

## Задание 11

### Условие

Основная цель переработать Задание 7 так, чтобы ввод данных контролировался внешним кодом. Интерфейс описан в файле `async.h`. Инициатором обмена будет выступать внешний код, вместо привычной входной точки `main()` будет три - `connect()`, `receive()` и `disconnect()`.

Порядок вызова следующий:

- Вызывается `connect()` с размером блока команд. Сохраняется значение возврата. Значение никак не интерпретируется и служит только для передачи контекста;
- Вызывается `receive()` с передачей указателя на начало буфера, размера буфера и контекста. Вызов повторяемый;
- Вызывается `disconnect()` с передачей контекста. Вызов разрушает контекст.

Необходимо реализовать эти функции так, чтобы сохранить прежнюю функциональность проекта и добавить новую возможность.

Реализация должна допускать множественные вызовы `connect()`. Вызовы `receive()` с разными контекстами не должны мешать друг другу.

В зависимости от внутренней реализации может возникнуть требование иметь одинаковый размер блока для всех команд. Нет цели вносить столь глубокие изменения. Если потенциальная возможность иметь разные размеры очереди влечет глубокую переработку функция `connect()` должна вернуть `nullptr`.

Вызовы могут осуществляться из разных потоков, однако вызовы с одинаковым контекстом выполняются из одного и того же потока.

Опционально реализовать возможность вызывать все функции из любых потоков.

Опционально взять за основу вместо однопоточной версии из Задания 7 многопоточную реализацию из Задания 10.

### Требования к реализации

Результатом работы должна стать библиотека устанавливаемая по стандартному пути. Библиотека должна называться `libasync.so` и находиться в пакете `async`.

Результат работы должен быть опубликован на `bintray`.

## Проверка

Задание считается выполненным успешно, если после установки пакета, линковки с тестовым кодом (пример в `main.cpp`) и запуска с тестовыми данными вывод соответствует описанию Задания 7. Данные подаются порциями в разных контекстах в большом объеме без пауз.

Будет отмечено отсутствие ограничений на вызов из разных потоков и взятая за основу многопоточная версия задания.