Лабораторная работа 13

ПСКП

ПОИТ-3

**Задание 01**

1. Разработайте приложение **13-01**, 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 **13-02**, представляющее собой TCP-клиента, проверяющего работоспособность сервера 13-01.
3. Разработайте TCP-сервер на языке С++, выполняющий те же функции, что и сервер 13-01. Проверьте его работоспособность с клиентом 13-02.
4. Разработайте TCP-клиента на языке С++, выполняющего те же функции, что и клиент 13-02. Проверьте его работоспособность с сервером 13-01.

**Задание 02**

1. Разработайте приложение **13-03**, представляющее собой TCP-сервер. 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) общую сумму.
2. Разработайте приложение **13-04**, представляющее собой TCP-клиента. Клиент через TCP-соединение отправляет 1 раз в секунду серверу 32-битовое число. Клиент принимает от сервера промежуточные общие суммы и выводит их на консоль. Клиент должен автоматически остановиться через 20 сек.
3. Продемонстрируйте работу сервера 13-03 с двумя экземплярами клиента 13-04.

**Задание 03**

1. Разработайте приложение **13-05**, представляющее собой TCP-сервер. 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 **13-06**, представляющее собой TCP-клиента. Клиент принимает 1 числовой параметр (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, запущенными с разными значениями параметров.
4. Проверьте работоспособность сервера с тремя клиентами, запущенными с разными значениями параметров.

**Задание 04**

1. Разработайте приложение **13-07**, представляющее собой TCP-сервер, прослушивающий 2 порта: 40000, 50000. Сервер должен через TCP-соединения принимать потоки 32-битовых чисел (по одному числу за каждую отправку клиентом). Сервер возвращает клиенту полученное сообщение (число) с префиксом **ECHO:**.
2. Разработайте приложение **13-08**, представляющее собой TCP-клиента. Клиент принимает 1 числовой параметр (номер порта) через командную строку. Клиент через TCP-соединение отправляет 1 раз в секунду серверу 32-битовые числа **X**. Клиент принимает от сервера сообщения и выводит их на консоль.
3. Продемонстрируйте работу сервера с двумя клиентами 13-08, устанавливающими соединение с сервером через разные порты.

**Задание 05**

1. Разработайте приложение **13-09**, представляющее собой UDP-сервер. Сервер должен принимать строковое сообщения от UDP-клиента и возвращает клиенту текст полученного сообщения с префиксом **ECHO:**.
2. Разработайте приложение **13-10**, представляющее собой UDP-клиента, проверяющего работоспособность сервера 13-09.
3. Разработайте UDP-сервер на языке С++, выполняющий те же функции, что и сервер 13-09. Проверьте его работоспособность с клиентом 13-10.
4. Разработайте UDP-клиента на языке С++, выполняющего те же функции, что и клиент 13-10. Проверьте его работоспособность с сервером 13-09.

**Задание 06.**Ответьте на следующие вопросы.

1. Поясните основные свойства протокола TCP.
2. Поясните процедуры установки и закрытия TCP-соединения.
3. Поясните понятие «порт».
4. Поясните понятие «сокет».
5. Поясните понятие «полудуплексный канал связи».
6. Поясните понятие «дуплексный канал связи».
7. Поясните отличие протокола UDP от TCP.