**7. РАЗРЕШЕНИЕ АДРЕСОВ В IP-СЕТЯХ**

**7.3. Определение MAC-адреса удаленного хоста по символьному адресу**

***7.3.1. Разработка программы определения MAC-адреса по DNS (NetBIOS) имени***

Задача опредления MAC-адреса по DNS-имени во многом аналогична уже рассмотренным задачам. Поэтому рассмотрим ее достаточно кратко (за более подробной информацией обратитесь к предыдущим разделам).

Во-первых необходимо включить в программу:

|  |
| --- |
| #include <stdio.h>  #include <iostream>  #include <windows.h>  #include <Winsock.h>  #include <Wsnetbs.h>  #include <Iphlpapi.h>  #pragma comment(lib,"WS2\_32.lib")  #pragma comment (lib, "Netapi32.lib")  #pragma comment(lib, "IPHlpApi.Lib")  using namespace std; |

**Инициализация Winsock**

Затем вызываем функцию WSAStartup – инициализируем библиотеку WS2\_32.dll:

|  |
| --- |
| int \_tmain(int argc, \_TCHAR\* argv[])  {  setlocale(LC\_ALL,"RUS");  WSADATA WsaData;  if(WSAStartup(0x0202,&WsaData)!=NULL)  {  cout<<"WSA Error!"<<endl;  return -1;  } |

**Разрешение имени в IP-адрес**

Вводим DNS-имя, проверяем введенное имя функцией CheckDNSName, работа которой была рассмотрена выше. Вызываем функцию gethostbyname, результат функции будет сохранен в структуре hostent:

|  |
| --- |
| hostent \*dns;  char\* host = new char[20];  SOCKADDR\_IN addr;  SOCKET udp\_sock;  do  {  cout<<"Host: ";  cin>>host;  }while(!CheckDNSName(host));  dns=gethostbyname(host);  if(dns==NULL)  {  cout<<"Не найдено"<<endl;  return -1;  } |

**Создание сокета**

Создаем сокет функцией socket, установив IP-адрес, извлеченный из структуры hostent:

udp\_sock = socket(AF\_INET,SOCK\_DGRAM,IPPROTO\_UDP);

**Передача данных**

Создаем объект, содержащий структуру sockaddr\_in и выполняем его инициализаицю. Отсылаем сообщение по сокету функцией sendto:

|  |
| --- |
| addr.sin\_family = AF\_INET;  addr.sin\_port = htons(1234);  addr.sin\_addr.S\_un.S\_addr =  \*(ULONG\*) dns->h\_addr\_list[0];  if(sendto(udp\_sock,"TESR",5,NULL,(SOCKADDR\*)&addr,  sizeof(addr))==NULL)  {  cout<<"Send"<<endl;  return -1;  } |

**Получение ARP-таблицы и MAC-адреса**

Через вызов функции GetIpNetTable извлекаем IPv4-таблицу маппирования физических адресов в структуру MIB\_IPNETTABLE:

|  |
| --- |
| MIB\_IPNETTABLE \*pTable =  (MIB\_IPNETTABLE\*) new char[0xFFFF];  ULONG cTable=0xFFFF;  if(GetIpNetTable(pTable,&cTable,true)!=NO\_ERROR)  {  cout<<"MAC not found"<<endl;  return -1;  } |

Из поля table объекта структуры MIB\_IPNETTABLE извлекаем физический адрес:

|  |
| --- |
| for(DWORD i=0;i<pTable->dwNumEntries;i++)  {  if(pTable->table[i].dwAddr ==  addr.sin\_addr.S\_un.S\_addr  && pTable->table[i].dwType!=2)  {  printf("IP: %i.%i.%i.%i\nMAC: %X-%X-%X-%X-%X-%X\n",  (int)addr.sin\_addr.S\_un.S\_un\_b.s\_b1,  (int)addr.sin\_addr.S\_un.S\_un\_b.s\_b2,  (int)addr.sin\_addr.S\_un.S\_un\_b.s\_b3,  (int)addr.sin\_addr.S\_un.S\_un\_b.s\_b4,  pTable->table[i].bPhysAddr[0],  pTable->table[i].bPhysAddr[1],  pTable->table[i].bPhysAddr[2],  pTable->table[i].bPhysAddr[3],  pTable->table[i].bPhysAddr[4],  pTable->table[i].bPhysAddr[5]);  }  }  closesocket( udp\_sock );  WSACleanup();  return 0;  } |

Результат работы программы приведен на рис. 7.6.

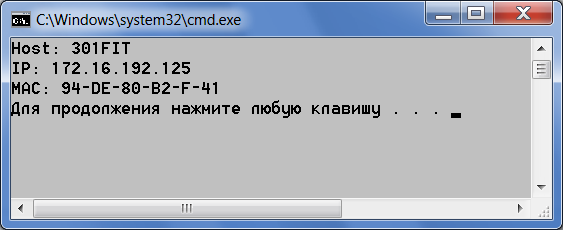


Рис. 7.6. Результат работы программы.

***7.3.2. Лабораторная работа № 13-14***

**Цель:** Изучение методов определения MAC-адреса удаленного хоста по символьному адресу.

**Задание:** Лабораторная работа заключается в разработке программы, выполняющей определение MAC-адреса удаленного хоста по введенному символьному (DNS, NetBIOS) адресу.