

# **Отчет по выполнению лабораторной работы №4**

**Операционные системы**

Беспутин Г. А.

# Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выводы	15
	Список литературы	16

# Список иллюстраций

3.1	.....	8
3.2	.....	8
3.3	.....	9
3.4	.....	9
3.5	.....	9
3.6	.....	9
3.7	.....	10
3.8	.....	10
3.9	.....	10
3.10	.....	10
3.11	.....	10
3.12	.....	10
3.13	.....	11
3.14	.....	11
3.15	.....	11
3.16	.....	11
3.17	.....	11
3.18	.....	11
3.19	.....	12
3.20	.....	12
3.21	.....	12
3.22	.....	12
3.23	.....	13
3.24	.....	13
3.25	.....	13
3.26	.....	13
3.27	.....	13
3.28	.....	14

## **Список таблиц**

# 1 Цель работы

Получение навыков правильной работы с репозиториями git.

## 2 Задание

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

### 3 Теоретическое введение

Gitflow Workflow опубликована и популяризована Винсентом Дриссеном. Gitflow Workflow предполагает выстраивание строгой модели ветвления с учётом выпуска проекта. Данная модель отлично подходит для организации рабочего процесса на основе релизов. Работа по модели Gitflow включает создание отдельной ветки для исправлений ошибок в рабочей среде. Последовательность действий при работе по модели Gitflow: Из ветки master создаётся ветка develop. Из ветки develop создаётся ветка release. Из ветки develop создаются ветки feature. Когда работа над веткой feature завершена, она сливается с веткой develop. Когда работа над веткой релиза release завершена, она сливается в ветки develop и master. Если в master обнаружена проблема, из master создаётся ветка hotfix. Когда работа над веткой исправления hotfix завершена, она сливается в ветки develop и master. # Выполнение лабораторной работы

Установка из коллекции репозитория Copr(рис. [3.1]).

```

\[\root@glebbesputin ~\]#dnf copr enable elegos/gitflow
Включение репозитория Copr. Обратите внимание, что этот репозиторий
не является частью основного дистрибутива, и качество может отличаться.

Проект Fedora не имеет какого-либо влияния на содержимое этого
репозитория за рамками правил, описанных в Вопросах и Ответах Copr в
<https://docs.pagure.org/copr.copr/user\_documentation.html#what-i-can-build-in-copr>,
а качество и безопасность пакетов не поддерживаются на каком-либо уровне.

Не отправляйте сообщения об ошибках этих пакетов в Fedora
Bugzilla. В случае возникновения проблем обращайтесь к владельцу этого репозитория.

Do you really want to enable copr.fedorainfracloud.org/elegos/gitflow? [y/N]: y
Репозиторий успешно подключен.
\[\root@glebbesputin ~\]# dnf install gitflow
Copr repo for gitflow owned by elegos          2.2 kB/s | 1.5 kB      00:00
Fedora 38 - x86_64 - [ === ] --- B/s | 0 B      --:-- ETA

```

Рис. 3.1:

На Node.js базируется программное обеспечение для семантического версионирования и общепринятых коммитов(рис. [3.2]).

```

\[\root@glebbesputin ~\]# dnf install nodejs

```

Рис. 3.2:

Для работы с Node.js добавим каталог с исполняемыми файлами, устанавливаемыми yarn, в переменную PATH(рис. [3.3]).



```

[root@glebbesputin ~]# pnpm setup
Appended new lines to /root/.bashrc

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

To start using pnpm, run:
source /root/.bashrc
[root@glebbesputin ~]# source ~/.bashrc
[root@glebbesputin ~]# pnpm add -g commitizen

```

Рис. 3.3:

Данная программа используется для помощи в форматировании коммитов(рис. [3.4]).

```

[root@glebbesputin ~]# pnpm add -g standard-changelog

```

Рис. 3.4:

Данная программа используется для помощи в создании логов(рис. [3.5]).

```

[root@glebbesputin ~]# pnpm add -g standard-changelog

```

Рис. 3.5:

Делаем первый коммит и выкладываем на github(рис. [3.6]).

```

[root@glebbesputin ~]# git commit -m "first commit"

```

Рис. 3.6:

Конфигурация для пакетов Node.js(рис. [3.7]).

```
one in 3.9s  
root@glebbesputin ~]# git commit -m "first commit"
```

Рис. 3.7:

Добавим новые файлы(рис. [3.8]).

```
[glebbesputin@glebbesputin git-extended]$ git add .
```

Рис. 3.8:

Выполним коммит(рис. [3.9]).

```
[glebbesputin@glebbesputin git-extended]$ git cz
```

Рис. 3.9:

Отправим на github(рис. [3.10]).

```
[glebbesputin@glebbesputin git-extended]$ git push
```

Рис. 3.10:

Инициализируем git-flow(рис. [3.11]).

```
[glebbesputin@glebbesputin git-extended]$ git flow init
```

Рис. 3.11:

Проверьте, что Вы на ветке develop(рис. [3.12]).

```
Branch name for production releases: [master] git branch
```

Рис. 3.12:

Проверьте, что Вы на ветке develop(рис. [3.13]).

```
branch name for "next release" development: [develop] git push --all
```

Рис. 3.13:

Проверьте, что Вы на ветке develop(рис. [3.14]).

```
[glebbesputin@glebbesputin git-extended]$ git branch --set-upstream-to=origin/develop develop
```

Рис. 3.14:

Создадим релиз с версией 1.0.0(рис. [3.15]).

```
[glebbesputin@glebbesputin git-extended]$ git flow release start 1.0.0
```

Рис. 3.15:

Создадим журнал изменений(рис. [3.16]).

```
[glebbesputin@glebbesputin git-extended]$ standard-changelog --first-release
```

Рис. 3.16:

Добавим журнал изменений в индекс(рис. [3.17]).

```
[glebbesputin@glebbesputin git-extended]$ git add CHANGELOG.md
```

Рис. 3.17:

Зальём релизную ветку в основную ветку(рис. [3.18]).

```
[glebbesputin@glebbesputin git-extended]$ git flow release finish 1.0.0
```

Рис. 3.18:

Отправим данные на github(рис. [3.19]).

```
[glebbesputin@glebbesputin git-extended]$ git push --all
Перечисление объектов: 6, готово.
Подсчет объектов: 100% (6/6), готово.
При сжатии изменений используется до 10 потоков
Сжатие объектов: 100% (4/4), готово.
Запись объектов: 100% (6/6), 645 байтов | 645.00 КиБ/с, готово.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:GLEBB2005/git-extended.git
 * [new branch]      main -> main
[glebbesputin@glebbesputin git-extended]$ git push --tags
```

Рис. 3.19:

Создадим релиз на github. Для этого будем использовать утилиты работы с github(рис. [3.20]).

```
[glebbesputin@glebbesputin git-extended]$ gh release create v1.0.0 -F CHANGELOG.
md
https://github.com/GLEBB2005/git-extended/releases/tag/v1.0.0
[glebbesputin@glebbesputin git-extended]$ git flow feature start feature_branch
Fatal: Not a gitflow-enabled repo yet. Please run 'git flow init' first.
```

Рис. 3.20:

Создадим ветку для новой функциональности(рис. [3.21]).

```
[glebbesputin@glebbesputin git-extended]$ git flow feature finish feature_branch
Fatal: Not a gitflow-enabled repo yet. Please run 'git flow init' first.
[glebbesputin@glebbesputin git-extended]$
```

Рис. 3.21:

По окончании разработки новой функциональности следующим шагом следует объединить ветку feature\_branch с develop(рис. [3.22]).

```
[glebbesputin@glebbesputin git-extended]$ git flow feature finish feature_branch
Fatal: Not a gitflow-enabled repo yet. Please run 'git flow init' first.
[glebbesputin@glebbesputin git-extended]$
```

Рис. 3.22:

Создадим релиз с версией(рис. [3.23]).

```
[glebbesputin@glebbesputin git-extended]$ git flow release start 1.2.3
fatal: Not a gitflow-enabled repo yet. Please run 'git flow init' first.
```

Рис. 3.23:

Создадим журнал изменений(рис. [3.24]).

```
[glebbesputin@glebbesputin git-extended]$ standard-changelog
```

Рис. 3.24:

Добавим журнал изменений в индекс(рис. [3.25]).

```
[glebbesputin@glebbesputin git-extended]$ git add CHANGELOG.md
```

Рис. 3.25:

Зальём релизную ветку в основную ветку(рис. [3.26]).

```
[glebbesputin@glebbesputin git-extended]$ git flow release finish 1.2.3
```

Рис. 3.26:

Отправим данные на github(рис. [3.27]).

```
[glebbesputin@glebbesputin git-extended]$ git push --all
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
При сжатии изменений используется до 10 потоков
Сжатие объектов: 100% (2/2), готово.
Запись объектов: 100% (3/3), 318 байтов | 318.00 КиБ/с, готово.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:GLEBB2005/git-extended.git
 59d2163..f6c1bd7 main -> main
[glebbesputin@glebbesputin git-extended]$ git push --tags
```

Рис. 3.27:

Создадим релиз на github с комментарием из журнала изменений(рис. [3.28]).

```
[glebbesputin@glebbesputin git-extended]$ gh release create v1.2.3 -F CHANGELOG.  
md
```

Рис. 3.28:

## 4 Выводы

Я получил навыки правильной работы с репозиториями git. Выполнил работу для тестового репозитория. Преобразовал рабочий репозиторий в репозиторий с git-flow и conventional commits.

## **Список литературы**