МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Костромской государственный университет»

(КГУ)

ИАСТ

(наименование института)

Кафедра автоматизированных систем и технологий

(наименование кафедры)

09.03.02

Направление подготовки/Специальность Информационные системы и технологии

(наименование направления подготовки/специальности)

Дисциплина Архитектура ЭВМ

(наименование дисциплины)

Лабораторная работа №5.

Сетевые соединения.

Выполнили студенты Копосов Лев Владимирович

Копосов Владимир Владимирович

(фамилия, имя, отчество)

Группа 22-ИСбо-1б

Проверил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество)

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_

Кострома

Вопросы:

1. В чем разница между потоковым (SOCK\_STREAM) и датаграммным (SOCK\_DGRAM) соединениями?

При использовании датаграммного сокета постоянное соединение не устанавливается, каждая датаграмма отправляется независимо от других. Факт и порядок доставки индивидуальной датаграммы определяется качеством сетевых каналов между узлами. Приложению приходится самостоятельно отличать сообщения от разных клиентов. Пример применения: протоколы, которые предусматривают короткий обмен «запрос- ответ».

Потоковый сокет позволяет обеспечить установление и поддержание соединения, а также гарантированную передачу и порядок передачи пакетов. Каждый клиент имеет свой объект - сокет, что упрощает их различение. Пример применения: протоколы, которые предусматривают длительный сеанс работы с одним сервером, где важен контроль целостности данных.

1. Что делает функция bind()?

После создания сокета, следует вызвать функцию bind() для указания адреса и порта. Функция связывает их с созданным сокетом. Это позволяет серверу обрабатывать входящие соединения от клиентов.

1. Как сервер может различать данные, поступающие от разных клиентов, при использовании протокола TCP? А при использовании UDP?

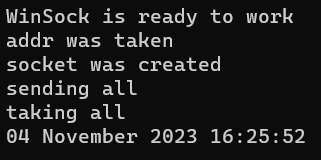
При использовании протокола TCP сервер может различать данные, поступающие от разных клиентов, по их IP- адресу и порту. Каждое подключение к серверу имеет уникальную комбинацию IP- адреса и порта, что позволяет серверу идентифицировать каждого клиента.

При использовании протокола UDP сервер не имеет установленного соединения с клиентом, поэтому он не может идентифицировать клиентов по IP- адресу и порту. Сервер может различать данные, поступающие от разных клиентов, по уникальному идентификатору, передаваемому вместе с данными. Например, специальный идентификатор, который генерируется клиентом.

Вывод в консоль.

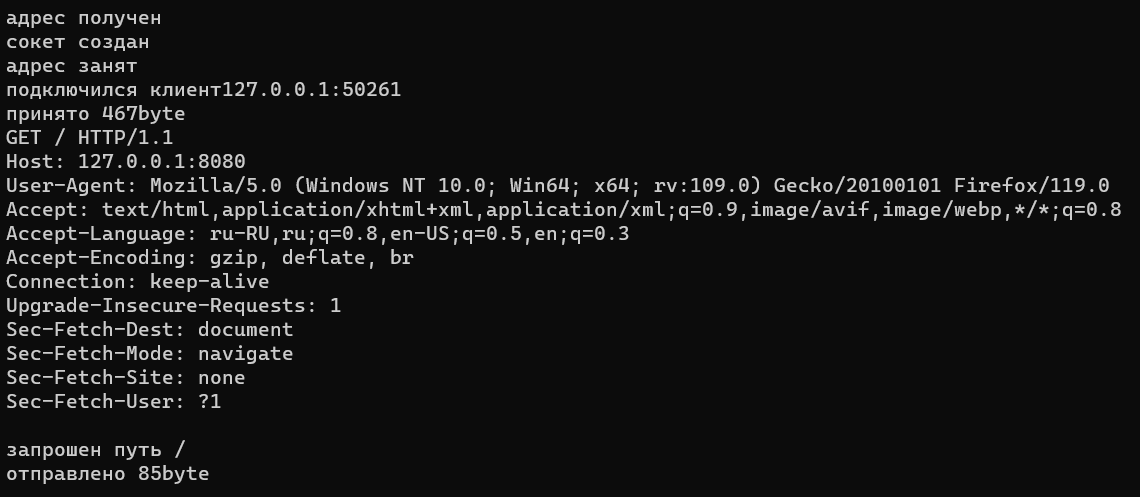
Задание 1.

Получить текущее время по протоколу NTP с сервера pool.ntp.org:123 (или другого доступного). Используйте getaddrinfo() для получения структуры с адресом, и передавайте данные по протоколу UDP. Формат сообщения NTP, а также сведения о функциях форматирования даты/времени приведены в справочных материалах.



Задание 2.

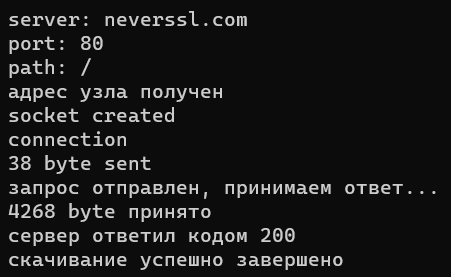
Создать простой TCP-сервер, слушающий порт 8080 по адресу 127.0.0.1 и отвечающий на запросы по протоколу HTTP/1.0 (без шифрования). Допустимо последовательное обслуживание клиентов. Определение содержимого ответа должно быть вынесено в отдельную подпрограмму. Проверьте корректность работы сервера, обратившись браузером по используемому адресу.





Задание 3.

Создать простой TCP-клиент, выполняющий запрос по протоколу HTTP/1.0 (без шифрования), и сохраняющий содержимое ответа в указанный файл. Проверьте корректность работы клиента, обратившись к вашему серверу из задания 2, или к любому серверу, поддерживающему протокол HTTP/1.0.



Дополнительное задание.

Измените программу из задания 2 таким образом, чтобы, получив от клиента запрос "/favicon.ico", она передавала клиенту содержимое приложенного к заданию файла со следующими параметрами:

* Размер равен размеру файла
* MIME-тип "image/x-icon"

