

Лабораторная работа №1.

Изучение сетевого окружения локального компьютера.

Цель работы: Изучение сетевого окружения локального компьютера на примере операционной системы Windows - NT.

Содержание работы: В процессе выполнения лабораторной работы должны быть изучены принципы адресации в сетях, структуры адресов локальных Host – ЭВМ и подсетей.

Каждое устройство, подключенное к Internet, имеет уникальный 32-х разрядный идентификатор, называемый IP – адресом. Адрес делится на четыре 8-ми разрядных значения, или октеты. Это значение обычно записывается через десятичные точки, таким образом, формат IP – адреса имеет вид: XXX.XXX.XXX.XXX. Например, 123.103.321.123.

Каждый IP – адрес состоит из двух частей: сетевого адреса и адреса главного компьютера сети (хост-компьютера). Старшие разряды первого октета имеют специальное значение: они определяют пять классов адресов сети. Например, сетевой адрес 128.1.2.3 имеет следующее двоичное представление: 10000000.00000001.00000010.00000011. В связи с тем, что старшие разряды (первые разряды первого октета, если читать слева направо) равны 10, данный адрес принадлежит к классу В. Классы IP – адресов приведены в таблице 1.

Таблица 1. Классы IP – адресов.

Класс	Первый октет (двоичное представление)	первый октет (десятичное представление)	Количество сетей	Количество хост-компьютеров на сеть.
Класс А	00000001-01111111	1 – 126	126	16 млн.
Класс В	10000000-10111111	128-191	16382	65534
Класс С	11000000-11011111	192 – 233	2 млн.	254
Класс D	11100000-11101111	224 – 239	Групповые	групповые
Класс E	11110000 – 11110111	240 – 247	Эксперим.	Эксперим.

В адресе класса А первый разряд равен 0, следующие семь идентифицируют сеть, а последние 24 идентифицируют Host – компьютер сети. В адресе класса В первые два разряда адреса равны 10, следующие 14 идентифицируют адрес сети, и оставшиеся 16 идентифицируют Host – компьютер. В адресе класса С первые три разряда равны 110, последующие 21 идентифицируют адрес сети, и оставшиеся 8 идентифицируют Host – ЭВМ.

Необязательно, чтобы некоторая сеть, например класса В, имела бы только одну локальную сеть, содержащую более 60 тыс. компьютеров. Такая сеть может иметь свое собственное межсетевое взаимодействие, которое способствует разветвлению локальной сети. Маска подсети разделяет сеть на меньшие сети, которые называются подсетями. С помощью маски подсети Internet поддерживает трехъярусную иерархию сеть/подсеть/Host- ЭВМ, а не двухъярусную модель сеть/Host – ЭВМ.

Маска подсети определяет раздел между разрядами адреса сети и разрядами адреса владельца. Подсеть известна только на локальном уровне, остальная часть Internet интерпретирует адрес стандартным способом. При подсоединении к Internet через локальную сеть важно использовать правильную маску подсети. Как и IP – адрес, маску подсети можно присвоить вручную или автоматически с помощью протокола DHCP (протокола динамической конфигурации Host – ЭВМ). В нижеприведенных таблицах показан эффект подсетей при разделении сетей классов В и С.

Подсети класса В.

Маска	Количество подсетей	Host – ЭВМ на сеть.
255.255.192.0	2	16382
255.255.224.0	6	8190
255.255.240.0	14	4094
255.255.248.0	30	2046
255.255.252.0	62	1022
255.255.254.0	126	510
255.255.255.0	254	254
255.255.255.128	510	126
255.255.255.192	1022	62
255.255.255.224	2046	30
255.255.255.240	4094	14
255.255.255.248	8190	6

Подсети класса С.

Маска	Количество подсетей	Host – ЭВМ на сеть.
255.255.255.192	2	62
255.255.255.224	4	30
255.255.255.240	14	14
255.255.255.248	30	6

В двоичном виде маска состоит из последовательности единиц, за которой следует последовательность нулей, а общая длина составляет 32 разряда. Затем эти последовательности маскируются с помощью поразрядной операции для разделения в сетевом адресе частей подсети и Host – ЭВМ.

Например, если сетевой адрес равен 172.20.0.0, а маска подсети равна 255.255.224.0, то сеть разбивается на шесть подсетей, как это показано в следующей таблице.

Маска подсети 255.255.224.0 делит класс сетей В на шесть подсетей.

Описание	Десятичное значение	Двоичное значение
Сетевой адрес	172.20.0.0	10101100.00001000.00000000.00000000
Маска подсети	255.255.224.0	11111111.11111111. 111 00000.00000000
Подсеть 1	172.20.32.0	10101100.00010100. 001 00000.00000000
Подсеть 2	172.20.64.0	10101100.00010100. 010 00000.00000000
Подсеть 3	172.20.96.0	10101100.00010100. 011 00000.00000000
Подсеть 4	172.20.128.0	10101100.00010100. 100 00000.00000000
Подсеть 5	172.20.160.0	10101100.00010100. 101 00000.00000000
Подсеть 6	172.20.192.0	10101100.00010100. 110 00000.00000000

В этой таблице часть номера, относящаяся к подсети, выделена жирным пунктиром. Любая Host – ЭВМ с IP – адресом от 172.20.32.1 (в двоичной форме 10101100.00010100.**001**00000.00000000) до 172.20.64.1 (в двоичной форме 10101100.00010100.**001**11111.11111111) является членом подсети 1, Host – ЭВМ с адресом от 172.20.64.1 до 172.20.95.254 является членом подсети 2 и т. д.

В операционной системе Windows NT значения адресов, масок, имена узлов и другая информация о локальной сети (параметры сетевого окружения) хранятся в системном реестре. Реестр представляет собой защищенную унифицированную базу данных для хранения конфигурационной информации. Доступ к реестру можно получить с помощью

редактора реестра RegEdit.exe, или с помощью функций API для работы с реестром. Описание функций работы с реестром Windows можно найти в справочной системе соответствующей интегрированной среды разработки, например, MSDN.

Реестр организован как набор 5 поддеревьев, которые состоят из ключей, содержащих базу данных по компьютеру и пользователям. Информация по компьютеру включает данные об аппаратных средствах и программном обеспечении, установленном на компьютере. Информация о пользователях включает данные о профиле каждого пользователя: настройку параметров рабочего стола, индивидуальную настройку некоторых видов программного обеспечения, а также персональные настройки для сети и принтеров. Отдельный ключ может содержать значимые элементы (value entries), а также дополнительные подключи (subkeys). Ключи в структуре реестра аналогичны каталогам, а значимые элементы файлам. Каждое из имен корневых поддеревьев начинается со строки HKEY_ - это дескриптор (handle) - значение, используемое для уникального описания реестра, к которому программа может получить доступ. Ниже приведено описание поддеревьев реестра.

Название ключа	Описание
HKEY_LOCAL_MACHINE	Информация о локальном компьютере (данные об аппаратных средствах и ОС – тип шины, системная Память, драйверы устройств и управляющие данные, используемые при запуске компьютера)
HKEY_CLASSES_ROOT	Содержит ассоциации между приложениями и типами файлов (по расширениям имен). Кроме того, содержит информацию OLE, ассоциированную с объектами COM. Значимые элементы этого поддерева совпадают со значимыми элементами, расположенными под ключом HKEY_LOCAL_MACHINE\Software\Classes
HKEY_CURRENT_CONFIG	Содержит конфигурацию для текущего аппаратного профиля (Hardware profiles) – изменения, внесенные в стандартную конфигурацию, установленную данными ключей Software и System поддерева HKEY_LOCAL_MACHINE . В поддереве HKEY_CURRENT_CONFIG находятся только изменения. Кроме того значимые элементы этого поддерева появляются под ключом HKEY_LOCAL_MACHINE\System\CurrentControlSet\HardwareProfiles\Current
HKEY_CURRENT_USER	Содержит пользовательский профиль пользователя, на данный момент зарегистрировавшегося с системе, включая переменные окружения, настройку рабочего стола, параметры настройки сети, принтеров и приложений.
HKEY_USERS	Содержит все активно загруженные пользовательские профили, включая HKEY_CURRENT_USER , а также профиль по умолчанию. Пользователи, получившие удаленный доступ к серверу, не имеют профилей, содержащихся под этим ключом, их профили загружаются в в реестры на их собственных компьютерах.

Значимый элемент реестра состоит из 3 частей: имени, определения типа данных и значения, которые могут представлять данные любой длины. Размер значимого элемента не может превышать 1 Мб. Значения от 0 до 0x7fffff зарезервированы для системных определений, а от 0x800000 до 0xfffff зарезервированы за приложениями.

Ниже в таблице перечислены типы данных, используемых в системе.

Тип данных	Описание
REG_BINARY	Не преобразованные двоичные данные. Редактор реестра отображает эту информацию в шестнадцатеричном формате, а программа Windows NT Diagnostics (Winmsdp.exe) в формате, удобном для чтения
REG_DWORD	Данные представлены 4 байтами. Этот тип данных используют многие драйверы. Отображаются в двоичном, шестнадцатеричном и десятичном формате.
REG_EXPAND_SZ	Расширяемая строка данных. Представляет собой текст, содержащий переменную (к примеру %Systemroot% - переменная, которая будет заменена фактическим именем пути к каталогу).
REG_MULTI_SZ	Многострочное поле. Значения, представленные списком текстовых строк, обычно имеют этот тип данных.
REG_SZ	Последовательность символов, представляющая собой текстовую строку. Этот тип данных обычно имеет значения, представляющие собой описания компонентов.

Для работы с реестром можно использовать следующие функции:

RegCreateKey (Key:HKKey; SubKey: PChar; var Result: HKKey): Longint;
Создать подраздел в реестре. Key указывает на "корневой" раздел реестра. SubKey - имя раздела - строится по принципу пути к файлу в DOS (пример subkey1\subkey2\ ...). Если такой раздел уже существует, то он открывается (в любом случае при успешном вызове Result содержит Handle на раздел). Об успешности вызова судят по возвращаемому значению, если ERROR_SUCCESS, то успешно, если иное - ошибка.

RegOpenKey(Key: HKKey; SubKey: PChar; var Result: HKKey):Longint;
Открыть подраздел Key\SubKey и возвращает Handle на него в переменной Result. Если раздела с таким именем нет, то он не создается. Возврат - код ошибки или ERROR_SUCCESS, если успешно.

RegCloseKey(Key: HKKey): Longint;
Закрывает раздел, на который ссылается Key. Возврат - код ошибки или ERROR_SUCCESS, если успешно.

RegDeleteKey(Key: HKKey; SubKey: PChar): Longint;
Удалить подраздел Key\SubKey. Возврат - код ошибки или ERROR_SUCCESS, если нет ошибок.

RegEnumKey(Key:HKKey; index: Longint; Buffer: PChar;cb: Longint):Longint;
Получить имена всех подразделов раздела Key, где Key - Handle на открытый или созданный раздел (см. RegCreateKey и RegOpenKey), Buffer - указатель на буфер, cb - размер буфера, index - индекс, должен быть равен 0 при первом вызове RegEnumKey. Типичное использование - в цикле While, где index увеличивается до тех пор, пока очередной вызов RegEnumKey не завершится

ошибкой

(см.

пример).

RegQueryValue(Key: HKEY; SubKey: PChar; Value: PChar; var cb: Longint): Longint;
Возвращает текстовую строку, связанную с ключом Key\SubKey.Value - буфер для строки; cb-размер, на входе - размер буфера, на выходе - длина возвращаемой строки. Возврат - код ошибки.

RegSetValue(Key: HKEY; SubKey: PChar; ValType: Longint; Value: PChar; cb: Longint): Longint;

Задать новое значение ключу Key\SubKey, ValType - тип задаваемой переменной, Value - буфер для переменной, cb - размер буфера. В Windows 3.1 допустимо только Value=REG_SZ. Возврат - код ошибки или ERROR_SUCCESS, если нет ошибок.

RegDeleteValue(HKEY hKey, LPCTSTR lpValueName);

Удаляет значение lpValueName находящееся в ключе hKey.hKey должен был быть открыт с доступом KEY_SET_VALUE процедурой RegOpenKey.Возвращает ERROR_SUCCESS если успешно.

LONG RegEnumValue(HKEY hKey, DWORD dwIndex, LPTSTR lpValueName, LPDWORD lpcbValueName, LPDWORD lpReserved, LPDWORD lpType, LPBYTE lpData, LPDWORD lpcbData);

, Выдает список значений у ключа hKey, dwIndex-этот параметр должен быть 0 при первом вызове, а далее по аналогии с RegEnumKey(т.е. можно использовать в цикле), lpValueName-буфер для названия значения, lpcbValueName-размер lpValueName, lpReserved должно быть всегда 0, lpType-буфер для названия типа (int), lpData- буфер для данных, lpcbData-размер для lpData.

Примечание: При каждой новом вызове функции после предыдущего нужно заново переназначить lpcbValueName. (lpcbValueName=sizeof lpValueName)

Пример :

```
// открытие раздела ... \NetworkCards
lError = RegOpenKey(HKEY_LOCAL_MACHINE, "Software\\Microsoft\\"
                  "Windows NT\\CurrentVersion\\NetworkCards", &hKey);
if( lError != ERROR_SUCCESS)
{
    MessageBox(NULL, "Ошибка при открытии раздела "
                "Software\\Microsoft\\Windows NT\\"
                "CurrentVersion\\NetworkCards",
                "ошибка", MB_OK|MB_ICONEXCLAMATION);
    return;
}

dwIndCards = 0;
// поиск подразделов NetworkCards
while(RegEnumKey(hKey, dwIndCards, szSubName, MAX_PATH+1) ==
ERROR_SUCCESS)
{
    // открытие подраздела NetworkCards
    lError = RegOpenKey(hKey, szSubName, &hSubKey);
    if( lError != ERROR_SUCCESS )
    {
        MessageBox(NULL, "Ошибка при открытии подраздела раздела NetworkCards",
                    "ошибка", MB_OK|MB_ICONEXCLAMATION);
        return;
    }
    // создание экземпляра класса Card
    theCard[dwIndCards] = new Card();
    theCard[dwIndCards]->SubName = szSubName;
    dwSizeSubName = sizeof(szSubName);
    dwSizeBuffer = sizeof(szBuffer);
    dwIndValues = 0;
    // чтение значения ключа реестра
```

```

while (RegEnumValue (hSubKey, dwIndValues, szSubName, &dwSizeSubName, NULL,
                    &dwType, (LPBYTE) szBuffer, &dwSizeBuffer) == ERROR_SUCCESS
)
{
    // имя сетевого сервиса
    if (!strcmp (szSubName, "ServiceName"))
        theCard [dwIndCards] -> ServiceName = szBuffer;
    // описание сетевой карты
    if (!strcmp (szSubName, "Description"))
        theCard [dwIndCards] -> Description = szBuffer;

    dwSizeSubName = sizeof (szSubName);
    dwSizeBuffer = sizeof (szBuffer);
    dwIndValues++;
}
// закрытие подраздела раздела NetworkCards
RegCloseKey (hSubKey);
if (++dwIndCards == 5) break;
}
// закрытие раздела ... \NetworkCards
RegCloseKey (hKey);
}

```

Ключи реестра, необходимые для выполнения работы:

Имя адаптера:

\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Network Cards\1

\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Network Cards\2

.....

ключ - ServiceName - строка содержит имя адаптера для дальнейшего поиска.

Параметры TCP/IP

\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\<AdapterName>\Parameters\Tcpip

Ключи:

EnableDhcp

Тип данных: REG_DWORD - Boolean

Возможные значения: 0 or 1 (False or True)

Значение по умолчанию: 0 (False)

Описание: если этот параметр равен 1 (True), то

параметры TCP/IP конфигурируются с использованием DHCP - сервера.

DefaultGateway

Тип данных: REG_MULTI_SZ - набор десятичных чисел (в кодировке) разделённых точками (IP addresses)

Возможные значения: Правильные ip адреса

Значение по умолчанию: нет

Описание: Параметр описывает адрес шлюза (шлюзов) по умолчанию, используемых для маршрутизации пакетов.

Hostname

Value Type: REG_SZ - строка

Valid Range: Правильное имя хоста

Default: Имя компьютера

Description: Этот параметр описывает сетевое имя системы, возвращаемое по команде hostname

IPAddress

Value Type: REG_MULTI_SZ - набор десятичных чисел (в кодировке) разделённых точками (IP addresses)

Valid Range: Правильные ip адреса

Default: Нет

Description: Этот параметр задаёт IP адрес, прикрепляемый к данному сетевому адаптеру.

Если он равен 0.0.0.0, то параметры TCP/IP конфигурируются с использованием DHCP-сервера

SubnetMask

Value Type: REG_MULTI_SZ - набор десятичных чисел (в кодировке) разделённых точками (IP addresses)

Valid Range: Правильные ip адреса

Default: Нет

Description: Этот параметр задаёт маску подсети, используемую вместе с IP адресом адаптера.

Если значение равно 0.0.0.0, то параметры TCP/IP конфигурируются с использованием DHCP-сервера

*** Примечание: В общем случае адаптеру может быть прикреплен не один IP адрес (logically multihomed), однако такой вариант рассматриваться не будет.

ПРИ ИСПОЛЬЗОВАНИИ DHCP-СЕРВЕРА

DhcpDefaultGateway

Value Type: REG_MULTI_SZ - набор десятичных чисел (в кодировке) разделённых точками (IP addresses)

Valid Range: Правильные ip адреса

Default: Нет

Description: Аналог DefaultGateway, сконфигурированный с помощью DHCP-СЕРВЕРА.

DhcpIPAddress

Value Type: REG_MULTI_SZ - набор десятичных чисел (в кодировке) разделённых точками (IP addresses)

Valid Range: Правильные ip адреса

Default: Нет

Description: Аналог IPAddress, сконфигурированный с помощью DHCP-СЕРВЕРА.

DhcpServer

Value Type: REG_MULTI_SZ - набор десятичных чисел (в кодировке) разделённых точками (IP addresses)

Valid Range: Правильные ip адреса

Default: Нет

Описание: Этот параметр задаёт адрес DHCP-Сервера.

DhcpSubnetMask

Value Type: REG_MULTI_SZ - набор десятичных чисел (в кодировке) разделённых точками (IP addresses)

Valid Range: Правильные ip адреса

Default: Нет

Description: Аналог SubnetMask, сконфигурированный с помощью DHCP-СЕРВЕРА.

Порядок выполнения работы.

Для выполнения лабораторной работы необходимо:

1. Изучить формат IP – адресов и масок подсети.
2. Изучить структуру системного реестра с помощью утилиты RegEdit.
3. Изучить правила обращения к функциям доступа к системному реестру.
4. Разработать приложение, извлекающее из системного реестра IP – адрес, имя локального узла, маску подсети.
5. Исходя из полученной информации, определить класс полученного адреса и структуру локальной сети.

Содержание отчета.

Отчет по лабораторной работе должен включать в себя:

1. IP – адрес локального узла и маску подсети в десятичном и двоичном представлениях с указанием класса подсети;
2. Имена найденных сетевых адаптеров и параметры их протоколов;
3. Блок – схему алгоритма определения параметров сетевого окружения;
4. Краткий листинг разработанного приложения с комментариями (В части реализации разработанного алгоритма).

Литература.

1. Семенов Ю. А. «Протоколы и ресурсы Internet». М.: «Радио и Связь», 1997 г.
- 2.