Отчет по лабораторной работе №4

Операционные системы

Карпова Анастасия Александровна

Содержание

# 1 Цель работы

Получение навыков правильной работы с репозиториями git.

# 2 Задание

1. Установка git-flow
2. Установка Node.js
3. Настройка Node.js
4. Общепринятые коммиты

# 3 Выполнение лабораторной работы

## 3.1 Установка git-flow

Установка из коллекции репозиторие Сopr с помощью dnf copr enable elegos/gitflow dnf install gitflow

## 3.2 Установка Node.js

На Node.js базируется программное обеспечение для семантического версионирования и общепринятых коммитов. Устанваливаю Node.js b pnpm при помощи

dnf install nodejs apt-get install pnpm

## 3.3 Настройка Node.js

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

Запускаю: pnpm setup Далее выполняю source ~/.bashrc

## 3.4 Общепринятые коммиты

1. commitizen

* Данная программа используется для помощи в форматировании коммитов. pnpm add -g commitizen При этом устанавливается скрипт git-cz, который мы и будем использовать для коммитов

1. standard-changelog

* Данная программа используется для помощи в создании логов. pnpm add -g standard-changelog

1. Практический сценарий использования git

* 3.1. Создание репозитория git 3.1.1. Подключение репозитория к github Создаю репозиторий на GitHub. Для примера назовём его git-extended Делаем первый коммит и выкладываем на github:
* git commit -m "first commit"  
   git remote add origin git@github.com:<username>/git-extended.git  
   git push -u origin master
* 3.2. Практический сценарий использования git 3.2.1. Создание репозитория git Подключение репозитория к github Создайте репозиторий на GitHub. Для примера назовём его git-extended. Делаем первый коммит и выкладываем на github:
* git commit -m "first commit"  
   git remote add origin git@github.com:<username>/git-extended.git  
   git push -u origin master  
    
   Сконфигурим формат коммитов. Для этого добавим в файл package.json команду для формирования коммитов:

“config”: { “commitizen”: { “path”: “cz-conventional-changelog” } }

Таким образом, файл package.json приобретает вид:

{ “name”: “git-extended”, “version”: “1.0.0”, “description”: “Git repo for educational purposes”, “main”: “index.js”, “repository”: “git@github.com:username/git-extended.git”, “author”: “Name Surname [username@gmail.com](mailto:username@gmail.com)”, “license”: “CC-BY-4.0”, “config”: { “commitizen”: { “path”: “cz-conventional-changelog” } }  
}

Добавим новые файлы: git add . Выполним коммит: git cz Отправим на github: git push

1. 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

1. Работа с репозиторием git

* Разработка новой функциональности
* Создадим ветку для новой функциональности:  
    
  git flow feature start feature\_branch  
    
  Далее, продолжаем работу c git как обычно.  
    
  По окончании разработки новой функциональyности следующим шагом следует объединить ветку feature\_branch c develop:  
    
  git flow feature finish feature\_branch
* Создание релиза git-flow
* Создадим релиз с версией 1.2.3:
* git flow release start 1.2.3
* Обновите номер версии в файле package.json. Установите её в 1.2.3.
* 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

# 4 Выводы

В ходе работы получила навыки работы с git

# Список литературы

1. Архитектура ЭВМ