Надо сделать: Получить оценку

Цель работы: Научиться применять сетевые утилиты командной строки Windows.

Теория

Утилитами называются сравнительно небольшие программы, предназначенные для решения каких-либо узкоспециализированных задач. В данной работе рассматриваются утилиты операционной системы Windows, используемые для диагностики сетевых подключений. Обзор утилит мы совместим с изучением основ теории компьютерных сетей.

Утилита ipconfig

Для связи с сетью компьютеры оснащаются сетевыми интерфейсами, к которым относятся Ethernet платы, Wi-Fi и WiMAX модули. Указанные интерфейсы должны иметь IP адреса. Пример такого адреса – **192.168.0.1**. Компьютер может иметь не одну плату, а две или три, и каждая из них будет иметь свой IP адрес. Если имеется WiMAX модуль, то и он будет иметь свой IP адрес. Таким образом, компьютер может иметь несколько адресов. Адреса необходимы для организации пересылки сообщений по сети. Адреса должны быть уникальными, т.е. неповторяющимися. Ведь если в сети находятся два компьютера с одинаковыми адресами, то кому из них будет адресовано сообщение с указанным адресом? Отметим также, что IP адреса разбиты на две категории: приватные и публичные. Приватные адреса имеют силу лишь для своей локальной сети и в глобальной сети они не видны. Примером такого адреса является **192.168.0.1**. Существуют сотни тысяч, а может быть, миллионы локальных сетей, в которых встречаются компьютеры с одинаковыми приватными адресами, и они никак не конфликтуют между собой из-за совпадения адресов. Публичные же адреса уникальны для всей глобальной сети.

Компьютеры образуют сети, которые также имеют свои адреса. Например, компьютер с адресом **192.168.0.1** находится в сети с адресом **192.168.0.0**. У адреса сети и адреса компьютера, как видим, совпадают первые три числа. Сколько же на самом деле должно совпадать чисел определяет так называемая маска подсети. Для нашего примера эта маска имеет вид **255.255.255.0**. Такое значение маски чаще всего и встречается в локальных сетях. Более подробно об IP адресации и, соответственно, о маске подсети будет изложено в теоретической части данного курса.

Сами компьютерные сети не изолированы друг от друга. Для связи их между собой используются специализированные компьютеры, называемые маршрутизаторами. Такие сетевые устройства имеют как минимум два сетевых интерфейса, один из которых принадлежит одной сети, другой же является частью второй сети. Маршрутизатор, перенаправляя сообщения с одного своего интерфейса на другой, обеспечивает межсетевой трафик. Если маршрутизатор имеет три платы, то он будет находиться на границе трех сетей. Широкое распространение получили двухточечные сети, которые образуют два маршрутизатора, соединенные общим кабелем. Интерфейсы обеих маршрутизаторов, присоединенные к разным концам одного кабеля, должны иметь адреса, относящиеся к одной и той же сети. Более часто встречаются тупиковые сети. Такие сети связаны лишь с одним маршрутизатором (отсюда и название сети - тупиковая).

Компьютеры, находящиеся в такой сети, отправляют сообщения, адресованные в другие сети, на интерфейс этого маршрутизатора. Компьютеры, следовательно, должны знать адрес интерфейса маршрутизатора своей сети. Такой адрес носит название «основной шлюз». Маршрутизатор полученные от компьютеров тупиковой сети сообщения перенаправляет дальше, передавая их своим соседям-маршрутизаторам по двухточечным каналам связи. Таким образом, сообщение последовательно перемещается по следующим сетям: тупиковая сеть, двухточечная сеть 1, двухточечная сеть 2, ..., двухточечная сеть N, тупиковая сеть. Если же маршрут перемещения изучать по узлам, то он будет таким: компьютер (отправитель сообщения), маршрутизатор 1, маршрутизатор 2, …, маршрутизатор N-1, компьютер (получатель сообщения).

Подытоживая вышесказанное, отметим, что таким образом для настройки сетевого интерфейса компьютера необходимо назначить ему IP адрес, маску подсети и основной шлюз.

Программа **ipconfig** предназначена для получения информации о настройках сетевых интерфейсов. Выполняется данная утилита в окне командной строки. Для этого необходимо нажать кнопку **Пуск** и выбрать пункт «**Выполнить…**»4. Далее следует ввести **cmd** и нажать **Enter**. В открывшемся окне командной строки (рис. 1.1) следует ввести команду **ipconfig** и нажать **Enter**.

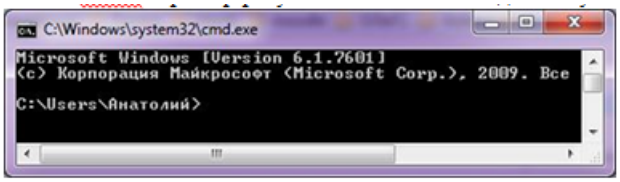

Рис.1.1. Окно командной строки

```
C:\Users\Анатолий>ipconfig
Настройка протокола IP для Windows
Ethernet adapter Сетевое подключение Bluetooth:
Состояние среды. . . . . : Среда передачи недоступна. DNS-суффикс подключения . . . . . :
Адаптер беспроводной локальной сети Беспроводное сетевое соединение:
DNS-суффикс подключения . . . . . :Home
Локальный IPv6-адрес канала . . . : fe80::595:5d8d:56f5:500c%41 IPv4-адрес. . . . . :
192.168.0.206
Маска подсети . . . . . : 255.255.255.0
Основной шлюз. . . . . : 192.168.0.10
Ethernet adapter Подключение по локальной сети:
Состояние среды. . . . . : Среда передачи недоступна. DNS-суффикс подключения . . . . . :
Туннельный адаптер isatap.{DDDBA9F8-664B-4B57-B2D0-93DE69D2FBE7}:
Состояние среды. . . . . : Среда передачи недоступна. DNS-суффикс подключения . . . . . :
Туннельный адаптер Подключение по локальной сети* 3:
Состояние среды. . . . . : Среда передачи недоступна. DNS-суффикс подключения . . . . . :
Туннельный адаптер isatap.{73A36F53-6EE2-4F00-B90B-D11719558242}:
Состояние среды. . . . . : Среда передачи недоступна. DNS-суффикс подключения . . . . . :
Туннельный адаптер isatap.Home:
Состояние среды. . . . . : Среда передачи недоступна. DNS-суффикс подключения . . . . . :Home
```

В данном примере команда выполнялась на ноутбуке. Связь с «внешним миром» осуществлена с помощью беспроводной сети. IPадрес интерфейса[5] ноутбука -

192.168.0.206, маска - **255.255.255.0**, шлюз[6] - **192.168.0.10**. Интерфейсы ноутбука и маршрутизатора находятся в одной сети **192.168.0.0**. Сетевой кабель не подключен (по Ethernet адаптеру среда передачи недоступна).

Утилита ping

Компьютеры и другие узлы сети помимо IP адресов имеют так называемые доменные адреса. Такие адреса удобны пользователям сети, так как они легче запоминаются. К примеру доменный адрес **mail.ru** запомнить намного проще чем его IP аналог в виде **94.100.180.70**. За соответствие доменных и IP адресов отвечает DNS[7] служба. Когда с компьютера исходит запрос на какой-либо сетевой ресурс по его доменному адресу, то DNS служба позволяет определить соответствующий этому ресурсу IP адрес.

Утилита **ping** позволяет проверить доступность какого-либо удаленного узла по сети. С этой целью на указанный узел отправляется сообщение в виде запроса, и утилита переходит в режим ожидания прихода ответного сообщения. По истечении некоторого времени посылается повторное сообщение. По результатам обмена сообщениями выводится статистика о качестве связи между двумя узлами. Для пингования удаленного узла можно использовать либо его IP адрес, либо его доменное имя.

Команда **ping 127.0.0.1** позволяет проверить настройку самого сетевого интерфейса. Адрес 127.0.0.1 является служебным и узлам сети не назначается. Сетевой интерфейс при использовании данного адреса пингует сам себя. Доменное имя адреса 127.0.0.1 – localhost.

```
ping 127.0.0.1
Обмен пакетами с 127.0.0.1 по с 32 байтами данных:
Ответ от 127.0.0.1: число байт=32 время<1мс TTL=128
Ответ от 127.0.0.1: число байт=32 время<1мс TTL=128
Ответ от 127.0.0.1: число байт=32 время<1мс TTL=128 Ответ от 127.0.0.1: число байт=32 время<1мс TTL=128
Статистика Ping для 127.0.0.1:
Пакетов: отправлено = 4, получено = 4, потеряно = 0 (0% потерь)
Приблизительное время приема-передачи в мс:
Минимальное = 0мсек, Максимальное = 0 мсек, Среднее = 0 мсек
```

В рассмотренном случае сетевой интерфейс настроен без ошибок, потери отсутствуют. Параметр TTL переводится как «время жизни» (time to life). Его создает узел, отправляющий в сеть свое сообщение. Маршрутизаторы, передавая данное сообщение из одной сети в другую, убавляют TTL на единицу. Если на каком-то маршрутизаторе TTL будет убавлено до нуля, то сообщение будет уничтожено. Маршрутизатор, удаливший из сети сообщение, извещает об этом отправителя, указывая свой адрес.

Второй вариант использования **ping** – это проверка состояния тупиковой сети, в которой находится сам узел. С этой целью пингуется основной шлюз:

```
ping 192.168.0.10
Обмен пакетами с 192.168.0.10 по с 32 байтами данных:
Ответ от 192.168.0.10: число байт=32 время=11мс TTL=64 Ответ от 192.168.0.10: число байт=32 время=10мс TTL=64 Ответ от 192.168.0.10: число байт=32 время=9мс TTL=64 Ответ от 192.168.0.10: число байт=32 время=8мс TTL=64
Статистика Ping для 192.168.0.10:
Пакетов: отправлено = 4, получено = 4, потеряно = 0 (0% потерь)
Приблизительное время приема-передачи в мс:
Минимальное = 8мсек, Максимальное = 11 мсек, Среднее = 9 мсек
```

В данном примере маршрутизатор доступен. Он в свои ответные сообщения помещает TTL (64) отличное от TTL (128) сетевого интерфейса компьютера.

Для проверки доступности удаленного хоста, как правило, применяются доменные адреса:

```
ping esstu.ru
Обмен пакетами с esstu.ru [212.0.68.2] с 32 байтами данных: Ответ от 212.0.68.2: число байт=32 время=7мс TTL=57

Ответ от 212.0.68.2: число байт=32 время=8мс TTL=57 Ответ от 212.0.68.2: число байт=32 время=10мс TTL=57 Ответ от 212.0.68.2: число байт=32 время=7мс TTL=57
Статистика Ping для 212.0.68.2:
Пакетов: отправлено = 4, получено = 4, потеряно = 0 (0% потерь)
Приблизительное время приема-передачи в мс:
Минимальное = 7мсек, Максимальное = 10 мсек, Среднее = 8 мсек
```

Удаленный узел доступен. В данном случае мы видим, что DNS служба определила IP адрес узла в виде **212.0.68.2**.

Утилита имеет несколько опций, из которых рассмотрим лишь одну: **-i**, позволяющую задать значение TTL:

```
Утилита имеет несколько опций, из которых рассмотрим лишь одну: -i, позволяющую задать значение TTL:

ping -i 1 esstu.ru
Обмен пакетами с esstu.ru [212.0.68.2] с 32 байтами данных:
Ответ от 192.168.0.10: Превышен срок жизни (TTL) при передаче пакета.
Ответ от 192.168.0.10: Превышен срок жизни (TTL) при передаче пакета.
Ответ от 192.168.0.10: Превышен срок жизни (TTL) при передаче пакета.
Ответ от 192.168.0.10: Превышен срок жизни (TTL) при передаче пакета.
Статистика Ping для 212.0.68.2:
Пакетов: отправлено = 4, получено = 4, потеряно = 0 (0% потерь)
```

Здесь TTL был принят равным 1 и сообщение было уничтожено на шлюзе (**192.168.0.10**). В следующем примере TTL=2

```
ping -i 2 esstu.ru
Обмен пакетами с esstu.ru [212.0.68.2] с 32 байтами данных:
Ответ от 192.168.1.1: Превышен срок жизни (TTL) при передаче пакета.
Ответ от 192.168.1.1: Превышен срок жизни (TTL) при передаче пакета.
Ответ от 192.168.1.1: Превышен срок жизни (TTL) при передаче пакета.
Ответ от 192.168.1.1: Превышен срок жизни (TTL) при передаче пакета.
Статистика Ping для 212.0.68.2:
Пакетов: отправлено = 4, получено = 4, потеряно = 0 (0% потерь)
```

На этот раз сообщение дошло до маршрутизатора (**192.168.1.1**), который был вторым за шлюзом. Так, постепенно меняя значение TTL, можно получить список всех маршрутизаторов, находящихся между компьютером и удаленным узлом **212.0.68.2**.

Утилита tracert

Эта утилита, последовательно применяя пинг с увеличивающимся TTL, позволяет получить список промежуточных маршрутизаторов:

```
tracert esstu.ru
Трассировка маршрута к esstu.ru [212.0.68.2] с максимальным числом прыжков 30:
1   1    ms    1    ms    1    ms  msWRT54GL [192.168.0.10]
2   3    ms    2    ms    2    ms  192.168.1.1
3   8    ms    4    ms    5    ms  ULND-BRAS3.sib.ip.rostelecom.ru
[213.228.116.203]
4   17   ms    5    ms    8    ms  213.228.114.27
5   8    ms    3    ms    3    ms  core-gi-0-2.burnet.ru [212.0.64.90]
6   7    ms    5    ms    6    ms  ws-70-71.burnet.ru [212.0.70.71]
7   8    ms    4    ms    3    ms  86.110.127.129
8   8    ms    4    ms    4    ms  212.0.68.2
Трассировка завершена.
```

Между двумя узлами в данном случае находится 7 маршрутизаторов.

Утилита pathping

```
Утилита pathping сочетает в себе черты команд ping и tracert, позволяя получить
дополнительную информацию, которую не обеспечивают две последние. Команда определяет
процент потерь сообщений на всех переходах, выявляя самые медленные и ненадежные
участки маршрута.

pathping esstu.ru
Трассировка маршрута к esstu.ru [212.0.68.2] с максимальным числом прыжков 30:
EXPHOME.Home [192.168.0.206]
WRT54GL [192.168.0.10]
Broadcom.Home [192.168.1.1]
ULND-BRAS3.sib.ip.rostelecom.ru [213.228.116.203]
213.228.114.27
core-gi-0-2.burnet.ru [212.0.64.90]
ws-70-71.burnet.ru [212.0.70.71]
86.110.127.129
212.0.68.2
```

```
Подсчет статистики за: 200 сек. ...
            Исходный узел         Маршрутный узел
Прыжок  RTT      Утер./Отпр.      %   Утер./Отпр.     %   Адрес
  0                                                     EXPHOME.Home
[192.168.0.206]
                                  0/ 100 =       0%   |
  1     6мс      0/ 100 =     0%   0/ 100 =       0%   WRT54GL [192.168.0.10]
                                  0/ 100 =       0%   |
  2     8мс      0/ 100 =     0%   0/ 100 =       0%   Broadcom.Home
[192.168.1.1]
                                  0/ 100 =       0%   |
3 11мс           0/ 100 =     0%   0/ 100 =       0%   ULND-
BRAS3.sib.ip.rostelecom.ru [213.228.116.203]
0/ 100 =     0%  |
  4     11мс     0/ 100 =     0%   0/ 100 =       0%   213.228.114.27
                                  0/ 100 =       0%   |
  5     10мс     0/ 100 =     0%   0/ 100 =       0%   core-gi-0-2.burnet.ru
[212.0.64.90]
0/ 100 =     0%  |
  6     15мс     0/ 100 =     0%   0/ 100 =       0%   ws-70-71.burnet.ru
[212.0.70.71]
                                  0/ 100 =       0%   |
7 ---         100/ 100 =100%      100/ 100 =100%      86.110.127.129
                                  0/ 100 =       0%   |
  8     12мс     0/ 100 =     0%   0/ 100 =       0%   212.0.68.2
Трассировка завершена.
```

В настройках некоторых маршрутизаторов может стоять запрет на выдачу ответа на пришедший пинг. В данном примере маршрутизатор с подобной настройкой имеет адрес **86.110.127.129**. Попытка отправить пинг на этот адрес убеждает нас в справедливости этого утверждения:

```
ping 86.110.127.129
Обмен пакетами с 86.110.127.129 по с 32 байтами данных: Превышен интервал ожидания
для запроса.
Превышен интервал ожидания для запроса. Превышен интервал ожидания для запроса.
Превышен интервал ожидания для запроса.
Статистика Ping для 86.110.127.129:
Пакетов: отправлено = 4, получено = 0, потеряно = 4
(100% потерь)
```

Утилита arp

Сетевые интерфейсы, такие как Ethernet, Wi-Fi и WiMAX, имеют вшитые в их микросхемы адреса. Пример подобного адреса: 70-F3-95-A6-FE-0C. Эти адреса, называемые аппаратными, физическими или MAC, должны добавляться к сообщениям, прежде чем они будут переданы через сеть. Не все сети используют такие адреса, но в тупиковых они, как правило, применяются. Узел, собирающийся отправить сообщение другому узлу, должен предварительно узнать MAC адрес получателя сообщения. Для решения данной проблемы узел применяет технологию ARP, отправляя запрос другим узлам своей локальной сети. Данный ARP запрос содержит IP адрес получателя. Из всех узлов, получивших данный запрос, отвечает лишь тот, у кого требуемый IP адрес. В своем ответе (ARP отклике) тот узел сообщает свой MAC адрес. И лишь после этого первый узел ему сможет отправить свое сообщение. В тупиковых сетях компьютеры чаще всего отправляют свои сообщения маршрутизатору и, следовательно, в своих ARP запросах они указывают адрес основного шлюза. Для уменьшения ARP трафика компьютеры хранят в своей памяти таблицу с IP и MAC адресами тех устройств, с которыми они в последнее время обменивались сообщениями.

Утилита **arp** позволяет получить таблицу соответствия IP адресов и MAC-адресов. Ниже приведен вывод, полученный командой **arp**, выполненной с ключом **–a**

```
arp -a

Интерфейс: 192.168.0.206 --- 0xe
адрес в Интернете          Физический адрес          Тип
192.168.0.10               20-aa-4b-2a-d5-21         динамический
192.168.0.255              ff-ff-ff-ff-ff-ff         статический
224.0.0.22                 01-00-5e-00-00-16         статический
224.0.0.252                01-00-5e-00-00-fc         статический
239.192.152.143            01-00-5e-40-98-8f         статический
239.255.255.250            01-00-5e-7f-ff-fa         статический
255.255.255.255            ff-ff-ff-ff-ff-ff         статический
```

В данном случае мы видим, что основной шлюз (**192.168.0.10**) имеет MAC адрес **20-aa-4b-2a-d5-21**.

Утилита netstat

Когда мы говорим: «компьютеры обмениваются сообщениями», то это не совсем точное утверждение. На самом деле обмен происходит между сетевыми приложениями. В оперативной памяти компьютера одновременно могут находиться и выполняться несколько программ, получающих сообщения из сети или отправляющие их в сеть. Как же сообщения, приходящие из сети в компьютер, распределяются между этими приложениями? На этот случай в сообщениях предусмотрены дополнительные адреса, называемые *портами*. Здесь уместно привести аналогию с обычной почтовой корреспонденцией. Для того чтобы письмо было доставлено в многоквартирный дом (компьютер), на конверте письма указывается номер дома (IP адрес компьютера). Затем письма необходимо разложить по почтовым ящикам согласно номерам квартир. Номер квартиры, присутствующий на конверте письма, и есть аналог портов. Далее жильцы (т.е. сетевые приложения) забирают эти письма (сообщения).

Когда приложение хочет обменяться сообщениями с другим удаленным приложением, оно должно знать не только IP адрес компьютера данного приложения, но и номер порта, которое то приложение использует. Эта связка из двух адресов (IP адрес и порт) называется *сокетом*. Как определяется номер порта, которое использует удаленное приложение - эта тема отдельного разговора. Оба приложения устанавливают между собой соединение, используя два сокета. Сокеты можно условно представить в виде двух разъемов (розеток), соединенных между собой неким виртуальным каналом связи. Когда одно приложение «помещает» в сокет свое сообщение, то оно доставляется на другой конец канала - на второй сокет, и попадает, таким образом, другому приложению.

Команда **netstat** позволяет получить список сокетов. Ниже приведен вывод, полученный с использованием опций **a**, **n**,и **o**.

```
netstat -ano
Активные   подключения

Имя         Локальный адрес         Внешний адрес           Состояние       PID
TCP         0.0.0.0:80              0.0.0.0:0               LISTENING       2944
TCP         0.0.0.0:135             0.0.0.0:0               LISTENING       892
TCP         0.0.0.0:443             0.0.0.0:0               LISTENING       2944
TCP         0.0.0.0:445             0.0.0.0:0               LISTENING       4
TCP         0.0.0.0:2869            0.0.0.0:0               LISTENING       4
TCP         0.0.0.0:5357            0.0.0.0:0               LISTENING       4
TCP         0.0.0.0:26143           0.0.0.0:0               LISTENING       4
TCP         0.0.0.0:45662           0.0.0.0:0               LISTENING       2920
TCP         0.0.0.0:49152           0.0.0.0:0               LISTENING       576
TCP         0.0.0.0:49153           0.0.0.0:0               LISTENING       1020
TCP         0.0.0.0:49154           0.0.0.0:0               LISTENING       724
TCP         0.0.0.0:49155           0.0.0.0:0               LISTENING       660
TCP         0.0.0.0:49160           0.0.0.0:0               LISTENING       640
TCP         0.0.0.0:61741           0.0.0.0:0               LISTENING       2944
TCP         127.0.0.1:5939          0.0.0.0:0               LISTENING       2580
TCP         127.0.0.1:10000         0.0.0.0:0               LISTENING       2920
TCP         127.0.0.1:49156         127.0.0.1:49157         ESTABLISHED     2952
TCP         127.0.0.1:49157         127.0.0.1:49156         ESTABLISHED     2952
TCP         127.0.0.1:49158         127.0.0.1:49159         ESTABLISHED     2952
TCP         127.0.0.1:49159         127.0.0.1:49158         ESTABLISHED     2952
TCP         192.168.0.206:139       0.0.0.0:0               LISTENING       4
TCP         192.168.0.206:54842     64.4.23.171:40013       ESTABLISHED     2944
TCP         192.168.0.206:54844     157.56.53.42:12350      ESTABLISHED     2944
TCP         192.168.0.206:54845     173.252.121.3:5222      ESTABLISHED     2944
TCP         192.168.0.206:54850     191.235.188.99:443      ESTABLISHED     2944
TCP         192.168.0.206:54893     64.4.61.132:443         ESTABLISHED     2944
TCP         192.168.0.206:54919     157.56.194.7:443        ESTABLISHED     2944
TCP         192.168.0.206:55125     185.39.80.24:80         ESTABLISHED     3684
TCP         192.168.0.206:57762     137.116.224.167:443     TIME_WAIT       0
TCP         192.168.0.206:57770     81.19.104.81:443        TIME_WAIT       0
TCP         192.168.0.206:57792     192.168.0.10:1780       TIME_WAIT       0
TCP         192.168.0.206:57793     192.168.0.10:1780       TIME_WAIT       0
TCP         192.168.0.206:57822     81.19.104.81:443        TIME_WAIT       0
TCP         192.168.0.206:57845     176.119.71.119:62348    ESTABLISHED     2920
            <дальнейший вывод был пропущен>
```

Данный вывод показывает, что сокеты обозначаются в виде пары **IP_адрес: порт** (с двоеточием между адресами). Например, **192.168.0.206:54842**. Виртуальный канал связи, существующий между двумя сетевыми приложениями, обозначен парой сокетов.

Например, **192.168.0.206:54842** и **64.4.23.171:40013**. Первый сокет открыт на компьютере, другой на удаленном узле. Адрес в виде **0.0.0.0** означает любые IP адреса. Если в качестве номера порта присутствует **0**, то это означает любые значения портов. В колонке "Состояние" отображается состояние соединения:

LISTENING – ожидание подключения;

ESTABLISHED– соединение установлено, идет обмен сообщениями;

TIME_WAIT–время ответа превышено.

Первый тип состояния (LISTENING) означает, что сетевое приложение ждет установления соединения по определенному порту. Например, сокет **0.0.0.0:443** означает, что какое-то удаленное приложение может отправить на компьютер сообщение на порт 443 с целью установить виртуальное соединение.

В последней колонке (PID) выводятся номера процессов. Под процессами понимаются приложения. Из вывода мы видим, что процесс 2944 ждет подключения по портам 80, 443 и 61741. Как выше было сказано, какая-то программа с другого узла может отправить запрос на установление соединения с процессом 2944. Такая программа своѐ сообщение может адресовать на любой из указанных трех портов. Чтобы выяснить, какая программа запущена под видом процесса 2944, вызовем диспетчер задач (Ctrl+Alt+Delete). В окне диспетчера перейдем на вкладку **Процессы** и войдем в меню **Вид**. Далее выберем строчку **Выбрать столбцы** и активируем чекбокс **ИД процесса (PID)**. Щелкнем по **OK**. Затем отсортируем таблицу по столбцу **ИД процесса (PID)**, щелкнув по его названию. Находим запись, соответствующую процессу 2944.

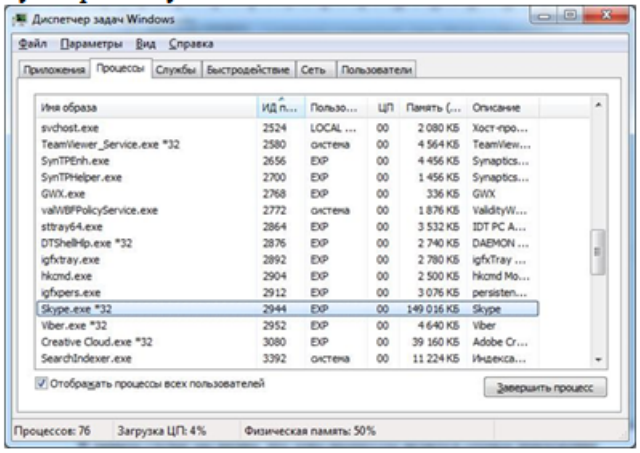

Рис. 1.2. Диспетчер задач Windows

В данном случае мы видим, что этим процессом является сетевое приложение Skype. Приведенный выше вывод **netstat** показывает, что данная программа поддерживает связь с шестью удаленными skype приложениями других пользователей. Для каждого соединения был создан отдельный сокет.

Подготовительная часть

Все команды выполняются в окне командной строки и дублируются с помощью пакетного файла. Правила запуска окна командной строки были рассмотрены выше (рис. 1.1). Отметим лишь, что команды в окне вводятся с клавиатуры и выполняются нажатием на клавишу **Enter**. Предыдущие команды можно вызвать для редактирования и повторного выполнения с помощью клавиши **Стрелка вверх**. Скриншоты окна помещайте в файл отчета со всеми необходимыми комментариями.

Рассмотрим приемы работы с пакетным файлом. Для этого Вам потребуется специализированный текстовый редактор PSPad. С помощью поисковых машин Интернета (Google, Яндекс и т.п.) найдите сайт разработчика данного редактора. Скачайте дистрибутив редактора и инсталлируйте его на своем компьютере. Также рекомендуется с данного сайта скачать русификатор, который следует распаковать в папку Lang.

Запустите PSPad и в его окне выполните команду **Файл/Новый**. Выберите тип файла **MS-DOS Batch**. В созданный файл введите строку текста ipconfig > 1.txt .

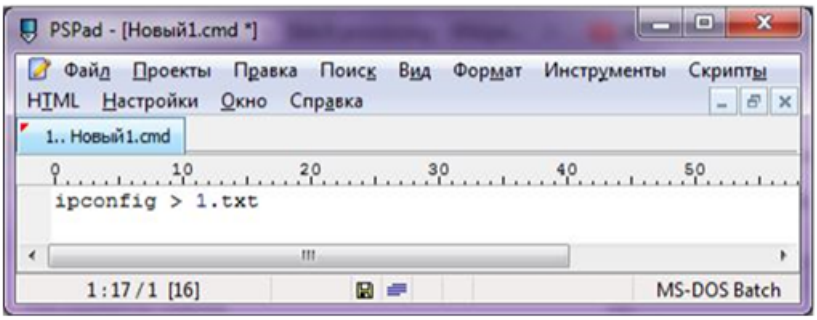

Рис. 1.3. PSPad

Сохраните файл на рабочий стол под именем **script**. На рабочем столе файл имеет вид

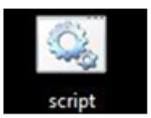

Рис. 1.4. Иконка скрипта

Для его выполнения выполните двойной щелчок по нему мышью. На рабочем столе появится новый файл в виде

Рис. 1.4. Созданный файл

Для открытия файла **1.txt** следует вернуться в окно PSPad. В меню **Формат** PSPad'а выберите **OEM** и затем откройте файл **1.txt** (**Файл/Открыть…**). Важно соблюдать порядок действий: вначале выбирать **OEM** и после открывать файл. Из данного файла некоторые фрагменты вывода можно скопировать в файл отчета по лабораторной работе.

Поменяйте строку в скрипте на ipconfig /all > 2.txt

Красный уголок у названия скрипта предупреждает, что файл был изменен, но не сохранен. Сохраните файл. Уголок должен исчезнуть. При выполнении команды ipconfig /all вывод будет помещен в текстовый файл **2.txt**. Запустите с рабочего стола скрипт и просмотрите в PSPad'е содержимое файла **2.txt**.

Порядок выполнения задания

Создайте в Word'е файл отчета. Образец титульного листа приведен в приложении.

В отчете создайте заголовки тем:

«Утилита ipconfig»

«Утилита ping»

«Утилита tracert»

«Утилита pathping»

«Утилита arp»

«Утилита netstat»

В окне командной строки выполните команду **ipconfig**. Скрин окна (**Alt+Print Scr**) поместите в отчет (**Ctrl+V**). Вывод команды **ipconfig** перенаправьте в текстовый файл, используя пакетный файл. Запишите в отчет информацию об IP адресе сетевого адаптера, маске сети и шлюзе по умолчанию.

Для получения более подробной информации о настройках адаптера запустите в окне командной строки утилиту **ipconfig** с ключом **/all**. Скрин окна поместите в отчет.

Повторите команду **ipconfig /all** с выводом в текстовый файл и запишите в отчет информацию о физическом адресе сетевой платы.

Применив команду **ping**, проверьте настройку платы, доступность основного шлюза и доступность удаленного узла. Адреса удаленных узлов выбирайте по своему варианту (см. **Варианты**). Скрины и выводы поместите в отчет.

Используя опцию **-i** команды **ping** определите адреса первых трех маршрутизаторов находящихся между вашим компьютером и удаленным узлом.

Применив команду **tracert**, получите список роутеров на маршруте от вашего компьютера до удаленного узла. Адреса и скрины поместите в отчет.

Используя **pathping**, изучите состояние линков на маршруте от вашего компьютера до удаленного узла и определите самые «узкие места» (т.е. самые медленные участки).

Получите таблицу ARP вашего компьютера. Выпишите в отчет MAC адрес основного шлюза.

Командой **netstat**, выполненной с ключами **–a**, **–n** и **–o**, получите список соединений, действующих на Вашем компьютере.

Определите имя любого приложения, установившего соединение с удаленной программой. Свой вывод обоснуйте соответствующим скрином.

Варианты

infpol.ru

vk.com

yandex.ru

ok.ru

mail.ru

rambler.ru

wikipedia.org

Web-ресурсы

Работа с сетью в командной строке Windows (http://ab57.ru/netcmd.html)

Сетевые утилиты. Нужные программы в составе Windows

(http://itboombox.ru/training/item/71-setevye-utility-nuzhnye-programmy-v-sostave-windows.html)

Добавить ответ на задание

Состояние ответа

Состояние ответа на задание	Ответы на задание еще не представлены
Состояние оценивания	Не оценено
Последнее изменение	-
Комментарии к ответу	▸ Комментарии (0)

Контакты

308503, Белгородская обл., Белгородский р-н, п. Майский, ул. Вавилова, 1, отдел электронных образовательных ресурсов и сетевого обучения, №321 (с 8.00 до 17.00, перерыв 12.00-13.00)

Телефон : +7 (4722) 39-22-51 (по вопросам ЭИОС). По вопросам справок: +7 (4722) 38-05-17 (МФЦ БелГАУ)

Эл.почта : help@belgau.ru