

САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО

КАФЕДРА КОМПЬЮТЕРНЫХ СИСТЕМ И ПРОГРАММНЫХ ТЕХНОЛОГИЙ

**Отчёт по лабораторной работе №3**

**Курс: «Администрирование компьютерных сетей»**

**Тема: «Администрирование сетевых сервисов»**

Выполнил студент:

Волкова Мария Дмитриевна

Группа: 13541/2

Проверил:

Малышев Игорь Алексеевич

Санкт-Петербург  
2019 г.

# Содержание

<b>1</b>	<b>Лабораторная работа №3</b>	<b>2</b>
1.1	Цели работы . . . . .	2
1.2	Сведения о системе . . . . .	2
1.3	Структура сети . . . . .	2
1.4	DHCP и удаленная загрузка . . . . .	2
1.4.1	Настройка DHCP на FreeBSD . . . . .	2
1.4.2	Настройка DHCP на Ubuntu . . . . .	3
1.4.3	Настройка TFTP на Ubuntu . . . . .	4
1.4.4	Проверка . . . . .	4
1.5	DNS сервисы . . . . .	5
1.5.1	Кэширующий сервер . . . . .	5
1.5.2	Первичный и вторичный DNS серверы . . . . .	6

# Лабораторная работа №3

## 1.1 Цели работы

1. Изучение состава и функциональных возможностей сетевых сервисов операционных систем.
2. Разработка и настройка сервисов локальной сети.
3. Разработка и настройка сервисов демилитаризованной зоны.
4. Разработка и настройка сервисов пограничной зоны.

## 1.2 Сведения о системе

Работа производилась на реальной системе, со следующими характеристиками:

Элемент	Значение
Имя ОС	Майкрософт Windows 10 Pro (Registered Trademark)
Версия	10.0.16299 Сборка 16299
RAM	16,00 ГБ
Процессор	Intel(R) Core(TM) i5-7300HQ CPU @ 2.50GHz, 2496 МГц, ядер: 4, логических процессоров: 4

Для выполнения работы использовалась **VMware Workstation 12 pro (12.5.7 build-5813279)**

## 1.3 Структура сети

В качестве сети для экспериментов, использовалась ККС из прошлой работы, ОС семейства Windows не использовались. Были добавлены две виртуальных машины с ОС приведенными далее.

Название	Версия	Объем RAM
Ubuntu	16.04 amd64	1.5 GB
Linux mint	18.3 Cinnamon 64 bit	1.5 GB

Также к сетям **VMnet2** и **VMnet3** были подключены бездисковые клиенты.

Название сети	Адрес сети	Подключенные узлы	DHCP
VMnet1	192.168.40.0	Linux mint, NetBSD, FreeBSD	-
VMnet2	192.168.80.0	FreeBSD, NoDisk1	-
VMnet3	192.168.120.0	FreeBSD, Ubuntu, NoDisk2	-
VMnet4	192.168.32.0	NetBSD	+

## 1.4 DHCP и удаленная загрузка

В данной работе используются два DHCP сервера, на FreeBSD и на Ubuntu, для двух бездисковых клиентов из разных сетей. Также на Ubuntu будет установлен TFTP сервер для удаленной загрузки ОС.

### 1.4.1 Настройка DHCP на FreeBSD

1. Переходим в каталог `/usr/ports/net/isc-dhcp3-server`
2. Устанавливаем следующей командой:

```
1 make install clean
```

3. Вносим в файл `/etc/rc.conf` следующие строки:

```
1 # Включаем DHCP
2 dhcpd_enable="YES"
3 # Отключаем вывод избыточной информации
4 dhcpd_flags="-q"
5 # Указываем интерфейс для запуска
6 dhcpd_ifaces="em1"
```

4. Создаем файл `/usr/local/etc/dhcpd.conf` и вносим в него следующие изменения:

```
1 option domain-name "example.org"; # доменное имя
2 option domain-name-servers 192.168.32.2; #DNS сервер
3 default-lease-time 600;
4 max-lease-time 7200;
5
6 subnet 192.168.80.0 netmask 255.255.255.0 {
7     range 192.168.80.127 192.168.80.224;
8     option routers 192.168.80.2;
9     option root-path "192.168.120.3:/usr/tftpboot/";
10    next-server 192.168.120.3;
11    filename "gpxelinux.0";
12 }
```

### 1.4.2 Настройка DHCP на Ubuntu

Для начала необходимо корректно сконфигурировать сетевой адаптер для выхода в сеть "Интернет". Для этого были заданы следующие параметры:

- Адрес - 192.168.120.3;
- Маска - 255.255.255.0;
- Шлюз - 192.168.120.2;
- DNS сервер - 192.168.32.2.

После настройки адаптера переходим к основной настройке TFTP сервера.

1. Выполняем команду:

```
1 sudo apt-get install isc-dhcp-server
```

2. В файл `/etc/dhcp/dhcpd.conf` вносим следующие изменения:

```
1 option domain-name "example.org";
2 option domain-name-servers 192.168.32.2;
3
4 default-lease-time 600;
5 max-lease-time 7200;
6
7 subnet 192.168.120.0 netmask 255.255.255.0 {
8     range 192.168.120.100 192.168.120.200;
9     option routers 192.168.120.2;
10    next-server 192.168.120.3; # TFTP server address
11    filename "gpxelinux.0"; # PXE boot loader filename
12 }
```

3. Перезапускаем DHCP-сервер

```
1 sudo /etc/init.d/isc-dhcp-server restart
```

### 1.4.3 Настройка TFTP на Ubuntu

1. Выполняем установку пакетов командой:

```
1 sudo apt-get install tftp tftpd-hpa
```

2. Создаем необходимые для работы директории:

```
1 mkdir -p /usr/tftpboot/images
2 mkdir /usr/tftpboot/pxelinux.cfg
```

3. В файл `/etc/rc.conf` вносим изменения:

```
1 tftpd_enable="YES"
2 tftpd_flags="-p -s /usr/tftpboot -B 1024 --ipv4"
```

4. Запускаем TFTP-сервер

```
1 service tftpd start
```

5. Скачиваем Syslinux версии 6.03 с сайта:

```
1 http://www.syslinux.org/wiki/index.php?title=The_Syslinux_Project
```

6. Извлекаем по пути `/usr/tftpboot` следующие файлы:

- chain.c32
- gpxlinux.0
- ldlinux.c32
- libutil.c32
- memdisk
- menu.c32
- reboot.c32
- vesamenu.c32

7. Скачиваем образ ОС, поддерживающей liveCD, например ubuntu;

8. Разархивируем образ ОС, по пути `/usr/tftpboot/images/ubuntu/`;

9. По пути `/usr/tftpboot/pxelinux.cfg` создаем файл **default** со следующим содержанием:

```
1 ui menu.c32
2 menu title Netboot OS
3
4 LABEL ubuntu
5     kernel images/ubuntu/casper/vmlinuz.efi
6     append root=/dev/nfs boot=casper netboot=nfs nfsroot=192.168.120.3:/usr/
    tftpboot/images/ubuntu initrd=images/ubuntu/casper/initrd.lz
```

10. Перезгружаем ОС.

### 1.4.4 Проверка

Запускаем бездисковые клиенты. При запуске всплывет меню, с выбором дальнейшей загрузки, необходимо выбрать пункт с **Ubuntu**. Далее, при успешной загрузке попадаем на рабочий стол ОС, что говорит об успешной удаленной загрузке. Дополнительно можно посмотреть с помощью команды **ifconfig** какой адрес был выдан DHCP сервером.

При корректных настройках, адрес будет находиться в диапазоне адресов, указанных на DHCP сервере, а также клиент будет иметь возможность выхода в сеть "Интернет".

## 1.5 DNS сервисы

В данном разделе будут приведены примеры различных DNS серверов, в частности:

1. **Кэширующий сервер** - ищет все ответы на запросы от пользователей и запоминает их на случай повторного запроса
2. **Первичный мастер** - читает данные зоны из локального файла и является ответственным за эту зону
3. **Вторичный мастер** - получает данные по зоне с другого сервера имен, отвечающего за эту зону

### 1.5.1 Кэширующий сервер

#### Настройка

Установим кэширующий DNS на Linux Mint(192.168.40.32)

1. Устанавливаем пакет **bind** командой

```
1 sudo apt-get install bind9
```

2. Внесем изменения в файл **/etc/bind/named.conf.options**:

```
1 options {  
2     directory "/var/cache/bind";  
3     forwarders {  
4         8.8.8.8;  
5     };  
6     dnssec-validation auto;  
7     auth-nxdomain no;  
8     listen-on-v6 {any};  
9 };
```

3. Перезапускаем DNS сервер

```
1 sudo /etc/init.d/bind9 restart
```

#### Проверка

Настроим сетевой адаптер хост **192.168.120.3** на использование данного DNS сервера. В файле **/etc/NetworkManager/NetworkManager.conf** комментируем строку **dns=dnsmasq**, иначе в nslookup будет другой адрес DNS. Перезагружаем систему. Вводим команду **nslookup ya.ru**, после чего в консоль будет выведено:

```
1 Server:      192.168.40.32  
2 Address:     192.168.40.32#53  
3  
4 Non-authoritative answer:  
5 Name:        ya.ru  
6 Address:     87.250.250.242
```

Как видно, DNS успешно был определен.

Дополнительно проверим кэширующую способность DNS сервера, для этого введем команду **dig google.com** два раза:

```
1 dig google.com  
2 ...  
3 ;; Query time: 31 msec  
4 ;; SERVER: 192.168.40.32#53(192.168.40.32)  
5 ;; WHEN: Fri Mar 23 11:55:03 PDT 2018  
6 ;; MSG SIZE rcvd: 346  
7  
8  
9 dig google.com  
10 ...  
11 ;; Query time: 0 msec  
12 ;; SERVER: 192.168.40.32#53(192.168.40.32)  
13 ;; WHEN: Fri Mar 23 11:55:03 PDT 2018  
14 ;; MSG SIZE rcvd: 346
```

Как видно, время запроса снизилось с 32 до 0 миллисекунд, что означает о наличии кэширующей возможности у DNS сервера.

## 1.5.2 Первичный и вторичный DNS серверы

Первичный сервер - Linux mint **192.168.40.32**, вторичный - Ubuntu **192.168.120.3**.

### Настройка первичного DNS сервера

В файл `/etc/bind/named.conf.local` вносим изменения:

```
1 zone "example.com" {
2     type master;
3     file "/etc/bind/db.example.com";
4     allow-transfer {192.168.40.0/24;192.168.80.0/24;192.168.120.0/24;};
5 };
6 zone "40.168.192.in-addr.arpa" {
7     type master;
8     file "/etc/bind/db.192";
9     allow-transfer {192.168.40.0/24;192.168.80.0/24;192.168.120.0/24;};
10 };
```

Были созданы зоны **example.com** и обратная ей зона(для получения имени по IP)

Содержимое файла **db.example.com** выглядит следующим образом:

```
1 ;
2 ; BIND data file for example.com
3 ;
4 $TTL      604800
5 @      IN      SOA      example.com.  root.example.com.  (
6         1              ; Serial
7         604800          ; Refresh
8         86400           ; Retry
9         2419200         ; Expire
10        604800 )        ; Negative Cache TTL
11      IN A      192.168.40.32
12 ;
13 @      IN      NS       ns.example.com
14 @      IN      A        192.168.40.32
15 @      IN      AAAA     ::1
16 ns     IN      A        192.168.40.32
```

Содержимое файла **db.192** выглядит следующим образом:

```
1 ;
2 ; BIND reverse data file for local 192.168.40.XXX net
3 ;
4 $TTL      604800
5 @      IN      SOA      ns.example.com.  root.example.com.  (
6         1              ; Serial
7         604800          ; Refresh
8         86400           ; Retry
9         2419200         ; Expire
10        604800 )        ; Negative Cache TTL
11 ;
12 @      IN      NS       ns.
13 32     IN      PTR      ns.example.com.
```

В данном случае адресу 192.168.40.32 соответствует имя example.com.

### Настройка вторичного DNS сервера

1. Устанавливаем пакет bind9;
2. В файл `/etc/bind/named.conf.local` вносим изменения:

```
1 zone "example.com" {
2     type slave;
3     file "db.example.com";
4     masters { 192.168.40.32; };
5 };
6 zone "40.168.192.in-addr.arpa" {
7     type slave;
8     file "db.192";
9     masters { 192.168.40.32; };
10};
```

Теперь данный сервер является вторичным по отношению к 192.168.40.32.

## Проверка

На первичном сервере вводим команду `cat /etc/log/syslog`

```
1 ...
2 ... zone example.com/IN: loaded serial 1
3 ... zone 40.168.192.in-addr.arpa/IN: loaded serial 1
4 ... zone 255.in-addr.arpa/IN: loaded serial 1
5 ... zone localhost/IN: loaded serial 2
6 ... all zones loaded
7 ... running
8 ... zone 40.168.192.in-addr.arpa/IN: sending notifies(serial 1)
9 ...
```

Как видно из части лога, зоны были успешно загружены и переданы на вторичный сервер.

## Вывод

В данной работе был получен опыт по настройке DHCP и TFTP серверов, что позволяет загружать по сети ОС или утилиту.

Также были применены навыки работы с различными по функциональности DNS серверами. Так был получен кэширующий DNS сервер, который существенно ускоряет обращение к сетевым ресурсам. Также были сконфигурированы первичный и вторичный DNS сервера, ответственные за домен example.com и обратную зону.