

Лабораторная работа № 3

Работа с системой доменных имён DNS

3.1. Цель работы:

Ознакомиться с системой доменных имён DNS и способами работы с данной системой с помощью средств библиотеки Boost::Asio.

3.2. Задание на лабораторную работу:

Разработать приложение, которое позволяет по известному имени хоста (доменному имени) определить IP адрес устройства в сети. Приложение должно использовать простейший графический интерфейс.

3.3. Методические указания по системе DNS:

DNS (Domain Name System) – иерархичная распределённая система доменных имён. Данная система позволяет по известному имени хоста (доменному имени) определить IP адрес устройства в сети, а также по известному IP адресу определить имя хоста (доменное имя). DNS значительно упрощает жизнь пользователей глобальной сети интернет, так как символьное выражение доменного имени (например “www.yandex.ru”) намного легче запомнить и использовать в дальнейшем, нежели реальный IP адрес (например “213.180.193.3”).

DNS является распределённой структурой, состоящей из множества DNS-серверов. Данная структура иерархична, что позволяет распределить ответственность среди разных уровней DNS-серверов.

Приведём пример использования системы DNS:

- 1) Пользователь глобальной сети интернет захотел проверить свою электронную почту, для этого он использует web-сервис www.yandex.ru. Но пользователь знает только доменное имя, которое он вводит в строку поиска своего браузера.

- 2) Браузер прочитав данное доменное имя обращается к DNS-серверу провайдера пользователя.
- 3) DNS-сервер ищет в своей базе данных полученное доменное имя. В случае успеха поисков он возвращает IP адрес для запрашиваемого web-сервиса.
- 4) В случае если в базе данных DNS-сервера не обнаружено записи о запрашиваемом доменном имени, DNS-сервер направляет тот же запрос к другим DNS-серверам с более высоким уровнем иерархии. Данные действия повторяются, пока искомое доменное имя не будет найдено или все уровни иерархии DNS-серверов не будут пройдены.
- 5) Как только ваш браузер узнаёт IP адрес запрашиваемого доменного имени, он начинает подключение к данному адресу.

3.4. Работа с системой DNS с помощью библиотеки Boost::Asio.

Для работы с системой DNS используется специализированный класс `boost::asio::ip::tcp::resolver`. Данный класс предоставляет функции для асинхронной работы с DNS.

Перед использованием этого объекта необходимо создать объект `io_service` для обработки асинхронных операций.

Таблица 3.1

Конструктор	<code>basic_resolver(boost::asio::io_service & io_service)</code>
Описание параметров	<code>io_service</code> – объект для работы с асинхронными операциями

Пример:

```
boost::asio::io_service io_service;
```

```
boost::asio::ip::tcp::resolver resolver(io_service);
```

После создания объекта необходимо вызвать асинхронную функцию `async_resolve()`, которая запустит механизм поиска доменного имени и определит вызываемый обработчик. А также необходимо вызвать метод `run()` объекта `io_service`.

Таблица 3.2

Функция	<pre>template< typename ResolveHandler > void async_resolve(const query & q, ResolveHandler handler)</pre>
Описание функции	Запускает механизм поиска доменного имени, хранящегося в объекте q, и определяет вызываемый обработчик handler
Описание параметров	q – объект запроса, хранящий запрашиваемое доменное имя handler – обработчик запроса

Для того, чтобы правильно сохранить доменное имя необходимо создать объект `resolver::query`.

Таблица 3.3

Конструктор	<pre>basic_resolver_query(const std::string & host, const std::string & service,)</pre>
Описание параметров	host – доменное имя искомого хоста в сети service – служба, необходимая для идентификации запрашиваемых услуг (для поиска доменного имени необходимо использовать службу "http")

Пример:

```
resolver.async_resolve(
    boost::asio::ip::tcp::resolver::query("www.yandex.ru", "http"),
    boost::bind(handle_resolve_query, _1, _2)
);
```

Сигнатура функции обработчика должна быть:

```
void handler(
    const boost::system::error_code& error,
    resolver::iterator iterator
);
```

Если объект `err` существует, то произошла ошибка, в противном случае поиск успешно завершён. Для просмотра списка найденных адресов используем итератор `iterator`.

Пример:

```
boost::asio::ip::tcp::resolver::iterator end;
for (; iter != end; ++iter){
    boost::asio::ip::tcp::endpoint endpoint = *iter;
```

```
        ui->ipLineEdit->setText(QString::fromStdString(endpoint.address().to_string()));  
    }
```

3.5. Контрольные вопросы:

- 1) Для чего используется служба DNS?
- 2) Опишите процесс запроса доменного имени у DNS-сервера (DNS-сервер провайдера нашёл запись).
- 3) Опишите процесс запроса доменного имени у DNS-сервера (DNS-сервер одного из верхних уровней иерархии нашёл запись).
- 4) Опишите процесс запроса доменного имени у DNS-сервера (DNS-сервера не нашли запись).
- 5) Назовите основные функции и классы для работы с DNS выбранной вами библиотеки.