

Санкт-Петербургский политехнический университет Петра Великого  
Институт компьютерных наук и технологий

**Кафедра компьютерных систем и программных технологий**

**Отчет о лабораторной работе №3**

**Курс:** Администрирование компьютерных сетей

**Тема:** Администрирование сетевых сервисов

Выполнил студент группы 13541/3

\_\_\_\_\_  
(подпись) Д.В. Круминьш

Преподаватель

\_\_\_\_\_  
(подпись) И.А. Малышев

Санкт-Петербург  
2018 г.

# 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.

## 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
Установленная оперативная память (RAM)	16,00 ГБ
Процессор	Intel(R) Core(TM) i5-7300HQ CPU @ 2.50GHz, 2496 МГц, ядер: 4, логических процессоров: 4

Таблица 1: Сведения о системе

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

## 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

Таблица 2: Операционные системы новых виртуальных машин

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	✓

Таблица 3: Сети и подключенные к ним узлы

## 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.

### 4.1 Настройка DHCP на FreeBSD

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

```
make install clean
```

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

```
# Включаем DHCP
dhcpd_enable="YES"
# Отключаем вывод избыточной информации
dhcpd_flags="-q"
# Указываем интерфейс для запуска
dhcpd_ifaces="em1"
```

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

```
option domain-name "example.org"; # доменное имя
option domain-name-servers 192.168.32.2; #DNS сервер
default-lease-time 600;
max-lease-time 7200;

subnet 192.168.80.0 netmask 255.255.255.0 {
```

```
range 192.168.80.127 192.168.80.224;
option routers 192.168.80.2;
option root-path "192.168.120.3:/usr/tftpboot/";
next-server 192.168.120.3;
filename "gpxelinux.0";
}
```

## 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:

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

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

```
option domain-name "example.org";
option domain-name-servers 192.168.32.2;

default-lease-time 600;
max-lease-time 7200;

subnet 192.168.120.0 netmask 255.255.255.0 {
    range 192.168.120.100 192.168.120.200;
    option routers 192.168.120.2;
    next-server 192.168.120.3; # TFTP server address
    filename "gpxelinux.0"; # PXE boot loader filename
}
```

3. Перезапускаем DHCP-сервер

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

## 4.3 Настройка TFTP на Ubuntu

1. Выполняем установку пакетов командой:

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

2. Создаем необходимые для работы директории:

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

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

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

4. Запускаем TFTP-сервер

```
service tftpd start
```

5. Скачиваем Syslinux версии 6.03 с сайта:

```
http://www.syslinux.org/wiki/index.php?title=The\_Syslinux\_Project
```

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

- |               |                |
|---------------|----------------|
| • chain.c32   | • memdisk      |
| • gpxlinux.0  | • menu.c32     |
| • ldlinux.c32 | • reboot.c32   |
| • libutil.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:

```
ui menu.c32  
menu title Netboot OS  
  
LABEL ubuntu  
    kernel images/ubuntu/casper/vmlinuz.efi  
    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.

## 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".

## 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

### 5.1 Кэширующий сервер

#### 5.1.1 Настройка

Установим кэширующий DNS на Linux Mint(192.168.40.32)

1. Устанавливаем пакет **bind** командой

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

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

```
options {  
    directory "/var/cache/bind";  
    forwarders {
```

```
    8.8.8.8;
};
dnssec-validation auto;
auth-nxdomain no;
listen-on-v6 {any};
};
```

### 3. Перезапускаем DNS сервер

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

#### 5.1.2 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:

```
Server:          192.168.40.32
Address:         192.168.40.32#53

Non-authoritative answer:
Name:   ya.ru
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:

```
dig google.com
...
;; Query time: 31 msec
;; SERVER: 192.168.40.32#53(192.168.40.32)
;; WHEN: Fri Mar 23 11:55:03 PDT 2018
;; MSG SIZE rcvd: 346

dig google.com
...
;; Query time: 0 msec
;; SERVER: 192.168.40.32#53(192.168.40.32)
;; WHEN: Fri Mar 23 11:55:03 PDT 2018
;; MSG SIZE rcvd: 346
```

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

## 5.2 Первичный и вторичный DNS серверы

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

### 5.2.1 Настройка первичного DNS сервера

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

```
zone "example.com" {
    type master;
    file "/etc/bind/db.example.com";
    allow-transfer {192.168.40.0/24;192.168.80.0/24;192.168.120.0/24;};
};
zone "40.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/db.192";
    allow-transfer {192.168.40.0/24;192.168.80.0/24;192.168.120.0/24;};
};
```

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

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

```
;
; BIND data file for example.com
;
$TTL      604800
@ IN SOA  example.com. root.example.com. (
        1           ; Serial
        604800      ; Refresh
        86400       ; Retry
        2419200     ; Expire
        604800 )    ; Negative Cache TTL
;
@ IN A    192.168.40.32
;
@ IN NS   ns.example.com
@ IN A    192.168.40.32
@ IN AAAA ::1
ns IN A    192.168.40.32
```



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

```
;
; BIND reverse data file for local 192.168.40.XXX net
;
$TTL      604800
@         IN      SOA      ns.example.com. root.example.com. (
                                1             ; Serial
                                604800        ; Refresh
                                86400         ; Retry
                                2419200       ; Expire
                                604800 )      ; Negative Cache TTL
;
@         IN      NS       ns.
32        IN      PTR      ns.example.com.
```

В данном случае адресу 192.168.40.32 соответствует имя example.com.

### 5.2.2 Настройка вторичного DNS сервера

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

```
zone "example.com" {
    type slave;
    file "db.example.com";
    masters { 192.168.40.32; };
};
zone "40.168.192.in-addr.arpa" {
    type slave;
    file "db.192";
    masters { 192.168.40.32; };
};
```

Теперь данный сервер является вторичным по отношению к 192.168.40.32.

### 5.2.3 Проверка

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

```
...
... zone example.com/IN: loaded serial 1
... zone 40.168.192.in-addr.arpa/IN: loaded serial 1
```

```
... zone 255.in-addr.arpa/IN: loaded serial 1
... zone localhost/IN: loaded serial 2
... all zones loaded
... running
... zone 40.168.192.in-addr.arpa/IN: sending notifies (serial 1)
...
```

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.