

## Лабораторная работа №4 «Использование фреймворка ACE и паттернов программирования Reactor, Acceptor, Connector»

### Введение

При разработке современного программного обеспечения часто возникает требование кроссплатформенности. Кроме того, прямое использование низкоуровневых библиотек требует большого внимания от программиста к различным деталям (например, все необходимые вызовы и константы для инициализации сокета в Linux). Для решения этих проблем используются уже готовые wrappers, такие как фреймворк The ADAPTIVE Communication Environment (ACE).

Также хорошей практикой является использование различных паттернов программирования.

### Исходные данные

Следует использовать исходные данные из лабораторной работы №1 (этап №3).

### Задание

Требуется кроссплатформенно реализовать распределенную систему, описанную в этапе №1 лабораторной работы №1. Для достижения этой цели необходимо использовать wrappers, представленные в фреймворке ACE.

Взаимодействие между процессами должно осуществляться посредством сокетов (см. классы `SOCK_Stream`, `SOCK_Acceptor`, `SOCK_Connector`), а не каналов. Установка соединения и логика самой распределенной системы должны быть разделены (см. паттерны `Acceptor/Connector`).

Обработка входящих сообщений должна быть реализована с применением паттерна `Reactor`. Для *лабораторной работы №4a* достаточно использовать однопоточный реактор. Отличие *лабораторной работы №4b* заключается в необходимости использовать многопоточный реактор (различным потокам делегируется обработка соответствующих типов сообщений).

Для проверки корректности работы системы необходимо реализовать алгоритм взаимного исключения Лэмпорта из этапа №3 лабораторной работы №1 (вместо функции `print()` можно использовать любой другой способ вывода).

Сохранение формата используемых сообщений рекомендуется, но не является обязательным.

### Требования к реализации и среда выполнения

Подробнее ознакомиться с внутренним устройством используемых паттернов можно на лекциях и в соответствующих лекционных презентациях. Описание классов, представленных в фреймворке ACE, находится в документации к фреймворку (для версии 6.5.2 - <http://www.dre.vanderbilt.edu/Doxygen/6.5.2/html/libace-doc/annotated.html>).

Актуальную версию фреймворка можно получить по ссылке: <http://download.dre.vanderbilt.edu/>

Реализацию необходимо выполнить на языке программирования C++, допускаются современные стандарты (такие как C++11, C++14, C++17). Грамотное использование средств ООП и хороших C++ практик приветствуется.

Рекомендуемая среда выполнения — Linux (Ubuntu версии  $\geq 12.10$ ). Рекомендуемая версия фреймворка ACE – 6.5.2 или новее.