Чат

Реализовать:

1. **асинхронный**: не должно быть ни одной синхронной операции ввода/вывода, предполагается использование набора технологий типа *boost::asio*, *java::nio2*, *Qt*...
2. **неблокирующий**
3. **многопоточный**: сервер должен принимать как параметр командной строки число потоков в пуле, который будет разгребать сетевые события

чат, работающий по единому протоколу, определённому в нотации *protobuf* и учитывающему как минимум имя пользователя:

```
// ------------------------------------------------------------------
/// \brief ChatMessage - сообщение, передаваемое между сервером и клиентами
// ------------------------------------------------------------------
message ChatMessage {
    required string Sender = 1; ///< Идентификатор отправителя
    required string Text = 2;   ///< Содержание сообщения
    optional string Data = 3;   ///< Internal field. Можно использовать для
реализации каких-либо
}                 ///< собственных фич у своего сервера/клиента. Однако, стоит помнить,
                  ///< что все сервера-клиенты должны быть совместимы друг с другом
```

В качестве механизма доставки сообщений между сторонами предлагается использовать протокол TCP. Этот протокол не допускает дублирование переданных пакетов и гарантирует доставку отправленных сообщений в рамках установленного соединения. Так как протокол TCP гарантирует доставку и правильную очередность данных, то для выделения отдельных сообщений из потока байт мы можем использовать самый простой из существующих способов - перед отправкой сообщения в канал сначала передавать его размер. Отметим, что большая часть сообщений, передаваемых в рамках сеанса управления, имеет небольшой размер. Поэтому использовать для представления размера сообщения фиксированное количество байт, позволяющее передать размер максимально возможного сообщения (например - 4 байта), черезчур избыточно. Поэтому, для представления размера сообщения мы будем использовать целое число в т.н. кодировке переменной длины (variable-length encoding), которая кодирует числа минимально возможным количеством байт и рекомендована Google. Функции для работы с этой кодировкой присутствуют в библиотеке Google Protocol Buffers.

Оганичения на язык: нет
Интерфейс пользователя: опционален

From:
http://wiki.osll.ru/ - **Open Source & Linux Lab**

Permanent link:
http://wiki.osll.ru/doku.php/courses:high_performance_computing:chat

Last update: **2018/04/01 20:27**