https://github.com/al-sapsan email: al.sapsan@mail.ru

ПРАКТИЧЕСКАЯ РАБОТА № 4

модуля «Работа с системой контроля версий Git»

курса «Linux для робототехников».

Основание:
изучение модуля «Работа с
системой контроля версий Git»
курса «Linux для робототехников».

Исполнитель:

Соколов Олег Вадимович

СОЧИ 2024

СОДЕРЖАНИЕ:

1. «Что нужно сделать» [1] [ссылка на содержание]:

1. Создайте Git-репозиторий для проекта, например RobotMotorsControl, на платформе GitHub (или GitLab). Клонируйте репозиторий на свой компьютер.
2. Создайте файлы в каталоге репозитория, например README.md и motors_control.py. Реализуйте скрипт для управления двигателями робота (например, на Python).
3. Сохраните изменения в репозитории с помощью команд git add, git commit и git push (initial commit).
4. Создайте новую ветку для внесения улучшений (git checkout -b feature_branch_name).
5. Внесите улучшения в код в новой ветке (например, функции плавного изменения скорости или обработки пользовательского ввода).
6. Сохраните внесённые улучшения в репозитории.
7. Создайте пул-реквест для слияния изменений с основной веткой (на GitHub/GitLab).
8. Опционально: дождитесь ревью кода от куратора и внесите необходимые правки, если потребуется.
9. Закройте пул-реквест и завершите слияние изменений с основной веткой.
10. Оформите краткий отчёт в файле README.md в Git-репозитории. Отчёт должен содержать следующие разделы:
    - 10.1. Общее описание проекта RobotMotorsControl: кратко изложите цель и назначение проекта.
    - 10.2. Описание работы с Git:
        - 10.2.1. Создание и клонирование репозитория: опишите, как был создан и склонирован репозиторий.
        - 10.2.2. Процесс сохранения изменений: опишите последовательность команд для добавления, фиксации и отправки изменений.
        - 10.2.3. Взаимодействие с ветками: опишите, как создавались, изменялись и сливались ветки в вашем репозитории.
        - 10.2.4. Пул-реквесты: опишите, какие пул-реквесты были созданы, каким был процесс ревью и какие изменения были предложены и применены (если таковые были).
        - 10.2.5. Основные изменения и улучшения в коде: кратко опишите ключевые добавленные или модифицированные функции и их назначение.

[1] В соответствии с заданием к Практической работе модуля 4 «Работа с системой контроля версий Git» курса «Linux для робототехников».

10.3. Заключение: изложите впечатления от работы с Git, а также возможные планы или предложения по дальнейшему улучшению проекта и взаимодействию с Git.

2. Фабула проекта[2] [ссылка на содержание]

В соответствии с техническим заданием необходимо создать программное обеспечение на языке Python для управления двигателями мультикоптера. Для реализации данного проекта в репозитории `linux` была создана ветвь с названием `module4`. При дальнейшем изучении технических характеристик проектного оборудования установлено, что мультикоптер представляет собой квадрокоптер, управляемый полетным контроллером, реализованным на базе платы Arduino Nano. Так как данная плата не поддерживает напрямую язык программирования Python, было принято решение дополнить проект скетчем на языке C++ для платы Arduino Nano, а также создать программное обеспечение на языке Python для управления двигателями квадрокоптера через последовательный порт с компьютера. Таким образом, было утверждено следующее проектное решение:

1. C++:
- инициализирует последовательный порт платы для связи с компьютером;
- считывает данные и изменяет мощность моторов (ESC) в зависимости от значения, полученного по последовательному порту.
2. Python:
- устанавливает связь с Arduino;
- вводит значение мощности и отправляет его на Arduino.

Для реализации проектного решения в ветви `module4` репозитория `linux` были созданы следующие ветви:

- ветвь с названием `module4-python` для реализации программного обеспечения на языке Python;
- ветвь с названием `module4-cpp` для реализации скетча на языке C++.

После реализации проектного решения в репозитории `linux` был создан пул-реквест для слияния изменений с основной веткой.

3. Описание работы с Git [ссылка на содержание]

3.1. Создание и клонирование репозитория[3].

---

[2] В данном случае – выдуманные обстоятельства проекта, необходимые для понимания проведенных действий в репозитории Git в соответствии с заданием модуля.

[3] Здесь стоит отметить тот факт, что репозиторий для выполнения и архивирования заданий курса уже существует, по этой причине была создана лишь ветка для модуля 4, а не полномасштабный репозиторий. Но так как назначением данной практической работы является изучение и иллюстрация навыков слушателя курса по работе с Git, то подобное отступление от параметров задания является логичным и не влияющим на процесс обучения и контроля за ним. Ведь наличие уже существующих и работоспособных репозиториев априори является основанием для положительной оценки знаний по работе с данной системой контроля версий.

В соответствии с заданием в репозитории `linux` с помощью команды:

```
git checkout -b module4
```

была создана ветка для выполнения практического задания данного модуля. После чего, с помощью текстового редактора VIM в каталоге репозитория был создан файл формата `.md`, поименованный как `README.md`, содержащий текстовое описание задания модуля, фабулу проекта и ссылку на настоящую практическую работу.

3.2. Процесс сохранения изменений [ссылка на содержание]:

Далее, путем последовательного использования команд:

```
git add README.md
git commit -m "…Сообщение …"
git push origin module4
```

файл `README.md` был помещен на главную страницу ветки `module4` репозитория `linux`.

Здесь следует отметить, что все вышеописанные действия могли быть реализованы и с помощью графического веб-интерфейса удаленного репозитория с дальнейшим сохранением (клонированием) изменений на локальный компьютер с помощью команды:

```
git clone https://github.com/git-skillbox/linux/tree/module4
```

3.3. Взаимодействие с ветками [ссылка на содержание]:

Далее, во исполнение фабулы задания была создана ветвь `module4-cpp`, якобы являющаяся ответвлением проекта `module4`[4] для реализации скетча на C++:

```
git checkout -b module4-cpp
```

После чего с помощью редактора VIM был создан файл `m4_engine_control.cpp`, содержащий соответствующий программный код.

Далее, вышеуказанный файл был помещен в удаленный репозиторий путем последовательного использования команд:

```
git add m4_engine_control.cpp
git commit -m "…Сообщение …"
git push origin module4-cpp
```

Подобным же образом была создана и другая ветка – `module4-python`, файл – `m4_engine_control.py`, содержащий программный код на языке Python, и проведена синхронизация данных с удаленным репозиторием.

---

[4] Фактически и в соответствии со своей структурой данных, Git не создает подветви ввиду сложности дальнейшего контроля изменений. Но в данном случае мы будем исходить из допущения, что мы создали не параллельные ветви репозитория (как это есть на самом деле), а ветви в отдельном проекте, якобы существующем у нас.

3.4. Пул-реквесты [ссылка на содержание]:

Создание пулл-реквеста обычно выполняется через веб-интерфейс платформы. Также как и слияние изменений обычно происходит на стороне платформы после того, как пулл-реквест одобрен [см. иллюстрацию 1]. Однако, в целях обучения было принято решение выполнить слияние ветки `module4-cpp` локально. Таким образом, после одобрения пулл-реквеста последовательность команд будет следующей:

`git status` – смотрим текущее состояние репозитория;

`git branch -a` – смотрим список всех веток репозитория: как локальные, так и удаленные. Активная (текущая) ветка помечена звездочкой `*`;

`git checkout module4` – переключаемся на основную ветку репозитория;

`git pull origin module4` – получаем новые изменения из удаленного репозитория и сливаем их с локальной веткой;

`git merge module4-cpp` – сливаем принятые изменения в ветке `module4-cpp` с ее предком – веткой `module4`;

`git push origin module4` – сливаем локальные изменения в удаленный репозиторий.

Приложение: иллюстративный материал на 1 л., в 1 экз.

Приложение 1
к практической работе № 4.

ИЛЛЮСТРАТИВНЫЙ МАТЕРИАЛ

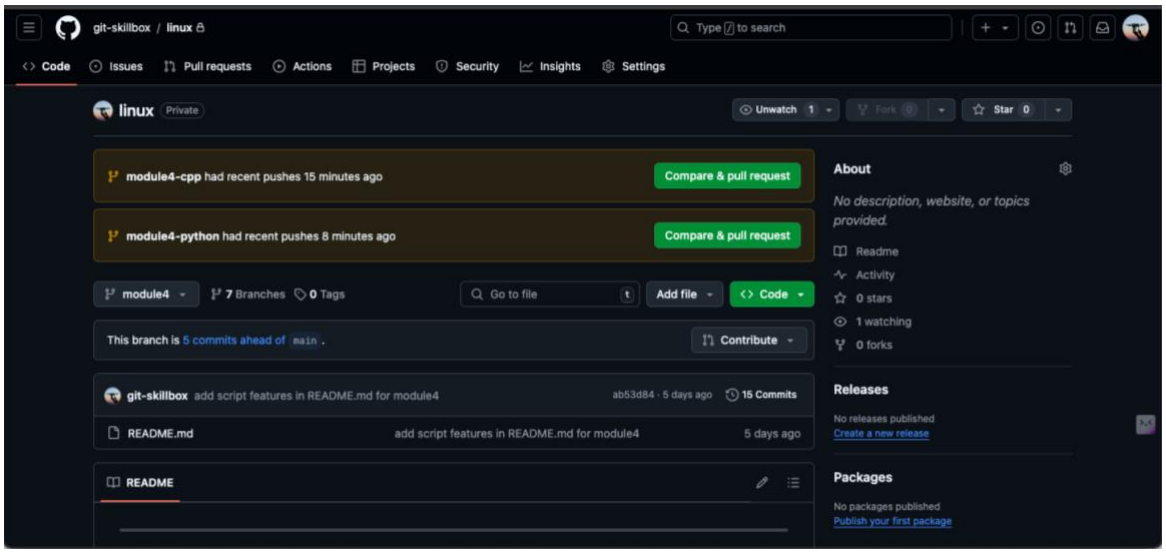

Иллюстрация 1. Окно браузера со страницей Git, иллюстрирующее пулл-реквест в веб-интерфейсе платформы [обратная ссылка].