

## лаборатория 4

David Michael Francis 10322249023

### цель работы

Получение навыков правильной работы с репозиториями git.

### Описание задачи

Я установил git flow из галереи хранилища copr

```
midavid@vbox:~$ sudo dnf copr enable elegos/gitflow
[sudo] password for midavid:
https://copr.fedorainfracloud.org/api_3/rp 100% | 1.5 KiB/s | 652.0 B | 00m00s
Enabling a Copr repository. Please note that this repository is not part
of the main distribution, and quality may vary.

The Fedora Project does not exercise any power over the contents of
this repository beyond the rules outlined in the Copr FAQ at
<https://docs.pagure.org/copr.copr/user_documentation.html#what-i-can-build-in-copr>
,
and packages are not held to any quality or security level.

Please do not file bug reports about these packages in Fedora
Bugzilla. In case of problems, contact the owner of this repository.
Is this ok [y/N]: y
midavid@vbox:~$ sudo dnf install gitflow
Updating and loading repositories:
Copr repo for gitflow owned by elegos      100% | 4.5 KiB/s | 2.4 KiB | 00m01s
Repositories loaded.
Package Arch Version Repository Size
Installing:
gitflow x86_64 1.12.3-1.fc34 copr:copr.fedora 261.7 KiB

Transaction Summary:
Installing: 1 package
```

### Screenshot 1

Установка Node.js Программное обеспечение для семантического версионирования и общих коммитов основано на Node.js

```
complete.  
midavid@vbox:~$ sudo dnf install nidejs  
Updating and loading repositories:  
Repositories loaded.  
Failed to resolve the transaction:  
No match for argument: nidejs  
You can try to add to command line:  
  --skip-unavailable to skip unavailable packages  
midavid@vbox:~$ sudo dnf install pnpm  
Updating and loading repositories:  
Repositories loaded.  


| Package                  | Arch   | Version          | Repository | Size      |
|--------------------------|--------|------------------|------------|-----------|
| Installing:              |        |                  |            |           |
| pnpm                     | noarch | 9.13.0-1.fc41    | updates    | 15.5 MiB  |
| Installing dependencies: |        |                  |            |           |
| nodejs                   | x86_64 | 1:22.14.0-2.fc41 | updates    | 149.5 KiB |


```

Screenshot 2

Настройка Node.js: Для работы с Node.js добавьте директорию с исполняемыми файлами, установленную в .yarnPATH

```
source /home/midavid/.bashrc  
midavid@vbox:~$ source ~/.bashrc  
midavid@vbox:~$ source /home/midavid/.bashrc
```

Screenshot 3

commitizen: Эта программа используется для форматирования коммитов. `pnpm add -g commitizen standard-changelog`: Эта программа используется для помощи в создании logs. `pnpm add -g standard-changelog`

```
midavid@vbox:~$ 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/midavid/.local/share/pnpm/global/5:  
+ commitizen 4.3.1  
  
Done in 9.1s  
midavid@vbox:~$ pnpm add -g standard-changelog  
WARN 2 deprecated subdependencies found: glob@7.2.3, inflight@1.0.6  
Packages: +39


```

Screenshot 4

### Практический сценарий использования git

1. Подключение репозитория к github Создайте репозиторий на GitHub. Например, назовем его .git-extended. Сделайте первый коммит и загрузите его на github

```
* master
midavid@vbox:~/git-extended$ git push -u origin master
Warning: Identity file /home/midavid/.ssh/id_ed22519 not accessible: No such file or directory.
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 236 bytes | 236.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Ushie47/git-extended.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

### SScreenshot 5

2. Общая конфигурация коммитов Конфигурация для Node.Мы будем использовать инициализацию PNPM

```
midavid@vbox:~/git-extended$ pnpm init
Wrote to /home/midavid/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"
}
midavid@vbox:~/git-extended$ git add .
midavid@vbox:~/git-extended$ git cz
[master (root-commit) 3e2003c] "config": {      "commitizen": {      "path
": "cz-conventional-changelog"      } }
1 file changed, 12 insertions(+)
create mode 100644 package.json
```

### SScreenshot 6

3. Конфигурация git-flow

Инициализируем git-flow

```
git flow init
```

Префикс для ярлыков установим в .v

Проверьте, что Вы на ветке :develop

```
git branch
```

Загрузите весь репозиторий в хранилище:

```
git push --all
```

Установите внешнюю ветку как вышестоящую для этой ветки:

```
git branch --set-upstream-to=origin/develop develop
```

Создадим релиз с версией 1.0.0

```
git flow release start 1.0.0
```

Создадим журнал изменений

```
standard-changelog --first-release
```

Добавим журнал изменений в индекс

```
git add CHANGELOG.md  
git commit -am 'chore(site): add changelog'
```

Зальём релизную ветку в основную ветку

```
git flow release finish 1.0.0
```

Отправим данные на github

```
git push --all  
git push --tags
```

Создадим релиз на github. Для этого будем использовать утилиты работы с github:

```
gh release create v1.0.0 -F CHANGELOG.md
```

```

midavid@vbox:~/git-extended$ git flow init

Which branch should be used for bringing forth production releases?
- master
Branch name for production releases: [master] master
Branch name for "next release" development: [develop] develop

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? [] v
Hooks and filters directory? [/home/midavid/git-extended/.git/hooks]
midavid@vbox:~/git-extended$ git branch
* develop
  master
midavid@vbox:~/git-extended$ git push --all
Warning: Identity file /home/midavid/.ssh/id_ed25519 not accessible: No such file or
directory.
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/Ushie47/git-extended/pull/new/develop

```

## Screenshot 7

```

midavid@vbox:~/git-extended$ git branch --set-upstream-to=origin/develop develop
branch 'develop' set up to track 'origin/develop'.
midavid@vbox:~/git-extended$ git flow release start 1.0.0
Switched to a new branch 'release1.0.0'

Summary of actions:
- A new branch 'release1.0.0' was created, based on 'develop'
- You are now on branch 'release1.0.0'

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'

midavid@vbox:~/git-extended$ standard-changelog --first-release
✓ created CHANGELOG.md
✓ output changes to CHANGELOG.md
midavid@vbox:~/git-extended$ git add CHANGELOG.md
midavid@vbox:~/git-extended$ git commit -am 'chore(site): add changelog'
[release1.0.0 9f73e9b] chore(site): add changelog
1 file changed, 4 insertions(+)
create mode 100644 CHANGELOG.md

```

## SCReenshot 8

### Работа с репозиторием git

1. Разработка новой функциональности Создадим ветку для новой функциональности:

```
git flow feature start feature_branch
```

Далее, продолжаем работу с git как обычно. По окончании разработки новой функциональности следующим шагом следует объединить ветку с :feature\_branchdevelop

```
git flow feature finish feature_branch
```

2. Создание релиза git-flow Создадим релиз с версией :1.2.3  

```
git flow release start 1.2.3
```

Обновите номер версии в файле . Установите её в .package.json1.2.3 Создадим журнал изменений

```
standard-changelog
```

Добавим журнал изменений в индекс

```
git add CHANGELOG.md  
git commit -am 'chore(site): update changelog'
```

Зальём релизную ветку в основную ветку

```
git flow release finish 1.2.3
```

Отправим данные на github

```
git push --all  
git push --tags
```

Создадим релиз на github с комментарием из журнала изменений:

```
gh release create v1.2.3 -F CHANGELOG.md
```

**Выход:** В этой лабораторной работе я изучал, как выполнить работу для тестового репозитория и конвертировать рабочий репозиторий в репозиторий с git-flow и обычными коммитами.