавление программы commitizen.

2. Далее, используя ту же команду устанавливаю **standart-changelog**, который будет использоваться для создания логов (рис. [3.6]).



```

iemashkov@iemashkov:~$ pnpm add -g standard-changelog
Packages: +56
+-----+
Progress: resolved 208, reused 152, downloaded 56, added 56, done

/home/iemashkov/.local/share/pnpm/global/5:
+ standard-changelog 5.0.0

Done in 7.1s

```

Рис. 3.6: Добавление программы standart-changelog.

3. Теперь мне нужно создать репозиторий для дальнейшей работы. Для этого мне нужно клонировать этот репозиторий (рис. [3.7]).

```

iemashkov@iemashkov:~/work$ git clone --recursive git@github.com:7S7eVe7N7/git-extended.git
Клонирование в «git-extended»...
warning: Похоже, что вы клонировали пустой репозиторий.

```

Рис. 3.7: Клонирование репозитория.

Затем создать файл, чтобы наш репозиторий не был пуст (рис. [3.8]).

```

iemashkov@iemashkov:~/work$ echo "# git-extended" >> README.md

```

Рис. 3.8: Создание файла.

Инициализирую репозиторий (рис. [3.9]).

```

iemashkov@iemashkov:~/work$ git init
Инициализирован пустой репозиторий Git в /home/iemashkov/work/.git/

```

Рис. 3.9: Инициализация репозитория на устройстве.

Теперь добавляю этот файл в список тех, которые я отправлю на сервер (рис. [3.10]).

```

iemashkov@iemashkov:~/work$ git add README.md
iemashkov@iemashkov:~/work$ git commit -m "first commit"
[master (корневой коммит) ec7027b] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md

```

Рис. 3.10: Подготовка к отправлению первого коммита.

И, наконец, отправляю мой первый коммит на сервер (рис. [3.11]).

```
iemashkov@iemashkov:~/work$ git remote add origin git@github.com:757eVe7N7/git-extended.git
iemashkov@iemashkov:~/work$ git push -u origin main
Перечисление объектов: 3, готово.
Подсчет объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 883 байта | 883.00 КиБ/с, готово.
Всего 3 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов
To github.com:757eVe7N7/git-extended.git
 * [new branch]      main -> main
Ветка «main» отслеживает внешнюю ветку «main» из «origin».
```

Рис. 3.11: Загрузка первого коммита на сервер.

4. Далее необходимо конфигурировать общепринятые коммиты. Для этого мне надо прописать команду **pnpm init**, чтобы создать конфигурационный файл (рис. [3.12]).

```
iemashkov@iemashkov:~/work/git-extended$ pnpm init
Wrote to /home/iemashkov/work/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. 3.12: Создание конфигурационного файла.

Конфигурирую формат коммитов в файле (рис. [3.13]).

```
{
  "name": "git-extended",
  "version": "1.0.0",
  "description": "Git repo for educational purposes",
  "main": "index.js",
  "repository": "git@github.com:7S7eVe7N7/git-extended.git",
  "author": "7S7eVe7N7 <ilia905.gt.gti@gmail.com>",
  "license": "CC-BY-4.0",
  "config": {
    "commitizen": {
      "path": "cz-conventional-changelog"
    }
  }
}
```

Рис. 3.13: Конфигурация файла.

Добавляю новый файл и выполняю коммит (рис. [3.14]).

```
lenashkov@lenashkov:~/work/git-extended$ git add .
lenashkov@lenashkov:~/work/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) package.json
? Write a short, imperative tense description of the change (max 80 chars):
(36) add configuration for a package.json
? 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 a7a88b2] feat(package.json): add configuration for a package.json
1 file changed, 15 insertions(+)
create mode 100644 package.json
```

Рис. 3.14: Добавление файла и выполнение коммита.

Затем отправляю это на сервер (рис. [3.15]).

```
lenashkov@lenashkov:~/work/git-extended$ git push
Перечисление объектов: 4, готово.
Подсчет объектов: 100% (4/4), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (3/3), 1.18 Киб | 1.18 Миб/с, готово.
Всего 3 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To github.com:7S7eVe7N7/git-extended.git
d0b04ee..a7a88b2 main -> main
lenashkov@lenashkov:~/work/git-extended$
```

Рис. 3.15: Загрузка файла на сервер.

## 3.5 Конфигурация git flow

Инициализирую **git flow**, указывая префиксом для ярлыков значение **v** (рис. [3.16]).

```

iemashkov@iemashkov:~/work/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/iemashkov/work/git-extended/.git/hooks]
iemashkov@iemashkov:~/work/git-extended$

```

Рис. 3.16: Инициализация git flow.

Проверяю, что нахожусь в ветке **develop** (рис. [3.17]).

```

iemashkov@iemashkov:~/work/git-extended$ git branch
* develop
main

```

Рис. 3.17: Проверка.

Загружаю весь репозиторий в хранилище командой **git push --all** (рис. [3.18]).

```

iemashkov@iemashkov:~/work/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/7S7eVe7N7/git-extended/pull/new/develop
remote:
To github.com:7S7eVe7N7/git-extended.git
 * [new branch]      develop -> develop

```

Рис. 3.18: Загрузка всего репозитория в хранилище.

Теперь устанавливаю внешнюю ветку вышестоящей для этой (рис. [3.19]).

```

iemashkov@iemashkov:~/work/git-extended$ git branch --set-upstream-to=origin/develop develop
Ветка «develop» отслеживает внешнюю ветку «develop» из «origin».

```

Рис. 3.19: Установка вышестоящей метки.

Создаю релиз с версией **1.0.0** (рис. [3.20]).

```

iemashkov@iemashkov:~/work/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'

iemashkov@iemashkov:~/work/git-extended$

```

Рис. 3.20: Создание релиза с версией 1.0.0.

Создаю журнал изменений (рис. [3.21]).

```

iemashkov@iemashkov:~/work/git-extended$ standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md

```

Рис. 3.21: Создание журнала изменений.

Добавляю этот журнал в индекс (рис. [3.22]).

```

iemashkov@iemashkov:~/work/git-extended$ git add CHANGELOG.md
iemashkov@iemashkov:~/work/git-extended$ git commit -am 'chore(site): add changelog'
[release/1.0.0 ca6e605] chore(site): add changelog
1 file changed, 9 insertions(+)
create mode 100644 CHANGELOG.md

```

Рис. 3.22: Добавление журнала изменений в индекс.

Заливаю релизную ветку в основную (рис. [3.23]).

```

iemashkov@iemashkov:~/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 | 9 ++++++++
  1 file changed, 9 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)
Переключились на ветку «develop»
Эта ветка соответствует «origin/develop».
Merge made by the 'ort' strategy.
  CHANGELOG.md | 9 ++++++++
  1 file changed, 9 insertions(+)
  create mode 100644 CHANGELOG.md
Ветка release/1.0.0 удалена (была сабе605).

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. 3.23: Добавление релизной ветки в основную.

Затем отправляю данные на сервер (рис. [3.24]).

```

iemashkov@iemashkov:~/work/git-extended$ git push --all
Перечисление объектов: 6, готово.
Подсчет объектов: 100% (6/6), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (5/5), готово.
Запись объектов: 100% (5/5), 2.83 Киб | 2.83 Миб/с, готово.
Всего 5 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To github.com:7S7eVe7N7/git-extended.git
   a7a88b2..fd0cd83  develop -> develop
   a7a88b2..6bc2800  main -> main
iemashkov@iemashkov:~/work/git-extended$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 169 байтов | 169.00 Киб/с, готово.
Всего 1 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To github.com:7S7eVe7N7/git-extended.git
   * [new tag]           v1.0.0 -> v1.0.0

```

Рис. 3.24: Загрузка данных на сервер.

Создаю релиз на github, используя установленные утилиты (рис. [3.25]).

```

iemashkov@iemashkov:~/work/git-extended$ gh release create v1.0.0 -F CHANGELOG.md
https://github.com/7S7eVe7N7/git-extended/releases/tag/v1.0.0

```

Рис. 3.25: Создание релиза на github.

## 3.6 Работа с репозиторием git

1. Создаю ветку для новой функциональности (рис. [3.26]).

```
iemashkov@iemashkov:~/work/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. 3.26: Создание ветки для новой функциональности.

Далее объединяю эту ветку с ветками **feature\_branch** и **develop** с помощью команды **git flow feature finish feature\_branch**

2. Создаю релиз с версией **1.2.3** (рис. [3.27]).

```
iemashkov@iemashkov:~/work/git-extended$ git flow release start 1.2.3
Переключились на новую ветку «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. 3.27: Создание релиза с новой версией.

Обновляю номер версии в файле **package.json** (рис. [3.28]).

```
3     "version": "1.2.3",
```

Рис. 3.28: Обновление данных в файле package.json.

Создаю журнал изменений (рис. [3.29]).

```
iemashkov@iemashkov:~/work/git-extended$ standard-changelog
✓ output changes to CHANGELOG.md
```

Рис. 3.29: Создание журнала изменений.

И добавляю его в индекс (рис. [3.30]).

```
iemashkov@iemashkov:~/work/git-extended$ git add CHANGELOG.md
iemashkov@iemashkov:~/work/git-extended$ git commit -am 'chore(site): update changelog'
[release/1.2.3 aca51e4] chore(site): update changelog
2 files changed, 5 insertions(+), 1 deletion(-)
```

Рис. 3.30: Добавление журнала изменений в индекс.

Заливаю релизную ветку в основную (рис. [3.31]).

```
iemashkov@iemashkov:~/work/git-extended$ git flow release finish 1.2.3
Переключились на ветку «main»
Эта ветка соответствует «origin/main».
Merge made by the 'ort' strategy.
 CHANGELOG.md | 4 ++++
 package.json | 2 +-
2 files changed, 5 insertions(+), 1 deletion(-)
Уже на «main»
Ваша ветка опережает «origin/main» на 3 коммита.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
Переключились на ветку «develop»
Эта ветка соответствует «origin/develop».
Merge made by the 'ort' strategy.
 CHANGELOG.md | 4 ++++
 package.json | 2 +-
2 files changed, 5 insertions(+), 1 deletion(-)
Ветка release/1.2.3 удалена (была aca51e4).

Summary of actions:
- Release branch 'release/1.2.3' has been merged into 'main'
- The release was tagged 'v1.2.3'
- Release tag 'v1.2.3' has been back-merged into 'develop'
- Release branch 'release/1.2.3' has been locally deleted
- You are now on branch 'develop'

iemashkov@iemashkov:~/work/git-extended$
```

Рис. 3.31: Добавление релизной ветки в основную.

Отправляю данные на github (рис. [3.32]).



```

iemashkov@iemashkov:~/work/git-extended$ git push --all
Перечисление объектов: 9, готово.
Подсчет объектов: 100% (9/9), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (6/6), готово.
Запись объектов: 100% (6/6), 2.79 КиБ | 2.79 МиБ/с, готово.
Всего 6 (изменений 2), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:757eVe7N7/git-extended.git
   fd0cd83..b5fb000  develop -> develop
   6bc2800..d6cf406  main -> main
iemashkov@iemashkov:~/work/git-extended$ git push --tags
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 168 байтов | 168.00 КиБ/с, готово.
Всего 1 (изменений 0), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
To github.com:757eVe7N7/git-extended.git
 * [new tag]         v1.2.3 -> v1.2.3

```

Рис. 3.32: Отправка данных на сервер.

Создаю релиз на github с комментарием из журнала изменений (рис. [3.33]).

```

iemashkov@iemashkov:~/work/git-extended$ gh release create v1.2.3 -F CHANGELOG.md
https://github.com/757eVe7N7/git-extended/releases/tag/v1.2.3
iemashkov@iemashkov:~/work/git-extended$

```

Рис. 3.33: Создание релиза.

## 4 Выводы

При выполнении лабораторной работы я получил навыки для правильной работы с репозиториями git.

# Список литературы

Операционные системы