Лабораторная работа 08

ПвИ

**Задание 01**

1. Разработайте приложение **08-01**, представляющее собой TCP-сервер. Сервер должен через TCP-соединение принимать строковое сообщение от TCP-клиента и возвращать клиенту текст полученного сообщения с префиксом **ECHO:**.
2. Разработайте приложение **08-01a**, представляющее собой TCP-клиента, проверяющего работоспособность сервера 08-01.
3. Разработайте TCP-сервер на языке С++, выполняющий те же функции, что и сервер 08-01. Проверьте его работоспособность с клиентом 08-01a.
4. Разработайте TCP-клиента на языке С++, выполняющего те же функции, что и клиент 08-01a. Проверьте его работоспособность с сервером 08-01.

**Задание 02**

1. Разработайте приложение **08-02**, представляющее собой TCP-сервер, прослушивающий 2 порта: 40000, 50000. Сервер должен через TCP-соединение принимать поток 32-битовых чисел (по одному числу за каждую отправку клиентом). Сервер суммирует полученные числа и каждые 5 сек. отправляет клиенту полученную (промежуточную) сумму. Сервер обеспечивает каждому подключенному клиенту получение правильных промежуточных сумм чисел, отправленных клиентом серверу. Сервер должен обеспечивать вывод на консоль диагностических сообщений, позволяющих проверить корректность его работы.
2. Разработайте приложение **08-02a**, представляющее собой TCP-клиента. Клиент принимает 2 числовых параметра (номер порта сервера и число **X**) через командную строку. Клиент через TCP-соединение отправляет 1 раз в секунду серверу 32-битовое число **X**. Клиент принимает от сервера промежуточные суммы и выводит их на консоль.
3. Проверьте работоспособность сервера с двумя клиентами, устанавливающими соединение с сервером через разные порты и отправляющими разные числа.
4. Проверьте работоспособность сервера с тремя клиентами, устанавливающими соединение с сервером через разные порты и отправляющими разные числа.

**Задание 03**

1. Разработайте приложение **08-03**, представляющее собой UDP-сервер. Сервер должен принимать строковое сообщения от UDP-клиента и возвращать клиенту текст полученного сообщения с префиксом **ECHO:**.
2. Разработайте приложение **08-03a**, представляющее собой UDP-клиента, проверяющего работоспособность сервера 08-03.
3. Разработайте UDP-сервер на языке С++, выполняющий те же функции, что и сервер 08-03. Проверьте его работоспособность с клиентом 08-03a.
4. Разработайте UDP-клиента на языке С++, выполняющего те же функции, что и клиент 08-03a. Проверьте его работоспособность с сервером 08-03.

**Задание 04.** Ответьте на следующие вопросы.

1. Поясните основные свойства протокола TCP.

Надежность: Гарантирует доставку данных без потерь и в правильном порядке.

Управление потоком: Регулирует скорость передачи данных между отправителем и получателем.

Установка соединения: Имеет процедуры для установки и разрыва соединения между устройствами.

1. Поясните процедуры установки и закрытия TCP-соединения.

Установка соединения (3-ступенчатый рукопожатия):

Шаг 1 - Синхронизация (SYN): A отправляет B сегмент с установленным флагом SYN (синхронизация) и выбранным начальным порядковым номером.

Шаг 2 - Подтверждение (SYN+ACK): B получает сегмент, устанавливает свой порядковый номер, устанавливает флаг SYN и ACK (подтверждение), а затем отправляет его обратно A.

Шаг 3 - Подтверждение (ACK): A получает сегмент, устанавливает флаг ACK и отправляет его B, завершая процедуру установки соединения.

Закрытие соединения (4-ступенчатая процедура):

Шаг 1 - Завершение передачи данных (FIN): A отправляет B сегмент с установленным флагом FIN (завершение передачи данных), указывая, что A закончил отправку данных.

Шаг 2 - Подтверждение (ACK): B получает сегмент, устанавливает флаг ACK и отправляет его A в знак того, что он принял завершение передачи данных.

Шаг 3 - Завершение передачи данных (FIN): B отправляет A сегмент с флагом FIN, указывая, что B закончил передачу данных.

Шаг 4 - Подтверждение (ACK): A получает сегмент, устанавливает флаг ACK, завершая процедуру закрытия соединения.

1. Поясните понятие «порт».

Определенный номер: Идентификатор процесса или службы на конкретном устройстве.

16-битное значение: Диапазон от 0 до 65535, где определенные диапазоны зарезервированы для определенных служб (например, порты 80 для HTTP).

1. Поясните понятие «сокет».

программный интерфейс, представляющий собой совокупность IP-адреса и номера порта.

В Node.js представляет собой экземпляр класса net.Socket.

1. Поясните понятие «полудуплексный канал связи».

двусторонний канал связи, данные по нему могут передаваться в одном направлении в одинмомент времени. Оба абонента имеют возможность принимать и передавать сообщения.Поток сообщений может идти в обоих направлениях, но не одновременно.Пример: рация, HTTP

1. Поясните понятие «дуплексный канал связи».

двусторонний канал связи, данные могут передаваться в обе стороны одновременно. Дуплекс имеет два физически отдельных канала связи, один из которых предназначен для движения данных в одном направлении, а другой – в противоположном направлении.

1. Поясните отличие протокола UDP от TCP.

Отсутствие механизмов обеспечения надежности(пакеты не упорядоч, прием не подтверждается), отсутствие гарантий доставки, отсутствие обработки соединений, отсутствует проверка контрольной суммы, отсутствие буферизации, отсутствует фрагментация.