

#Лаборатория 4

#Кусоро Майова Джеймс

#1032244581

## Цель работы

Получение навыков правильной работы с репозиториями git.

##Теоретические сведения

Рабочий процесс Gitflow Рабочий процесс Gitflow Workflow. Будем описывать его с использованием пакета git-flow.

## Описание задачи

Я установил git flow из галереи хранилища copr

```
james@fedora:~$ dnf copr enable elegos/gitflow
https://copr.fedorainfracloud.org/api_3/rpmrepo/elegos/gitflow/fedora-41/ 100% | 588.0 B/s | 652.0 B | 00%
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
filesystem error: cannot set permissions: Operation not permitted [/etc/yum.repos.d/_copr:copr.fedorainfracloud.o
james@fedora:~$ dnf install gitflow
The requested operation requires superuser privileges. Please log in as a user with elevated rights, or use the "
the system state.
james@fedora:~$ dnf copr enable elegos/gitflow
https://copr.fedorainfracloud.org/api_3/rpmrepo/elegos/gitflow/fedora-41/ 100% | 1.3 KiB/s | 652.0 B | 00%
of the main distribution, and quality may vary.
```

## Screenshot 1

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

```

Nothing to do.
james@fedora:~$ sudo dnf install nodejs
Updating and loading repositories:
Repositories loaded.
Package "nodejs-1:22.14.0-2.fc41.x86_64" is already installed.

Nothing to do.
james@fedora:~$ 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 M
Transaction Summary:
Installing: 1 package
Total size of inbound packages is 3 MiB. Need to download 3 MiB.
After this operation, 16 MiB extra will be used (install 16 MiB, remove 0 B).
Is this ok [y/N]: y
[1/1] pnpm-0:9.13.0-1.fc41.noarch 100% | 6.5 MiB/s | 3.0 MiB | 00mG

```

## Screenshot 2

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

```

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>
Initialized empty Git repository in /home/james/.git/
james@fedora:~$ git add README.md
james@fedora:~$ git commit -m "first commit"
[master (root-commit) 8f93a7d] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
james@fedora:~$ git remote add origin git@github.com:James-4321/git-extended.git
bash: git: command not found...
james@fedora:~$ git remote add origin git@github.com:James-4321/git-extended.git
james@fedora:~$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 229 bytes | 229.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:James-4321/git-extended.git
* [new branch] master -> master

```

## 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

```
james@fedora:~$ pnpm add -g commitizen
WARN 2 deprecated subdependencies found: glob@7.2.3, inflight@1.0.6
Already up to date
Progress: resolved 190, reused 190, downloaded 0, added 0, done
Done in 5.2s using pnpm v10.6.1
james@fedora:~$ sudo dnf install commitizen
[sudo] password for james:
Sorry, try again.
[sudo] password for james:
Updating and loading repositories:
Repositories loaded.
Failed to resolve the transaction:
No match for argument: commitizen
You can try to add to command line:
--skip-unavailable to skip unavailable packages
```

#### 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

```
james@fedora:~$ git remote add origin git@github.com:james-4321/git-extended.git
james@fedora:~$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 229 bytes | 229.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:James-4321/git-extended.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
james@fedora:~$ pnpm init
Wrote to /home/james/package.json
```

#### Screenshot 5

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

```
james@fedora:~$ pnpm init
Wrote to /home/james/package.json

{
  "name": "james",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "packageManager": "pnpm@10.6.1"
}
```

#### Screenshot 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
```

```

james@fedora:~/git-extended$ git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main] master
Fatal: Local branch 'master' does not exist.
james@fedora:~/git-extended$ git branch
* main
james@fedora:~/git-extended$ git branch -m main master
james@fedora:~/git-extended$ git flow init

Which branch should be used for bringing forth production releases?
- master
Branch name for production releases: [master]
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? []
Hooks and filters directory? [/home/james/git-extended/.git/hooks]
james@fedora:~/git-extended$ git branch

```

## Screenshot 7

```

Deleted branch release/1.0.0 (was b085143).

Summary of actions:
- Release branch 'release/1.0.0' has been merged into 'master'
- The release was tagged '1.0.0'
- Release tag '1.0.0' has been back-merged into 'develop'
- Release branch 'release/1.0.0' has been locally deleted; it has been remotely deleted from 'origin'
- You are now on branch 'develop'

james@fedora:~/git-extended$ git flow release start 1.2.3
Branches 'develop' and 'origin/develop' have diverged.
And local branch 'develop' is ahead of 'origin/develop'.
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'

james@fedora:~/git-extended$ standard-changelog
✓ output changes to CHANGELOG.md
james@fedora:~/git-extended$ git add CHANGELOG.md
james@fedora:~/git-extended$ git commit -am 'chore(site): update changelog'
On branch release/1.2.3

```

## 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.