

До питання використання можливостей відкритої програмної бібліотеки компонентів при вивченні роботи із сокетами

Харченко В.М., Харченко М.В.

*Ніжинський державний університет ім. Миколи Гоголя, volmkhar@gmail.com;
Київський національний університет ім. Тараса Шевченка*

The authors have considered a capability of using open components library Indy 10 in Lazarus. They have described the features which are using blocking sockets when sending files in the network. Advantages of usage of Synapse library in network programming are listed.

При вивченні мережевого програмування на базі сокетів автори пропонують використовувати різні мови програмування. Зокрема, в [1] пропонується використовувати C++, в [2] – Java, а в [4] – Delphi для програмування в мережі. Якщо ж студенти вивчали програмування на основі Lazarus, то в цьому середовищі можна виконувати і завдання робіт з мережевого програмування. Опис відкритої програмної бібліотеки компонентів Indy 9, 10 можна знайти в [3; 4]. Ця бібліотека підтримує практично всі популярні мережеві протоколи типу TCP, UDP, SMTP, POP3, NNTP, HTTP тощо, що дає змогу програмувати досить складні мережеві проекти.

Зауважимо, на відміну від Delphi, в Lazarus слід самостійно установити відкриту програмну бібліотеку Indy 10. Це можна вважати своєрідною платою за безкоштовність: розпакування бібліотеки, рознесення по каталогах `src` і `lazarus`, видозміна файла конфігурації. На відміну від Turbo Delphi 2006 чи Delphi 7, в Lazarus компоненти розбиті на більшу кількість закладок панелі інструментів: `Indy Clients Protocols`, `Indy Servers Protocol` тощо. Це, на наш погляд, тільки полегшує пошук потрібної компоненти.

При виконанні такого роду лабораторних робіт до основних завдань відносять: пересилання одного або кількох файлів, пересилання повідомлень, пошук потрібної інформації в мережі Інтернет.

При розв'язуванні цих завдань з використанням Indy 10 доводиться наголошувати студентам, що у цій бібліотеці використовуються блокуючі сокети [3, с. 26] – на відміну від `Winsock`. При багатьох перевагах використання такого режиму є й недолік – виклик такого сокета не повертає управління до того часу, поки не буде виконана поставлена задача. А тому відбувається “заморожування” користувацького інтерфейсу. Це особливо часто трапляється під час пересилання файлів, розмір яких більший за 4 МВ. Щоб усунути дану проблему – необхідно використовувати в програмі компоненту `TIdAntiFreeze`.

Напевно, найскладніше дається програмування передачі кількох файлів. Обговорюючи написання цього алгоритму доводиться наголошувати на важливості використання конструкції обробки виняткових ситуацій типу `try/finally`, особливо у разі розробки сервера. Наголошувати доводиться і на

потребі перевірки відкритого потоку на атаку. Оскільки в цей момент деструктивні дії можуть привести до непередбачуваних наслідків. Під час написання серверної частини передачі файлу доречним буде шаблон:

```
procedure TForm1.IdTCPServer1Execute(AThread: TIdPeerThread);
var s:string;
    F: TFileStream;
begin
    s:=AThread.Connection.ReadLn;           // шлях, куди качати файл
    //перевірка s на атаку
    .....
    F:= TFileStream.Create(memo1.Lines.Strings[memo1.Lines.Count-1], fmCreate);
    // створюється файл
    try
    AThread.Connection.ReadStream(F,-1,true); // потік, що передається клієнтом
    finally
    end;
    AThread.Connection.Disconnect;
end;
```

Тоді досить легко здійснити програмування передачі кількох файлів від сервера до клієнта:

```
procedure TForm1.Button1Click(Sender: TObject);
var F: TFileStream;
    i: integer;
begin
    for i:=0 to memo1.Lines.Count-1 do
    begin
        IdTCPClient.Connect;
        IdTCPClient.WriteLn(memo2.Lines.Strings[i]); // шлях куди копіювати
        F:= TFileStream.Create(memo1.Lines.Strings[i],fmOpenRead);
    // що передавати
        try
        IdTCPClient.WriteStream(F);
        finally
        F.Free;
        end;
        IdTCPClient.Disconnect;
    end;
end;
```

Отже, використовуючи можливості більш ніж 100 компонентів бібліотеки Indy 10 можна створювати сучасні програми для роботи в мережах. Проте частина із даних компонентів функціонує ненадійно. Тому можна запропонувати студентам використовувати відкриту бібліотеку Synapse [5], яку в Lazarus можна використовувати без встановлення.

Джерела:

1. Григор'єв В.М. Лабораторний практикум із програмування у мережах TCP/IP за допомогою бібліотеки Winsock операційних систем Windows: Навч.-мет. посіб. [Текст] / В.М. Григор'єв. – Дніпропетровськ: ДНУ, 2009. – 148 с.
2. Методичні вказівки до лабораторних робіт з дисципліни «Проектування комп'ютерних мереж» для студентів фаху 7.05010201, 8.05010201 «Комп'ютерні системи та мережі» та 7.05010203, 8.05010203 «Спеціальні комп'ютерні системи», усіх форм навчання. Програмування. Розробка додатків. [Текст] / Укл.: Г.Г. Киричек, Т.М. Семерюк. – Запоріжжя: ЗНТУ, 2013. – 50 с.
3. Хувер Ч. Глубины Indy. [Текст] / Чад Хувер, Хади Харрири. – Atozed Computer Software Ltd, 2006. – 128 с.
4. Фленов М. Программирование в Delphi глазами хакера [Текст] / М. Фленов. – СПб.: БХВ-Петербург, 2003. – 368 с.
5. Synapse. [Електронний ресурс]. – Режим доступу: <http://synapse.ararat.cz/doku.php>