



ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ ГОРОДА  
МОСКВЫ

Государственное автономное профессиональное  
образовательное учреждение города Москвы  
**ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ № 8**  
имени дважды Героя Советского Союза И.Ф.  
Павлова

---

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ  
ПО ПРОВЕДЕНИЮ ЛАБОРАТОРНОЙ РАБОТЫ № \_5\_**

**по учебной дисциплине *Операционные системы и среды***

**Раздел 1. Основы теории операционных систем**

**Тема 1.5. Работа в различных операционных системах**

**Специальность 09.02.03 «Программирование в компьютерных системах»  
(базовый уровень)**

## Лабораторная работа №5

**Тема:** Утилиты в ОС Windows 7

**Цель работы:** «Изучение утилит TCP/IP в ОС Windows»

### Диагностические утилиты TCP/IP

В состав TCP/IP входят диагностические утилиты, предназначенные для проверки конфигурации стека и тестирования сетевого соединения.

Утилита	Применение
hostname	Выводит имя локального хоста. Используется без параметров.
ipconfig	Выводит значения для текущей конфигурации стека TCP/IP: IP-адрес, маску подсети, адрес шлюза по умолчанию, адреса WINS (Windows Internet Naming Service) и DNS (Domain Name System)
ping	Осуществляет проверку правильности конфигурирования TCP/IP и проверку связи с удаленным хостом.
tracert	Осуществляет проверку маршрута к удаленному компьютеру путем отправки эхо-пакетов протокола ICMP (Internet Control Message Protocol). Выводит маршрут прохождения пакетов на удаленный компьютер.
arp	Выводит для просмотра и изменения таблиц трансляции адресов, используемую протоколом разрешения адресов ARP (Address Resolution Protocol - определяет локальный адрес по IP-адресу)
route	Модифицирует таблицы маршрутизации IP. Отображает содержимое таблицы, добавляет и удаляет маршруты IP.
netstat	Выводит статистику и текущую информацию по соединению TCP/IP.

nslookup	Осуществляет проверку записей и доменных псевдонимов хостов, доменных сервисов хостов, а также информации операционной системы, путем запросов к серверам DNS.
----------	--

## 1. Проверка правильности конфигурации TCP/IP с помощью ipconfig.

При устранении неисправностей и проблем в сети TCP/IP следует сначала проверить правильность конфигурации TCP/IP. Для этого используется утилита ipconfig.

Эта команда полезна на компьютерах, работающих с DHCP (Dynamic Host Configuration Protocol), так как дает пользователям возможность определить, какая конфигурация сети TCP/IP и какие величины были установлены с помощью DHCP.

### Синтаксис:

```
ipconfig [/all | /renew[adapter] | /release]
```

### Параметры:

**all** выдает весь список параметров. Без этого ключа отображается только IP-адрес, маска и шлюз по умолчанию;

**renew[adapter]** обновляет параметры конфигурации DHCP для указанного сетевого адаптера;

**release[adapter]** освобождает выделенный DHCP IP-адрес;

**adapter** – имя сетевого адаптера;

**displaydns** выводит информацию о содержимом локального кэша клиента DNS, используемого для разрешения доменных имен.

Таким образом, утилита ipconfig позволяет выяснить, инициализирована ли конфигурация и не дублируются ли IP-адреса:

- если конфигурация инициализирована, то появляется IP-адрес, маска, шлюз;
- если IP-адреса дублируются, то маска сети будет 0.0.0.0;

- если при использовании DHCP компьютер не смог получить IP-адрес, то он будет равен 0.0.0.0 .

## **2. Тестирование связи с использованием утилиты *ping*.**

Утилита *ping* (Packet Internet Grouper) используется для проверки конфигурирования TCP/IP и диагностики ошибок соединения. Она определяет доступность и функционирование конкретного хоста. Использование *ping* лучший способ проверки того, что между локальным компьютером и сетевым хостом существует маршрут. Хостом называется любое сетевое устройство (компьютер, маршрутизатор), обменивающееся информацией с другими сетевыми устройствами по TCP/IP.

Команда *ping* проверяет соединение с удаленным хостом путем отправки к этому хосту эхо-пакетов ICMP и прослушивания эхо-ответов. *Ping* ожидает каждый посланный пакет и печатает количество переданных и принятых пакетов. Каждый принятый пакет проверяется в соответствии с переданным сообщением. Если связь между хостами плохая, из сообщений *ping* станет ясно, сколько пакетов потеряно.

По умолчанию передается 4 эхо-пакета длиной 32 байта (возможны и другие варианты значения по умолчанию) - периодическая последовательность символов алфавита в верхнем регистре. *Ping* позволяет изменить размер и количество пакетов, указать, следует ли записывать маршрут, который она использует, какую величину времени жизни (ttl) устанавливать, можно ли фрагментировать пакет и т.д.. При получении ответа в поле *time* указывается, за какое время (в миллисекундах) отправленный пакет доходит до удаленного хоста и возвращается назад. Так как значение по умолчанию для ожидания отклика равно 1 секунде, то все значения данного поля будут меньше 1000 миллисекунд. Если вы получаете сообщение «Request time out» (Превышен интервал ожидания), то, возможно, если увеличить время ожидания отклика, пакет дойдет до удаленного хоста. Это можно сделать с помощью ключа *-w*.

Ping можно использовать для тестирования как имени хоста (DNS или NetBIOS), так и его IP-адреса. Если ping с IP-адресом выполнялась успешно, а с именем – неудачно, это значит, что проблема заключается в распознавании соответствия адреса и имени, а не в сетевом соединении.

Утилита ping используется следующими способами:

1) Для проверки того, что TCP/IP установлен и правильно сконфигурирован на локальном компьютере, в команде ping задается адрес петли обратной связи (loopback address):

```
ping 127.0.0.1
```

Если тест успешно пройден, то вы получите следующий ответ:

```
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
```

2) Чтобы убедиться в том, что компьютер правильно добавлен в сеть и IP-адрес не дублируется, используется IP-адрес локального компьютера:

```
ping IP-адрес_локального_хоста
```

3) Чтобы проверить, что шлюз по умолчанию функционирует и что можно установить соединение с любым локальным хостом в локальной сети, задается IP-адрес шлюза по умолчанию:

```
ping IP-адрес_шлюза
```

4) Для проверки возможности установления соединения через маршрутизатор в команде ping задается IP-адрес удаленного хоста:

```
ping IP-адрес_удаленного_хоста
```

### **Синтаксис:**

```
ping [-t] [-a] [-n count] [-l length] [-f] [-i ttl] [-v tos] [-r count] [-s count] [ [-j host-list] |
```

```
[-k host-list] ] [-w timeout] destination-list
```

### **Параметры:**

-t выполняет команду ping до прерывания. Control-Break - посмотреть статистику и продолжить. Control-C - прервать выполнение команды;

-a позволяет определить доменное имя удаленного компьютера по его IP-адресу;

-n count посылает количество пакетов ECHO, указанное параметром count;

-l length посылает пакеты длиной length байт (максимальная длина 8192 байта);

-f посылает пакет с установленным флагом «не фрагментировать». Этот пакет не будет фрагментироваться на маршрутизаторах по пути своего следования;

-i ttl устанавливает время жизни пакета в величину ttl (каждый маршрутизатор уменьшает ttl на единицу);

-v tos устанавливает тип поля «сервис» в величину tos;

-r count записывает путь выходящего пакета и возвращающегося пакета в поле записи пути. Count - от 1 до 9 хостов;

-s count позволяет ограничить количество переходов из одной подсети в другую (хопов). Count задает максимально возможное количество хопов;

-j host-list направляет пакеты с помощью списка хостов, определенного параметром host-list. Последовательные хосты могут быть отделены промежуточными маршрутизаторами (гибкая статическая маршрутизация). Максимальное количество хостов в списке, позволенное IP, равно 9;

-k host-list направляет пакеты через список хостов, определенный в host-list. Последовательные хосты не могут быть разделены промежуточными маршрутизаторами (жесткая статическая маршрутизация). Максимальное количество хостов – 9;

-w timeout указывает время ожидания (timeout) ответа от удаленного хоста в миллисекундах (по умолчанию – 1 сек);

destination-list указывает удаленный хост, к которому надо направить пакеты ping.

**Пример использования утилиты ping:**

C:\WINDOWS>ping -n 10 www.netscape.com

Обмен пакетами с www.netscape.com [205.188.247.65] по 32 байт:

Ответ от 205.188.247.65: число байт=32 время=194мс TTL=48

Ответ от 205.188.247.65: число байт=32 время=240мс TTL=48

Ответ от 205.188.247.65: число байт=32 время=173мс TTL=48

Ответ от 205.188.247.65: число байт=32 время=250мс TTL=48

Ответ от 205.188.247.65: число байт=32 время=187мс TTL=48

Ответ от 205.188.247.65: число байт=32 время=239мс TTL=48

Ответ от 205.188.247.65: число байт=32 время=263мс TTL=48

Ответ от 205.188.247.65: число байт=32 время=230мс TTL=48

Ответ от 205.188.247.65: число байт=32 время=185мс TTL=48

Ответ от 205.188.247.65: число байт=32 время=406мс TTL=48

Статистика Ping для 205.188.247.65:

Пакетов: послано = 10, получено = 10, потеряно = 0 (0% потерь)

Приблизительное время передачи и приема:

Наименьшее = 173мс, наибольшее = 406мс, среднее =236мс

В случае невозможности проверить доступность хоста утилита выводит информацию об ошибке. Ниже приведен пример ответа утилиты ping при попытке послать запрос на несуществующий хост.

Обмен пакетами с 172.16.6.21 по 32 байт:

Превышен интервал ожидания для запроса.

Превышен интервал ожидания для запроса.

Превышен интервал ожидания для запроса.

Превышен интервал ожидания для запроса.

Статистика Ping для 172.16.6.21:

Пакетов: отправлено = 4, получено = 0, потеряно = 4 (100% потерь),

Приблизительное время передачи и приема:

наименьшее = 0мс, наибольшее = 0мс, среднее = 0мс

Утилита сообщает не об отсутствии хоста, а о том, что за отведенное время не был получен ответ на посланный запрос. Причиной этого не обязательно является отсутствие хоста в сети. Проблема может крыться в сбоях связи, перегрузке или неправильной настройке маршрутизаторов и т. п. Ошибка «сеть недоступна» (network unreachable) прямо указывает на проблемы маршрутизации.

### **3. Изучение маршрута между сетевыми соединениями с помощью утилиты *tracert*.**

Tracert - это утилита трассировки маршрута. Она использует поле TTL (time-to-live, время жизни) пакета IP и сообщения об ошибках ICMP для определения маршрута от одного хоста до другого.

Утилита tracert может быть более содержательной и удобной, чем ping, особенно в тех случаях, когда удаленный хост недостижим. С помощью нее можно определить район проблем со связью (у Internet-провайдера, в опорной сети, в сети удаленного хоста) по тому, насколько далеко будет отследен маршрут. Если возникли проблемы, то утилита выводит на экран звездочки (\*), либо сообщения типа «Destination net unreachable», «Destination host unreachable», «Request time out», «Time Exceeded».

Утилита tracert работает следующим образом: посылаются по 3 пробных эхо-пакета на каждый хост, через который проходит маршрут до удаленного хоста. На экран при этом выводится время ожидания ответа на каждый пакет (Его можно изменить с помощью параметра -w). Пакеты посылаются с различными величинами времени жизни. Каждый маршрутизатор, встречающийся по пути, перед перенаправлением пакета уменьшает величину TTL на единицу. Таким образом, время жизни является счетчиком точек промежуточной доставки (хопов). Когда время жизни пакета достигнет нуля, предполагается, что маршрутизатор пошлет в компьютер-источник сообщение ICMP «Time Exceeded» (Время истекло). Маршрут определяется путем послышки первого эхо-пакета с TTL=1. Затем TTL увеличивается на 1 в каждом последующем пакете до тех пор, пока пакет не достигнет удаленного



хоста, либо будет достигнута максимально возможная величина TTL (по умолчанию 30, задается с помощью параметра -h).

Маршрут определяется путем изучения сообщений ICMP, которые присылаются обратно промежуточными маршрутизаторами.

Примечание: некоторые маршрутизаторы просто уничтожают пакеты с истекшим TTL и не будут видны утилите tracert.

### **Синтаксис:**

```
tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout]  
имя_целевого_хоста
```

### **Параметры:**

-d указывает, что не нужно распознавать адреса для имен хостов;

-h maximum\_hops указывает максимальное число хопов для того, чтобы искать цель;

-j host-list указывает нежесткую статическую маршрутизацию в соответствии с host-list;

-w timeout указывает, что нужно ожидать ответ на каждый эхо-пакет заданное число мсек.

## **4. Утилита *arp*.**

Основная задача протокола ARP – трансляция IP-адресов в соответствующие локальные адреса. Для этого ARP-протокол использует информацию из ARP-таблицы (ARP-кэша). Если необходимая запись в таблице не найдена, то протокол ARP отправляет широковещательный запрос ко всем компьютерам локальной подсети, пытаясь найти владельца данного IP-адреса. В кэше могут содержаться два типа записей: статические и динамические. Статические записи вводятся вручную и хранятся в кэше постоянно. Динамические записи помещаются в кэш в результате выполнения широковещательных запросов. Для них существует понятие времени жизни. Если в течение определенного времени (по умолчанию 2 мин.) запись не была востребована, то она удаляется из кэша.

### **Синтаксис:**

arp [-s inet\_addr eth\_addr] | [-d inet\_addr] | [-a]

### **Параметры:**

- s занесение в кэш статических записей;
- d удаление из кэша записи для определенного IP-адреса;
- a просмотр содержимого кэша для всех сетевых адаптеров локального компьютера;
- inet\_addr - IP-адрес;
- eth\_addr - MAC-адрес.

### **5. Утилита route.**

Утилита **route** предназначена для работы с локальной таблицей маршрутизации. Она имеет следующий **синтаксис**:

route [-f] [-p] [команда [узел] [MASK маска] [шлюз] [METRIC метрика] [IF интерфейс]]

### **Параметры:**

- f Очистка таблицы маршрутизации.
- p При указании совместно с командой ADD создает постоянную запись, которая сохраняется после перезагрузки компьютера. По умолчанию записи таблицы маршрутов не сохраняются при перезагрузке.

*команда* одна из четырех команд:

PRINT - вывод информации о маршруте;

ADD - добавление маршрута;

DELETE - удаление маршрута;

CHANGE - изменение маршрута.

*узел* адресуемый узел

*маска* маска подсети; по умолчанию используется маска 255.255.255.255

*шлюз* адрес шлюза

*метрика* метрика маршрута;

*интерфейс* идентификатор интерфейса, который будет использован для пересылки пакета

Для команд PRINT и DELETE возможно использование символов подстановки при указании адресуемого узла или шлюза. Параметр шлюза для этих команд может быть опущен. При добавлении и изменении маршрутов утилита route осуществляет проверку введенной информации на соответствие условию (УЗЕЛ & МАСКА) == УЗЕЛ. Если это условие не выполняется, то утилита выдает сообщение об ошибке и не добавляет или не изменяет маршрут.

Утилита осуществляет поиск имен сетей в файле networks. Поиск имен шлюзов осуществляется в файле hosts. Оба файла расположены в папке %systemroot%\system32\drivers\etc. Наличие и заполнение этих файлов не обязательно для нормального функционирования утилиты route и работы маршрутизации.

Хотя в большинстве случаев на рабочей станции это не требуется, можно вручную редактировать таблицы маршрутизации.

#### ***Пример использования утилиты route:***

Добавление статического маршрута:

```
route add 172.16.6.0 MASK 255.255.255.0 172.16.11.1 METRIC 1 IF 0x1000003
```

#### **6. Утилита netstat.**

Утилита netstat позволяет получить статическую информацию по некоторым из протоколов стека (TCP, UDP, IP, ICMP), а также выводит сведения о текущих сетевых соединениях. Особенно она полезна на брандмауэрах, с ее помощью можно обнаружить нарушения безопасности периметра сети.

#### **Синтаксис:**

```
netstat [-a] [-e] [-n] [-s] [-p protocol] [-r]
```

#### **Параметры:**

- a выводит перечень всех сетевых соединений и прослушивающихся портов локального компьютера;
- e выводит статистику для Ethernet-интерфейсов (например, количество полученных и отправленных байт);

- n выводит информацию по всем текущим соединениям (например, TCP) для всех сетевых интерфейсов локального компьютера. Для каждого соединения выводится информация об IP-адресах локального и удаленного интерфейсов вместе с номерами используемых портов;
- s выводит статистическую информацию для протоколов UDP, TCP, ICMP, IP. Ключ «/more» позволяет просмотреть информацию постранично;
- r выводит содержимое таблицы маршрутизации.

### **7. Утилита *nslookup*.**

Утилита **nslookup** предназначена для диагностики службы DNS, в простейшем случае - для выполнения запросов к DNS-серверам на разрешение имен в IP-адреса. В общем случае утилита позволяет просмотреть любые записи DNS-сервера:

*A* – каноническое имя узла, устанавливает соответствие доменного имени ip-адресу.

*SOA* – начало полномочий, начальная запись, единственная для зоны;

*MX* – почтовые серверы (хосты, принимающие почту для заданного домена);

*NS* – серверы имен (содержит авторитетные DNS-серверы для зоны);

*PTR* – указатель (служит для обратного преобразования ip-адреса в символьное имя хоста)

и т. д.

Утилита *nslookup* достаточно сложна и содержит свой собственный командный интерпретатор.

В простейшем случае (без входа в командный режим) утилита **nslookup** имеет следующий

#### **Синтаксис:**

`nslookup хост [сервер]`

#### **Параметры:**

*Хост*           DNS-имя хоста, которое должно быть преобразовано в IP-адрес.

*Сервер* Адрес DNS-сервера, который будет использоваться для разрешения имени. Если этот параметр опущен, то будут последовательно использованы адреса DNS-серверов из параметров настройки протокола TCP/IP.

***Примеры использования утилиты nslookup:***

1. Получение списка серверов имен для домена yandex.ru без входа в командный режим (с использованием ключей).

```
C:\> nslookup -type=ns yandex.ru
```

```
Server: dns01.catv.ext.ru
```

```
Address: 217.10.44.35
```

```
Non-authoritative answer:
```

```
yandex.ru    nameserver = ns4.yandex.ru
```

```
yandex.ru    nameserver = ns5.yandex.ru
```

```
yandex.ru    nameserver = ns2.yandex.ru
```

```
yandex.ru    nameserver = ns1.yandex.ru
```

```
ns2.yandex.ru internet address = 213.180.199.34
```

```
ns5.yandex.ru internet address = 213.180.204.1
```

2. Получение записи SOA домена yandex.ru с авторитетного сервера с использованием командного интерпретатора nslookup.

```
C:\>nslookup
```

```
Default Server: dns04.catv.ext.ru
```

```
Address: 217.10.39.4
```

```
> set type=SOA
```

```
> server ns2.yandex.ru
```

```
Default Server: ns2.yandex.ru
```

```
Address: 213.180.199.34
```

```
> yandex.ru
```

```
Server: ns1.yandex.ru
```

Address: 213.180.193.1

>yandex.ru

primary name server = ns1.yandex.ru

responsible mail addr = sysadmin.yandex-team.r

serial = 2009022707

refresh = 1800 (30 mins)

retry = 900 (15 mins)

expire = 2592000 (30 days)

default TTL = 900 (15 mins)

yandex.ru nameserver = ns5.yandex.ru

yandex.ru nameserver = ns1.yandex.ru

yandex.ru nameserver = ns4.yandex.ru

yandex.ru nameserver = ns2.yandex.ru

ns1.yandex.ru internet address = 213.180.193.1

ns2.yandex.ru internet address = 213.180.199.34

ns4.yandex.ru internet address = 77.88.19.60

ns5.yandex.ru internet address = 213.180.204.1

> exit

3. Получение адреса почтового сервера для домена yandex.ru.

C:\ >nslookup

Default Server: dns01.catv.ext.ru

Address: 217.10.44.35

> set q=mx

> yandex.ru

Server: dns01.catv.ext.ru

Address: 217.10.44.35

Non-authoritative answer:

yandex.ru MX preference = 10, mail exchanger = mx2.yandex.ru

yandex.ru MX preference = 10, mail exchanger = mx3.yandex.ru

yandex.ru MX preference = 10, mail exchanger = mx1.yandex.ru

```
yandex.ru    nameserver = ns2.yandex.ru
yandex.ru    nameserver = ns1.yandex.ru
yandex.ru    nameserver = ns4.yandex.ru
yandex.ru    nameserver = ns5.yandex.ru
mx1.yandex.ru internet address = 77.88.21.89
mx2.yandex.ru internet address = 93.158.134.89
mx3.yandex.ru internet address = 213.180.204.89
ns2.yandex.ru internet address = 213.180.199.34
ns4.yandex.ru internet address = 77.88.19.60
ns5.yandex.ru internet address = 213.180.204.1
```

>

Указав ключ `type=any`, можно получить все записи о узле или домене. Ключи `querytype`, `t`, `q` эквивалентны `type`.

### **Задания на лабораторную работу**

1. Изучите методические указания к лабораторной работе.
2. Выполните упражнения.
3. Оформите отчет по лабораторной работе, описав выполнение упражнений и дав краткие ответы на контрольные вопросы.

#### **Упражнение 1. Получение справочной информации по командам.**

Выведите на экран справочную информацию по всем рассмотренным утилитам (см. таблицу п.1). Для этого в командной строке введите имя утилиты без параметров и дополните `/?`.

Сохраните справочную информацию в отдельном файле.

Изучите ключи, используемые при запуске утилит.

#### **Упражнение 2. Получение имени хоста.**

Выведите на экран имя локального хоста с помощью команды `hostname`.

Сохраните результат в отдельном файле.

#### **Упражнение 3. Изучение утилиты `ipconfig`.**

Проверьте конфигурацию TCP/IP с помощью утилиты `ipconfig`. Заполните таблицу:

Имя хоста	
IP-адрес	
Маска подсети	
Основной шлюз	
Используется ли DHCP (адрес DHCP-сервера)	
Описание адаптера	
Физический адрес сетевого адаптера	
Адрес DNS-сервера	
Адрес WINS-сервера	

#### **Упражнение 4. Тестирование связи с помощью утилиты ping.**

1. Проверьте правильность установки и конфигурирования TCP/IP на локальном компьютере.
2. Проверьте функционирование основного шлюза, пошлав 5 эхо-пакетов длиной 64 байта.
3. Проверьте возможность установления соединения с удаленным хостом.
4. С помощью команды ping проверьте адреса (взять из списка локальных ресурсов на сайте aspu.ru) и для каждого из них отметьте время отклика. Попробуйте изменить параметры команды ping таким образом, чтобы увеличилось время отклика. Определите IP-адреса узлов.

#### **Упражнение 5. Определение пути IP-пакета.**

С помощью команды tracert проверьте для перечисленных ниже адресов, через какие промежуточные узлы идет сигнал. Изучите ключи команды.

- a) aspu.ru
- b) mathmod.aspu.ru
- c) yarus.aspu.ru



### **Упражнение 6: Просмотр ARP-кэша.**

С помощью утилиты arp просмотрите ARP-таблицу локального компьютера.

Внести в кэш локального компьютера любую статическую запись.

### **Упражнение 7: Просмотр локальной таблицы маршрутизации.**

С помощью утилиты route просмотреть локальную таблицу маршрутизации.

### **Упражнение 8. Получение информации о текущих сетевых соединениях и протоколах стека TCP/IP.**

С помощью утилиты netstat выведите перечень сетевых соединений и статистическую информацию для протоколов UDP, TCP, ICMP, IP.

### **Упражнение 9. Получение DNS-информации с помощью nslookup.**

- 1) Узнайте ip-адреса узлов, список которых приводится на странице <http://www.aspu.ru/id/2433>.
- 2) Узнайте авторитетные (компетентные) сервера для этих узлов.
- 3) Получите запись SOA с одного из этих серверов для домена mathmod.aspu.ru.

## **Контрольные вопросы**

1. Раскрыть термины: хост, шлюз, хоп, время жизни пакета, маршрут, маска сети, авторитетный/неавторитетный (компетентный) DNS-сервер, порт TCP, петля обратной связи, время отклика.
2. Какие утилиты можно использовать для проверки правильности конфигурирования TCP/IP?
3. Каким образом команда ping проверяет соединение с удаленным хостом?
4. Каково назначение протокола ARP?
5. Как утилита ping разрешает имена узлов в ip-адреса (и наоборот)?
6. Какие могут быть причины неудачного завершения ping и tracert? (превышен интервал ожидания для запроса, сеть недоступна, превышен срок жизни при передаче пакета).

7. Всегда ли можно узнать символическое имя узла по его ip-адресу?
8. Какой тип записи запрашивает у DNS-сервера простейшая форма nslookup?