Программа сделана по заданию, которое приложено.

**Клиент.**

Консольное приложение позволяет вводить 8 байтовые числа и отправляет их на сервер. Клиент отслеживает состояние подключения отправляя два раза в секунду сообщение ping (keep alive). Имеется возможность задавать адрес сервера и порт параметрами командной строки. Формат параметров можно узнать через параметр --help.

**Сервер.**

Факторизует числа и отправляет результат клиенту. Сервер ставит вычислительные задачи в очередь и отправляет клиенту по мере их готовности. Имеется возможность делать задачи одновременно в несколько потоков. Сервер может работать с несколькими клиентами одновременно.

Сервер можно запускать как консольное приложение, как windows сервис или unix daemon. Сервер поддерживает чтение файла конфигурации и логирование. Файл конфигурации должен лежать в папке с выполняемым файлом сервера, иметь одинаковое с ним имя и одно из следующих расширений: properties, xml, ini. В файле конфигурации возможны следующие параметры: port – порт который сервер будет слушать, threads – количество потоков на которых будут выполняться задачи, конфигурация логирования. Пример файла конфигурации с логированием есть в архиве сервера.

**Сетевой протокол.**

Клиент имеет два типа пакетов, ping и data. Первый байт отвечает за тип. Ping состоит из одного байта (типа). Data имеет тип, 4-х байтный номер сессии и 64-х битное число введенное пользователем.

Пакет сервера состоит из 4-х байтного номера сессии, 4-х байтной длины массива и собственно самого массива - результата.

**Сборка и зависимости.**Клиенту необходим компилятор поддерживающий С++11 и boost. Серверу С++11, boost и библиотека Poco.

Система сборки это cmake не ниже версии 3.0. Для того чтобы собрать клиент или сервер под unix надо зайти в папку bin и выполнить команду «cmake -G 'Unix Makefiles' ..» и в случае успеха make. Сервер содержит один юнит тест для проверки функции факторизации, который можно выполнить после сборки, в этой же папке bin командой «ctest». Программы собираются в папках bin/Source Files.

Приложение было протестировано под windows 7 и linux Ubuntu, под x86 и AMD64. Сборка производилась Visual Studio 2013 и GNU g++ 4.9.2

**Примечание.**

Алгоритм факторизации был выбран самый простой из возможных. Это позволяет хорошо показать параллельность и независимость работы клиента и сервера. Интерфейс клиента тоже самый простой, он имеет особенность что если в момент набора числа пользователем приходит результат от сервера, то вывод результата пересекается с вводом. Предполагается что числа будут положительными, но можно ввести и отрицательное, в этом случае оно будет сконвертировано в положительное. Таким образом самое большое число это -1 и сервер вычисляет его дольше всех.