



Программа курса Сетевое программирование

Мережеве програмування
Network Programming

Для групп стационара. Версия 1.0.0

Продолжительность курса: 18 пар (9 дней)

Цель курса

Обучить слушателя разработке Windows-приложений с использованием механизмов сетевого взаимодействия. Получить теоретические и практические знания об основах сетевого программирования в Windows.

По окончании курса слушатель будет:

- уметь строить клиент-серверные приложения с использованием сокетов;
- использовать механизмы асинхронности при работе с сокетами;
- осуществлять широковебательную рассылку;
- посылать почту с использованием пространства System.Net.Mail;
- получать и посылать данные по FTP протоколу;
- использовать протокол HTTP для получения и отправки данных.

По окончании данного курса студент сдает практическое задание и теоретический экзамен по материалам курса. Для допуска к экзамену должны быть сданы все домашние и практические задания.

Практическое задание должно охватывать максимум материала из различных разделов курса.

Тематический план

Модуль 1.	Введение в сети	2 пары
Модуль 2.	Сокеты	2 пары
Модуль 3.	TCP и UDP сокеты	6 пар
Модуль 4.	Unicast, Broadcast, Multicast	2 пары
Модуль 5.	HTTP, SMTP, FTP	4 пары
Модуль 6.	Экзамен	2 пары

Модуль 1

Введение в сети

1. Что такое сетевое программирование?
2. Цели и задачи сетевого программирования.
3. Что такое сеть?
4. Типы сетей.
5. Модель OSI.
6. Базовые термины:
 - сетевые протоколы;
 - IP адрес;
 - сокет;
 - порт.

Модуль 2

Сокеты

1. Типы сокетов.
2. Класс Socket:
 - общая последовательность вызовов для серверной и клиентской части;
 - цели и задачи класса Socket;
 - анализ методов необходимых для построения серверной части:
 - метод Bind;
 - метод Listen;
 - метод Accept;
 - методы Receive/Send;
 - метод Close.
 - анализ методов необходимых для построения клиентской части:
 - метод Connect;
 - методы Receive/Send;
 - метод Close.

3. Пример построения клиент/серверного приложения с использованием сокетов.

4. Асинхронные сокет:

- цели и задачи асинхронных сокетов;
- общая последовательность вызовов для серверной и клиентской части;
- анализ методов необходимых для построения серверной части:
 - метод BeginAccept;
 - метод BeginReceive;
 - метод BeginSend;
 - метод EndAccept;
 - метод EndReceive;
 - метод EndSend.
- анализ методов необходимых для построения клиентской части:
 - метод BeginConnect;
 - метод BeginReceive;
 - метод BeginSend;
 - метод EndConnect;
 - метод EndReceive;
 - метод EndSend.

5. Пример построения клиент/серверного приложения с использованием асинхронных сокетов.

Модуль 3

TCP и UDP сокет

1. Обзор TCP протокола:

- общий обзор;
- терминология TCP;
- TCP заголовки;
- преимущества и недостатки TCP.

2. Обзор UDP протокола:

- общий обзор;
- терминология UDP;
- преимущества и недостатки UDP.

3. Использование классов, характерных для TCP:

- класс TcpListener;
- класс TcpClient;
- пример использования.

4. Использование классов, характерных для UDP:

- класс UdpClient;
- пример использования.

5. Использование сокетов в программе на C++:

- функция socket;
- функция setsockopt;
- функция bind;
- функция listen;
- функция accept;
- функция connect;
- практические примеры использования.

Модуль 4

Unicast, Broadcast, Multicast

1. Что такое Unicast?

2. Что такое Broadcast?

3. Что такое Multicast?

4. Пример реализация Multicast приложения.

Модуль 5

HTTP, SMTP, FTP

1. Обзор HTTP протокола:

- общий обзор;
- терминология HTTP;
- HTTP заголовки;
- отправка запросов и получение ответов.

2. Классы для работы с HTTP:

- отправка запросов с использованием класса HttpRequest;
- получение ответов с использованием класса HttpResponse;

- использование класса WebClient;
- примеры использования.

3. Работа с электронной почтой:

- общий обзор почтовых протоколов SMTP, POP3, IMAP;
- SMTP:
 - общий обзор;
 - терминология SMTP;
 - команды SMTP;
 - коды ответов;
 - анализ формата электронного письма;
 - что такое MIME?
- POP3:
 - общий обзор;
 - терминология POP3;
 - команды POP;
 - коды ответов.
- IMAP:
 - общий обзор.
- классы для работы SMTP:
 - пространство System.Net.Mail;
 - класс MailMessage;
 - класс Attachment;
 - класс Smtplib;
 - пример отправки почты.

4. Использование FTP:

- общий обзор;
- терминология FTP;
- пример типичной FTP сессии;
- классы для работы с FTP:
 - класс FtpWebRequest;
 - класс FtpWebResponse.
- пример использования FTP.

Модуль 6

Экзамен