ЛЕКЦИЯ №6

# 1. СОЗДАНИЕ КЛИЕНТ - СЕРВЕРНЫХ СИСТЕМ

# С ИСПОЛЬЗОВАНИЕМ СОКЕТНОЙ ТЕХНОЛОГИИ ПРОТОКОЛА TCP/IP

# В настоящее время наиболее распространенным протоколом передачи данных в локальной сети и Интернет является протокол TCP/IP. Данный протокол позволяет передавать любые данные на любое расстояние. Протокол – это специальное программное обеспечение, входящее в операционную систему, которое реализует прием и передачу данных по сети.

# При связи двух программ (компьютеров) между собой по сети через протокол TCP/IP для каждой программы операционная система создает (открывает) свой «сокет». Сокет – это буфер (канал) обмена данными по сети. Сокет открывается для конкретного экземпляра программы для указанного сетевого адреса (идентифицирующий компьютер в сети) и порта TCP/IP (идентифицирующий конкретную сетевую программу). Каждый сокет получает от операционной системы уникальный номер (дескриптор, хэндл).

# При обмене данными по протоколу TCP/IP одна машина обычно выступает в роли "сервера", другая - в роли "клиента". Клиент, как и сервер, может передавать и принимать данные, отличие между ними заключается в том, что программа "сервер" может одновременно работать с несколькими "клиентами", а "клиент" - только с одним сервером. На рис. 1 показана схема модели клиент – серверного взаимодействия на основе сокетов для протокола TCP/IP.

# Для создания сокетного клиента TCP/IP в Делфи используется компонент «ClientSocket», а для создания сокетного сервера компонент «ServerSocket». Используя свойства, методы и события данных компонент создадим программы «клиент» и «сервер».

# 2. СОЗДАНИЕ ПРОГРАММЫ TCP/IP - КЛИЕНТ НА ОСНОВЕ СОКЕТОВ

Компоненты программы – клиента:

1. ClientSocket1: TClientSocket - компонент сокета клиента;
2. Edit1 - IP-адрес сервера;
3. Edit2 -порт TCP/IP;
4. Memo1 - посылаемые на сервер данные;
5. ListBox1 – получаемые с сервера данные;
6. Button1 - активизация клиента (подключение);
7. Button2 - посылка данных;
8. Button3 - отключение клиента.

Свойства ClientSocket1:

1. ClientType – тип передачи данных по протоколу TCP/IP (блочная передача, не блочная);
2. Active – открытие/закрытие связи с сервером (сокета);
3. Address – IP адрес сервера TCP/IP;
4. Host - имя сервера TCP/IP (задается либо адрес, либо хост сервера);
5. Port – номер порта TCP/IP сервера;
6. Socket – объект доступа к сокету.

События ClientSocket1:

В процедуры-обработчики данных событий

передается параметр «Socket» типа

“TCustomWinSocket”. Данный параметр имеет

свойство «Socket . ReceiveText» - хранящий принятые с сервера данные.

# OnConnect – подключение к серверу;

# OnDisconnect – отключение от сервера;

# OnError – ошибка при подключении к серверу;

# OnRead – чтение данных с сервера.

Методы ClientSocket1:

* Socket . SendText – отправка текста на сервер;

# Open – подключение к серверу (открытие сокета);

# Close – отключение от сервера (закрытие сокета).

# Рис. 1 Модель клиент – серверного взаимодействия на основе сокетов TCP/IP

Операционная система PC 1

### Программа – сервер TCP/IP

## SERVERSOCKET

## *Процедура чтения данных с клиентов*

## *Процедура отправки данных на клиенты*

#### Socket N1

#### Socket N2

#### Socket N3

Операционная система PC 2

### Программа – клиент TCP/IP

## CLIENTSOCKET

## *Процедура чтения данных с сервера*

## *Процедура отправки данных на сервер*

#### Socket NN1

Клиент N2

Клиент N3

procedure TForm1.Button1Click(Sender: TObject);

{ Подключение к серверу }

begin

{ Если соединение уже установлено - прерываем его }

ClientSocket1.Close;

{ Устанавливаем не блочную передачу/прием данных }

ClientSocket1.ClientType:=ctNonBlocking;

{ Присваиваем свойствам Address и Port нужные значения }

ClientSocket1.Address := Edit1.Text; ClientSocket1.Port := StrToInt(Edit2.Text);

{ Пытаемся открыть сокет и установить соединение с сервером }

ClientSocket1.Open;

end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);

{ Отключение от сервера }

begin

ClientSocket1.Close;

end;

procedure TForm1.ClientSocket1Read(Sender: TObject; Socket: TCustomWinSocket);

{ Прием текста от сервера }

begin

ListBox1.Items.Text:=Socket.ReceiveText;

end;

procedure TForm1.Button2Click(Sender: TObject);

{ Отправка текста на сервер }

begin

if ClientSocket1.Socket.SendText(Memo1.lines.Text)=0 then

MessageDlg('Не могу отправить данные на сервер',

mtError,[mbOk], 0) else showmessage('Ok!');

end;

procedure TForm1.ClientSocket1Error(Sender: TObject;

Socket: TCustomWinSocket; ErrorEvent: TErrorEvent; var ErrorCode: Integer);

{ Произошла ошибка соединения с сервером }

begin

ErrorCode:=0; MessageDlg('Нет связи с сервером', mtError,[mbOk], 0);

end;

procedure TForm1.Button3Click(Sender: TObject);

{ Отключение от сервера }

begin

ClientSocket1.Close;

end;

procedure TForm1.ClientSocket1Connect(Sender: TObject; Socket: TCustomWinSocket);

{ Факт подключения к серверу }

begin

ListBox1.Items.Text:='On line...';

end;

procedure TForm1.ClientSocket1Disconnect(Sender: TObject;

Socket: TCustomWinSocket);

{ Факт отключения к серверу }

begin

ListBox1.Items.Text:='Off line...';

end;

# 3. СОЗДАНИЕ ПРОГРАММЫ TCP/IP - СЕРВЕР НА ОСНОВЕ СОКЕТОВ

Компоненты программы – сервера:

1. ServerSocket1: TServerSocket – компонент - сокет сервера TCP/IP;
2. Edit1 - IP-адрес сервера;
3. Memo1 - посылаемые на клиент данные;
4. ListBox1 - полученные от клиента данные;
5. Button1 - запуск сервера;
6. Button2 - остановка сервера;
7. Button3 - отправка данных на клиенты.

Свойства ServerSocket1:

1. Active – запуск/остановка сервера (сокета);
2. Port – номер порта TCP/IP сервера;
3. Socket – объект доступа к сокету;
4. Socket . ActiveConnections – количество подключенных к серверу клиентов.

События ServerSocket1:

В процедуры-обработчики данных событий передается параметр

«Socket» типа “TCustomWinSocket”.

Данный параметр имеет свойство

«Socket . ReceiveText» - хранящий принятые с клиента данные.

# OnClientConnect – подключение клиента к серверу;

# OnClientDisconnect – отключение клиента от сервера;

# OnClientError – ошибка связи с клиентом;

# OnClientRead – получение данных от клиента;

# OnListen – ожидание подключения клиента.

Методы ServerSocket1:

* Socket . Connections[i] . SendText – отправка на i-й клиент текста;

# Open – активизация сервера (открытие сервера);

# Close – деактивизация сервера (закрытие сервера).

# *(!) Каждый сокет имеет свойство «Socket . SocketHandle» - уникальный дескриптор данного сокета, используемый для идентификации клиента (в обработчике события «OnClientRead».*

procedure TForm2.Button1Click(Sender: TObject);

{ Определяем порт и запускаем сервер }

begin

ServerSocket1.Close; ServerSocket1.Port := StrToInt(Edit1.Text); ServerSocket1.Open;

end;

procedure TForm2.Button2Click(Sender: TObject);

{ Останавливаем сервер }

begin

ServerSocket1.Close; ListBox1.Items.Text:='Server is stop.';

end;

procedure TForm2.ServerSocket1ClientError(Sender: TObject;

Socket: TCustomWinSocket; ErrorEvent: TErrorEvent; var ErrorCode: Integer);

{ Произошла ошибка связи с клиентом }

begin

MessageDlg('Нет связи с клиентом', mtError,[mbOk], 0);

end;

procedure TForm2.ServerSocket1ClientRead(Sender: TObject;

Socket: TCustomWinSocket);

{ Получение данного с клиента }

begin

Listbox1.Items.Text:=Socket.ReceiveText;

end;

procedure TForm2.Button3Click(Sender: TObject);

{ Отправка данных на клиент }

label 1;

begin

with ServerSocket1.Socket do begin

if ActiveConnections=0 then goto 1;

if Connections[0].SendText(Memo1.Lines.text)=0 then

1: MessageDlg('Не могу отправить данные на клиент',

mtError,[mbOk], 0) else showmessage('Ok!');

end;

end;

procedure TForm2.ServerSocket1ClientConnect(Sender: TObject;

Socket: TCustomWinSocket);

{ Запрет доступа "вторым" клиентам }

begin

if ServerSocket1.Socket.ActiveConnections>1 then abort;

end;

procedure TForm2.FormClose(Sender: TObject; var Action: TCloseAction);

{ Остановка сервера }

begin

ServerSocket1.Close;

end;

procedure TForm2.ServerSocket1Listen(Sender: TObject;

Socket: TCustomWinSocket);

{ Ожидание подключения клиентов }

begin

ListBox1.Items.Text:='Server is active...';

end;