

Презентация к лабораторной работе №1

Операционные системы

Беспутин Г. А.

Российский университет дружбы народов, Москва, Россия

Получение навыков правильной работы с репозиториями git.

Задачи:

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

Gitflow Workflow опубликована и популяризована Винсентом Дриссенем. 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.

Установка из коллекции репозитория Copr
(<https://copr.fedorainfracloud.org/coprs/elegos/gitflow/>):

```
dnf copr enable elegos/gitflow
```

```
dnf install gitflow
```

На Node.js базируется программное обеспечение для семантического версионирования и общепринятых коммитов.

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

Делаем первый коммит и выкладываем на github.

Конфигурация для пакетов Node.js.

Добавим новые файлы.

Выполним коммит.

Отправим на github.

Инициализируем git-flow. Проверьте, что Вы на ветке develop.

Проверьте, что Вы на ветке develop.

Проверьте, что Вы на ветке develop.

Создадим релиз с версией 1.0.0.

Создадим журнал изменений.

Добавим журнал изменений в индекс.

Зальём релизную ветку в основную ветку.

Отправим данные на github.

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

Создадим ветку для новой функциональности.

По окончании разработки новой функциональности следующим шагом следует объединить ветку `feature_branch` с `develop`.

Создадим релиз с версией.

Создадим журнал изменений.

Добавим журнал изменений в индекс.

Зальём релизную ветку в основную ветку..

Отправим данные на github.

Создадим релиз на github с комментарием из журнала изменений.

Полученил навыки правильной работы с репозиториями git.

Выполнил работу для тестового репозитория.

Преобразовал рабочий репозиторий в репозиторий с git-flow и conventional commits.

⋮