

Презентация по лабораторной работе №4

Архитектура компьютеров и операционные
системы

Мохамед Муса

Цель работы

Целью данной работы является приобретение практических навыков работы с расширенными возможностями git, включая установку необходимого программного обеспечения, настройку инструментов для работы с коммитами и использование git-flow.

Задание

1. Установка необходимого программного обеспечения
2. Настройка инструментов для работы с коммитами
3. Практический сценарий использования git
4. Настройка и использование git-flow

Установка программного обеспечения

Установка git-flow Linux (Fedora)

Установка из коллекции репозитория Copr:

```
# Enable the copr repository
sudo dnf copr enable elegos/gitflow
# Install gitflow
sudo dnf install gitflow
```

Установка Node.js

На Node.js базируется программное обеспечение для семантического versionирования и общепринятых КОММИТОВ.

```
sudo dnf install nodejs
sudo dnf install npm
```

Настройка Node.js

Для работы с Node.js добавим каталог с исполняемыми файлами в переменную PATH.

```
npm setup
```

Инструменты для работы с коммитами

commitizen

Данная программа используется для помощи в форматировании коммитов.

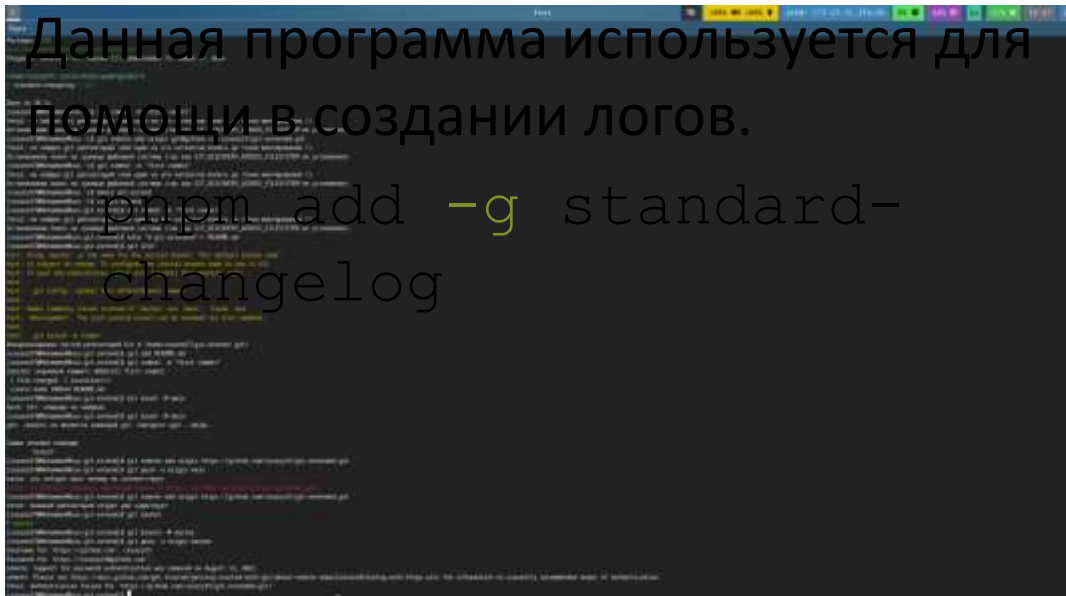
```
pnpm add -g commitizen
```

Устанавливается скрипт `git-cz` для форматирования коммитов.

standard-changelog

Данная программа используется для помощи в создании логов.

```
pnpm add -g standard-changelog
```



Коммиты в интерфейсе git-cz

Практический сценарий использования git

Создание репозитория git

1. Создайте репозиторий на GitHub (например, git-extended).
2. Первоначальная настройка:

```
git init
git commit -m "first commit"
git remote add origin
git@github.com:<username>/git-extended.git
git push -u origin main
```

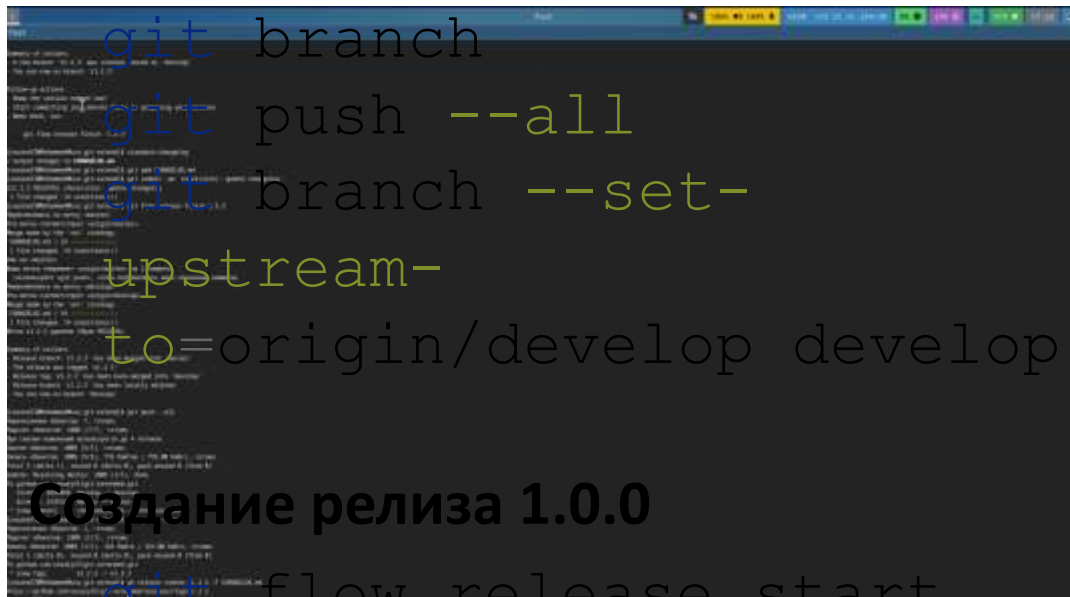
Проверка ветки и настройка

Настройка git-flow

Инициализация git-flow

```
git flow init
```

*# Установите префикс для ярлыков
(например, v)*



Создание релиза 1.0.0

```
git flow release start  
1.0.0
```

```
git commit -m "Initial  
release commit"
```

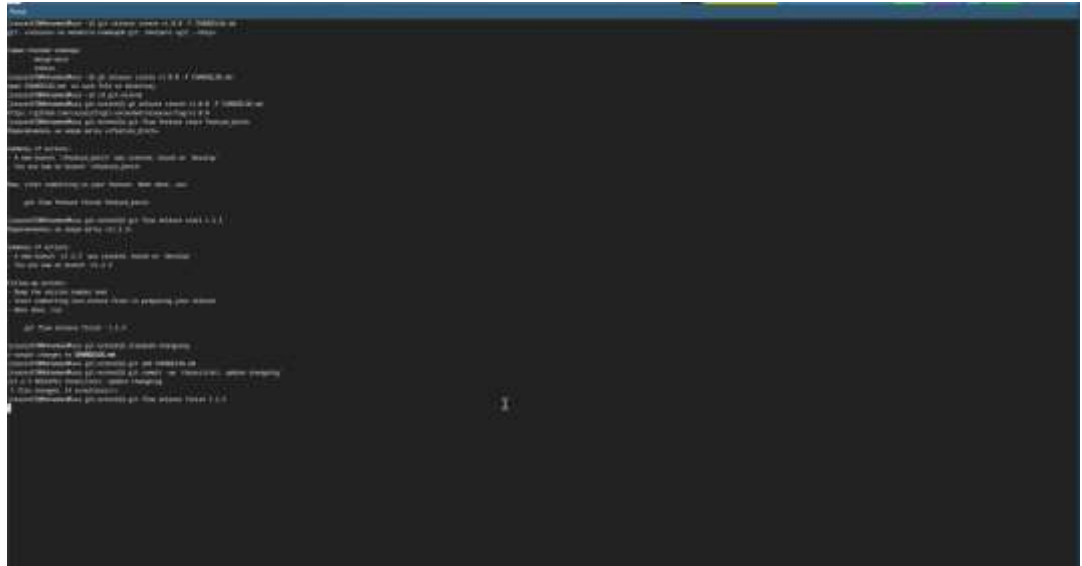
```
git flow release finish
```


Подписывание тега

Генерация changelog

```
standard-changelog -v 1.0.0 -p
```

angular -s



Создание changelog

Сгенерированный changelog

Выводы

При выполнении данной лабораторной работы были приобретены практические навыки работы с расширенными возможностями git, включая:

- Установку необходимого программного обеспечения
- Настройку инструментов для работы с коммитами
- Практическое применение git в проектах
- Настройку и использование git-flow для управления ветвлением
- Генерацию логов изменений