



Министерство науки и высшего образования Российской Федерации  
Калужский филиал  
федерального государственного автономного  
образовательного учреждения высшего образования  
«Московский государственный технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(КФ МГТУ им. Н.Э. Баумана)

**ФАКУЛЬТЕТ** ИУК «Информатика и управление»

**КАФЕДРА** ИУК4 «Программное обеспечение ЭВМ, информационные технологии»

## ЛАБОРАТОРНАЯ РАБОТА 2

**«Настройка и использование серверов в локальной  
вычислительной сети»**

**ДИСЦИПЛИНА: «Компьютерные сети и интернет-технологии»**

Выполнил: студент гр. ИУК4-62Б \_\_\_\_\_ (\_\_\_\_ Губин Е.В.\_\_\_\_)  
(Подпись) (Ф.И.О.)

Проверил: \_\_\_\_\_ (\_\_\_\_ Прудяк П.Н.\_\_\_\_)  
(Подпись) (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Калуга , 2025

**Цель:** получить практические навыки по работе и настройке прокси-сервера Squid под ОС FreeBSD, по настройке системы DNS в ОС FreeBSD.

**Задачи:**

- научиться получать и устанавливать прокси-сервер Squid под ОС FreeBSD
- научиться настраивать и управлять прокси-сервером Squid под ОС FreeBSD
- научиться настраивать DNS-клиент и DNS-сервер в ОС FreeBSD

## Ход выполнения работы

```
# pkg install squid
```

### 1 – Установка Squid

Установка прокси-сервера Squid с помощью команды `pkg install squid`. Это базовый этап, позволяющий получить необходимое ПО для последующей настройки каскадного прокси-сервера в ОС FreeBSD.

```
# cd /usr/local
# ls
bin      etc      include lib      libdata libexec sbin      share
# cd etc
# ls
bash_completion.d  periodic      pkg.conf.sample  squid
man.d              pkg.conf
# cd squid/
# ls
cachemgr.conf      errorpage.css      errors            mib.txt           mime.conf.sample  squid.conf.documented
cachemgr.conf.sample  errorpage.css.sample  icons            mime.conf          squid.conf         squid.conf.sample
# vi squid.conf
```

### 2 – Переход в конфигурационный каталог и редактирование squid.conf

После установки производится переход к конфигурационному файлу. Открытие `squid.conf` позволяет задать поведение прокси-сервера, включая кэш, порты.

```
#
# Add any of your own refresh_pattern entries above these.
#
refresh_pattern ^ftp:          1440      20%      10080
refresh_pattern -i (/cgi-bin/|\?) 0        0%        0
refresh_pattern .              0        20%      4320
visible_hostname minazuki
squid.conf: 85 lines, 3075 characters.
#
```

### 3 – Изменение конфигурации Squid

Добавлен `visible_hostname`. Это обеспечивает корректную работу кэша и идентификацию сервера.

```
# cd /usr
# cd local/
# cd etc/
# ls
bash_completion.d      periodic                pkg.conf.sample       squid
man.d                  pkg.conf               rc.d
# cd squid
# mkdir -p ./var/cache
# mkdir -p ./var/logs
# chmod 777 ./var/logs
# chmod 777 ./var/cache
#
```

#### 4 – Создание директорий для логов и кэша

Команды `mkdir` и `chmod` подготавливают систему к работе Squid. Без этих директорий сервер не сможет вести логи и сохранять кэш. Это часть настройки ПО.

```
periodic                rc.d
# cd ../
# ls
bin      etc      include lib      libdata libexec sbin      share
# pwd
/usr/local
# cd sbin
# ls
pkg      pkg-static  purge      squid      squidclie
# cd ../
# pwd
/usr/local
# /usr/local/squid/sbin/squid -z
su: /usr/local/squid/sbin/squid: not found
# /usr/local/sbin/squid -z
2025/03/25 00:58:46! Processing Configuration File: /usr/local/etc/squid/
onf (depth 0)
# 2025/03/25 00:58:46! Created PID file (/var/run/squid/squid.pid)
2025/03/25 00:58:46 kid1! Processing Configuration File: /usr/local/etc/s
uid.conf (depth 0)
2025/03/25 00:58:46 kid1! Set Current Directory to /var/squid/cache
2025/03/25 00:58:46 kid1! Creating missing swap directories
2025/03/25 00:58:46 kid1! No cache_dir stores are configured.
2025/03/25 00:58:46! Removing PID file (/var/run/squid/squid.pid)
#
```

#### 5 – Инициализация кэша Squid

Запуск `squid -z` инициализирует директории хранения кэша, что необходимо для нормального функционирования сервиса.

```
# squid
# ps aux | grep squid
root 1301  0.0  0.4 100332 17276 -  Is  21:59   0:00.00 squid
squid 1303  0.0  0.9 204764 38068 -  S  21:59   0:00.06 (squid-1) --kid squid-1 (squid)
squid 1304  0.0  0.2 21312  8280 -  I  21:59   0:00.01 (logfile-daemon) /var/log/squid/access.log (log_file_daemon)
squid 1305  0.0  0.2 22152  9152 -  S  21:59   0:00.01 (pinger) (pinger)
root 1318  0.0  0.1 12716  2172 v0  S+  22:00   0:00.00 grep squid
# sockstat -4 -6 | grep squid
squid  pinger    1305 0    udp6    ::1:54223    ::1:27620
squid  pinger    1305 1    udp6    ::1:54223    ::1:27620
squid  squid      1303 7    udp46   *:46067      *:
squid  squid      1303 8    udp4    *:63659      *:
squid  squid      1303 11   tcp46   *:3128       *:
squid  squid      1303 13   udp6    ::1:27620    ::1:54223
```

#### 6 – Проверка процессов и портов Squid

Команды `ps aux` и `sockstat` подтверждают, что служба Squid работает и слушает нужные порты.

```
# printf "HTTP/1.1 200 OK\r\nContent-Length: 13\r\n\r\nHello, world!\n" | nc -l 8080 &
# env http_proxy="http://127.0.0.1:3128" fetch http://127.0.0.1:8080
GET / HTTP/1.1
Host: 127.0.0.1:8080
Accept: */*
User-Agent: fetch libfetch/2.0
Via: 1.1 minazuki (squid/6.12)
X-Forwarded-For: 127.0.0.1
Cache-Control: max-age=259200
Connection: keep-alive

127.0.0.1:8080          13 B   77 kBps   00s
[1] Done               printf HTTP/1.1 200 OK\r\nContent-Length: 13\r\n\r\nHello, world!\n | nc -l 8080
# env http_proxy="http://127.0.0.1:3128" fetch https://www.google.com
fetch: https://www.google.com: size of remote file is not known
www.google.com         19 kB  190 kBps   00s
#
```

## 7 – Настройка прокси и проверка доступа к локальному серверу

Через `http_proxy` и `fetch` проверяется возможность выполнения запроса через Squid.

```
# vi /etc/resolv.conf
```

## 8 – Редактирование `/etc/resolv.conf`

Редактирование файла `/etc/resolv.conf` вручную позволяет внести нужные DNS-серверы и проверить работу резолвера.

```
# Generated by resolvconf
search study.local
nameserver 127.0.0.1
nameserver 8.8.8.8
nameserver 8.8.4.4
~
```

## 9 – Файл `/etc/resolv.conf`

DNS-клиент теперь настроен на локальный сервер (127.0.0.1), а также резервные DNS Google.

```
# pkg search bind
R-cran-abind-1.4.8          Combine Multidimensional Arrays
R-cran-bindr-0.1.2          Parametrized Active Bindings
R-cran-bindr-0.2.3          Rcpp Interface to Active Bindings
bind-tools-9.20.6           Command line tools from BIND: delv, dig, host, nslookup...
bind9-devel-9.21.5          BIND DNS suite with updated DNSSEC and DNS64
bind918-9.18.34             BIND DNS suite with updated DNSSEC and DNS64
bind920-9.20.6              BIND DNS suite with updated DNSSEC and DNS64
bind_exporter-0.8.0_1       Prometheus exporter for BIND server statistics
bindgraph-0.3.1             RRDtool frontend for BIND statistics
bindtest-1.56.2             Test bind() semantics of IPv6 sockets
fusefs-bindfs-1.17.7_1     FUSE based bindfs implementation way more powerful than nullfs
glbinding-3.3.0             C++ binding for the OpenGL API
go-bindata-3.1.3.22         Generates Go code from any file
go-bindata-assetsfs-1.0.1_22 Serves embedded files from go-bindata with Go's net/http support
keybinder-0.3.1.6           Library for registering keyboard shortcuts
keybinder-gtk3-0.3.2_3     Library for registering keyboard shortcuts
libbind-6.0_2              Standard C resolver library
libindicator-12.10.1_2     Symbols and convenience functions for desktop indicators
luabind-0.9.1.17           Library that helps you create bindings between C++ and Lua
nanobind-2.4.0             Tiny and efficient C++/Python bindings (C++ version)
nsgenbind-0.9              NetSurf Generator for JavaScript bindings
p5-BIND-Conf_Parser-0.95_1  Perl5 module to parse BIND 8 configuration files
p5-BIND-Config-Parser-0.01_1 Parse BIND Config files
p5-Catalyst-Controller-BindLex-0.03_2 Stash your lexical goodness
p5-Data-Bind-0.30_2         Perl module to bind and alias variables
p5-Regexp-Bind-0.05_1       Bind variables to captured buffers
p5-Text-Bind-0.04_1         Bind Perl structures to text files
py311-argon2-cffi-bindings-21.2.0_1 Low-level CFFI bindings for Argon2
py311-nanobind-2.4.0        Tiny and efficient C++/Python bindings
py311-pybind11-2.13.6       Seamless interoperability between C++11 and Python
pybind11-2.13.6             Seamless interoperability between C++11 and Python (cmake part)
pybind11-json-0.2.14        Using nlohmann::json with pybind11
pybind11_abseil-202402.0     Pybind11 bindings for the Abseil C++ Common Libraries
pybind11_protobuf-0.0.0.20240809_4 Pybind11 bindings for Google's Protocol Buffers
redland-bindings-1.0.17.1_4 Language bindings for the Redland package
rubygem-apipie-bindings-0.6.0 Ruby bindings for API calls that are documented with Apipie
rubygem-bindata-2.4.15      Ruby library to read and write binary file formats
rubygem-bindex-0.8.1        Bindings for your Ruby exceptions
rubygem-binding_of_caller-1.0.1 Retrieve the binding of a methods caller
rust-bindgen-cli-0.71.1     Generate Rust bindings from C (and some C++) code
rust-cbindgen-0.26.0_11     Generate C bindings from Rust code
socketbind-1.1              Library to bind applications on multihomed machines to specific IP address
vbindiff-3.0.b5_1           Visual binary diff, visually compare binary files
xapian-bindings-1.4.27       Bindings allowing Xapian to be used from various programming languages
xbindkeys-1.8.7_1           Allows you to launch shell commands under X with your keyboard
xkbind-2010.05.20           Minimal keyboard layout indicator program
#
```

## 10 – Поиск пакетов, связанных с bind

`pkg search bind` помогает найти подходящую версию BIND для установки. Это часть подготовки по настройке DNS-сервера.

```
# pkg install bind918
```

## 11 – Установка BIND

Установка пакета `bind918` позволяет начать настройку кэширующего DNS-сервера в FreeBSD.

```
# vi /etc/rc.conf
```

## 12 – Редактирование /etc/rc.conf

Открытие `rc.conf` для добавления параметров автозапуска `named`. Это обеспечит запуск DNS-сервера при старте ОС.

```
hostname="fadeyjo"
ifconfig_em0="DHCP"
sshd_enable="YES"
ntpd_enable="YES"
moused_nondefault_enable="NO"
# Set dumpdev to "AUTO" to enable crash dumps, "NO" to disable
dumpdev="AUTO"
named_enable="YES"
```

## 13 – Содержимое /etc/rc.conf

Добавлена строка `named_enable="YES"` — это ключевой параметр для включения службы DNS на старте.

```
// Refer to the named.conf(5) and named(8) man pages, and the documentation
// in /usr/local/share/doc/bind for more details.
//
// If you are going to set up an authoritative server, make sure you
// understand the hairy details of how DNS works. Even with
// simple mistakes, you can break connectivity for affected parties,
// or cause huge amounts of useless Internet traffic.

options {
    // All file and path names are relative to the chroot directory,
    // if any, and should be fully qualified.
    directory      "/usr/local/etc/namedb/working";
    pid-file       "/var/run/named/pid";
    dump-file      "/var/dump/named_dump.db";
    statistics-file "/var/stats/named.stats";

// If named is being used only as a local resolver, this is a safe default.
// For named to be accessible to the network, comment this option, specify
// the proper IP address, or delete this option.
    listen-on      { any; };

// If you have IPv6 enabled on this system, uncomment this option for
// use as a local resolver. To give access to the network, specify
// an IPv6 address, or the keyword "any".
//    listen-on-v6   { ::1; };

// These zones are already covered by the empty zones listed below.
// If you remove the related empty zones below, comment these lines out.
    disable-empty-zone "255.255.255.255.IN-ADDR.ARPA";
    disable-empty-zone "0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.IP6.ARPA";
    disable-empty-zone "1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.IP6.ARPA";

// If you've got a DNS server around at your upstream provider, enter
// its IP address here, and enable the line below. This will make you
// benefit from its cache, thus reduce overall DNS traffic in the Internet.
/*
    forwarders {
        8.8.8.8;
        8.8.4.4;
    };
*/
```

## 14 – Конфигурация BIND (named.conf)

Настройка конфигурационного файла DNS-сервера: указаны пути к логам, настройка прослушивания, а также DNS-forwarding на Google.





```

study.local.      IN NS    ns1.study.local.
study.local.      IN NS    ns2.study.local.

ns1               IN A     192.168.1.10
ns2               IN A     192.168.1.11

study.local.      IN A     192.168.1.20

pc1               IN A     192.168.1.21

www               IN CNAME study.local.

```

## 192.168.1.rev:

```

$TTL 3600
1.168.192.in-addr.arpa.  IN SOA ns1.study.local. admin.study.local. (
                                2025032401
                                3600
                                900
                                604800
                                3600 )

                                IN NS    ns1.study.local.
                                IN NS    ns2.study.local.

10  IN PTR    ns1.study.local.
11  IN PTR    ns2.study.local.
20  IN PTR    study.local.
21  IN PTR    pc1.study.local.

```

```

# pwd
/usr/local/etc/namedb/master
# cd ..
# ls
bind.keys          master          named.conf.sample  primary          secondary
dynamic            named.conf      named.root         rndc.key         working
# named-checkconf named.conf
# cd master/
# named-checkzone study.local study.local
zone study.local/IN: loaded serial 2025032401
OK
# named-checkzone 1.168.192.in-addr.arpa 192.168.1.rev
zone 1.168.192.in-addr.arpa/IN: loaded serial 2025032401
OK
# service named restart
Stopping named.
Waiting for PID5: 1700.
Starting named.
# service named status
named is running as pid 1829.
#

```

## 18 – Проверка зоны и перезапуск службы

Проверка корректности zone-файлов через named-checkzone и перезапуск службы.



```
# dig vk.com

; <<>> DiG 9.20.6 <<>> vk.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 16995
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 0182ec031bb04f1e0100000067e2204422387938c24f436f (good)
;; QUESTION SECTION:
;vk.com.                                IN      A

;; Query time: 1980 msec
;; SERVER: 127.0.0.1#53(127.0.0.1) (UDP)
;; WHEN: Tue Mar 25 06:17:24 MSK 2025
;; MSG SIZE rcvd: 69

# dig vk.com

; <<>> DiG 9.20.6 <<>> vk.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 10493
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 4ccdb103854b20120100000067e2204aacc37353873ed2da (good)
;; QUESTION SECTION:
;vk.com.                                IN      A

;; Query time: 59 msec
;; SERVER: 127.0.0.1#53(127.0.0.1) (UDP)
;; WHEN: Tue Mar 25 06:17:30 MSK 2025
;; MSG SIZE rcvd: 69

#
```

21 – Команда dig для проверки того, что сервер кеширует запросы (проверяется тем, что второй запрос отправился намного быстрее 1980 ms → 59 ms, за счет кешируемого сервера)

```
# cd /var
# ls
account      authpf      crash       empty       log         preserve    spool       unbound
at           backups    cron        games       mail        run         squid       yp
audit        cache      db          heimdal     msgs        rwho       tmp
# mkdir -p dump
# chown bind:bind /var/dump/
# rndc dumpdb
#
```

22 – Создание дампа DNS и просмотр /var  
Создание директории /var/dump и дампа DNS через rndc dumpdb. Это дополнительная диагностика для анализа содержимого DNS-сервера.

```

vk.com.      172539  NS      ns1.vk.com.
             172539  NS      ns2.vk.com.
             172539  NS      ns3.vk.com.
             172539  NS      ns4.vk.com.
; pending-answer
             640    A      87.240.129.133
             640    A      87.240.132.67
             640    A      87.240.132.72
             640    A      87.240.132.78
             640    A      87.240.137.164
             640    A      93.186.225.194
; glue
ns1.vk.com.  172539  A      87.240.131.131
; glue
ns2.vk.com.  172539  A      95.213.21.21
; glue
ns3.vk.com.  172539  A      93.186.238.238
; glue
ns4.vk.com.  172539  A      87.240.136.136
; glue
a.gtld-servers.net. 171756  A      192.5.6.30
; glue
             171756  AAAA   2001:503:a83e::2:30
; glue
b.gtld-servers.net. 171756  A      192.33.14.30
Search wrapped

```

23 – DNS-записи vk.com (успешный вывод)

Пример успешного ответа DNS — содержимое зоны vk.com. Это финальная проверка работы резолвера и сервера.

```

# host pc1.study.local
pc1.study.local has address 192.168.1.21
# host 192.168.1.21
21.1.168.192.in-addr.arpa domain name pointer pc1.study.local.
# host vk.com
vk.com has address 87.240.129.133
vk.com has address 87.240.132.72
vk.com has address 87.240.137.164
vk.com has address 93.186.225.194
vk.com has address 87.240.132.67
vk.com has address 87.240.132.78
vk.com mail is handled by 0 mxs.mail.ru.
#

```

24 - Результат выполнения команды host в FreeBSD.

Эта команда используется для DNS-запросов (разрешения имён)

Производится прямой DNS-запрос: запрашивается IP-адрес, соответствующий имени pc1.study.local (зона study.local настроена правильно, и DNS-сервер вернул IP-адрес 192.168.1.21)

Обратный DNS-запрос (reverse lookup): проверяется, какое доменное имя соответствует IP-адресу 192.168.1.21 (подтверждает корректную настройку обратной зоны DNS. Сервер возвращает имя `pc1.study.local` для указанного IP.)

Запрос к внешнему DNS-серверу: какие IP-адреса (A-записи) и почтовые серверы (MX-записи) зарегистрированы у домена `vk.com`. (подтверждает, что DNS-клиент в системе корректно работает и может разрешать внешние домены, что важно для проверки работоспособности после настройки резолвера (`resolv.conf`) и сервера BIND.)

**Вывод:** в ходе лабораторной работы были получены практические навыки по установке и настройке прокси-сервера Squid и системы DNS в ОС FreeBSD. Успешно настроены резолвер, кэширующий DNS-сервер и зоны отображения, что подтвердило работоспособность всех компонентов.

### **Ответы на контрольные вопросы**

#### **1. Что такое сетевой интерфейс?**

Сетевой интерфейс — это программно-аппаратная точка взаимодействия устройства с сетью, обеспечивающая обмен данными между операционной системой и сетевой инфраструктурой.

#### **2. Что такое виртуальный сервер?**

Виртуальный сервер — это программная реализация сервера, работающая на физическом оборудовании и обеспечивающая изоляцию ресурсов для выполнения отдельных сервисов или приложений.

#### **3. Какая утилита позволяет читать и изменять настройки сетевых интерфейсов?**

Утилита `ifconfig` позволяет читать и изменять настройки сетевых интерфейсов.

#### **4. Что такое «синоним» (alias) сетевого интерфейса?**

Синоним (alias) сетевого интерфейса — это дополнительный IP-адрес, привязанный к существующему сетевому интерфейсу для обслуживания нескольких сетей или сервисов через одно физическое устройство.

#### **5. Что такое DHCP?**

DHCP (Dynamic Host Configuration Protocol) — это протокол автоматической выдачи сетевых настроек (IP-адрес, шлюз, DNS) устройствам в сети.

#### **6. Как настраивать DHCP сервер?**

Для настройки DHCP сервера необходимо установить соответствующее программное обеспечение (например, `isc-dhcp-server`), настроить

конфигурационный файл с параметрами сети (например, диапазон IP-адресов, шлюз, DNS) и запустить службу.

### **7. Как настраивать DHCP клиент?**

Для настройки DHCP клиента нужно запустить клиентскую утилиту (например, `dhclient` в FreeBSD), которая автоматически запросит и получит сетевые параметры от DHCP сервера.

### **8. Что такое прокси-сервер?**

Прокси-сервер — это сервер, который принимает запросы от клиентов, пересылает их к целевым ресурсам и передает полученные ответы обратно клиентам.

### **9. В каких целях применяются прокси-серверы?**

Прокси-серверы применяются для повышения безопасности, анонимизации пользователей, фильтрации контента, кэширования данных и оптимизации сетевого трафика.

### **10. Какие преимущества в работе в сети Интернет с прокси?**

Преимущества: повышение скорости доступа за счет кэширования, повышение конфиденциальности, возможность обхода ограничений, фильтрация нежелательного контента, снижение нагрузки на внешние каналы связи.

### **11. Какие существуют виды прокси?**

Существуют: HTTP-прокси, HTTPS-прокси, FTP-прокси, Socks-прокси, кэш-прокси, прозрачные прокси, анонимные прокси.

### **12. Что такое HTTP-прокси? Какими возможностями обладает?**

HTTP-прокси — это прокси, работающий с HTTP-трафиком. Возможности: кэширование веб-страниц, фильтрация контента, контроль доступа по URL, анонимизация пользователей.

### **13. Что такое FTP-прокси?**

FTP-прокси — это прокси-сервер, предназначенный для пересылки FTP-запросов, обеспечивающий контроль и безопасность передачи файлов через протокол FTP.

### **14. Что такое HTTPS-прокси? В чем отличие от HTTP-прокси?**

HTTPS-прокси — это прокси для безопасного HTTPS-трафика. Отличие в том, что HTTPS-прокси поддерживает шифрование данных через SSL/TLS, обеспечивая конфиденциальность передаваемой информации.

### **15. Что такое Mapping-прокси?**

Mapping-прокси — это прокси-сервер, выполняющий преобразование одного адреса запроса в другой, обеспечивая маршрутизацию трафика согласно заданным правилам.

### **16. Что такое Socks-прокси?**

Socks-прокси — это универсальный прокси-сервер, работающий на уровне транспортного протокола (TCP/UDP) и пересылающий любой тип трафика без анализа содержимого.

### **17. Что такое кэширование?**

Кэширование — это процесс временного хранения копий часто запрашиваемых данных для ускорения доступа к ним в будущем.

### **18. Что такое сетевые кэши? Для чего они нужны?**

Сетевые кэши — это устройства или сервисы, сохраняющие сетевые данные для ускорения доступа пользователей к популярным ресурсам и снижения нагрузки на внешние каналы связи.

### **19. Что такое ICP? HTCP?**

ICP (Internet Cache Protocol) и HTCP (Hypertext Caching Protocol) — это протоколы, предназначенные для обмена информацией между кэш-серверами о наличии закэшированных объектов с целью оптимизации распределения запросов.

### **20. В чем отличие кэш-сервера от прокси-сервера?**

Кэш-сервер основное внимание уделяет сохранению и выдаче ранее запрашиваемого контента, а прокси-сервер занимается пересылкой запросов и может включать кэширование как дополнительную функцию.

### **21. Зачем нужен кэш-сервер?**

Кэш-сервер нужен для ускорения загрузки контента, снижения нагрузки на внешние каналы, увеличения доступности ресурсов и экономии трафика.

### **22. Что такое прокси-кэш-сервер? Как это работает?**

Прокси-кэш-сервер — это прокси-сервер с функцией кэширования, который сохраняет передаваемые через него данные и при повторных запросах выдает их из локального хранилища без обращения к внешнему ресурсу.

### **23. Что значит прозрачное кэширование?**

Прозрачное кэширование — это метод, при котором трафик пользователей автоматически перенаправляется через кэш-сервер без необходимости настройки клиентов.

**24. Какие существуют архитектуры (модели) прокси-кэш-сервера? В чем их суть?**

- Индивидуальная модель: каждый сервер обслуживает свои запросы отдельно.
- Иерархическая модель: кэш-серверы организованы в дерево, запросы передаются вверх по иерархии.
- Распределенная модель: кэш-серверы работают совместно, обмениваясь данными.

**25. Как можно классифицировать прокси-кэш-серверы, представленные на рынке?**

По способу работы (прозрачные, непрозрачные), по уровню интеграции (аппаратные, программные решения), по функционалу (универсальные, специализированные).

**26. Какие фирмы занимаются разработкой и производством прокси-кэш-серверов? Чем характеризуется их продукция?**

Компании: Blue Coat, Cisco, Squid (опенсорс). Продукция характеризуется масштабируемостью, поддержкой высоких нагрузок, расширенными функциями безопасности и аналитики.

**27. На что стоит обратить внимание при покупке прокси-кэш-сервера?**

На производительность, объем доступного кэша, поддерживаемые протоколы, возможность интеграции с сетевой инфраструктурой и наличие функций безопасности.

**28. Что такое активное и пассивное кэширование? В чем отличие?**

Активное кэширование — это предзагрузка данных на сервер по расписанию или предсказанию спроса.

Пассивное кэширование — это кэширование только тех данных, которые реально запрашивают пользователи.

**29. Каким образом размещают прокси-кэш-серверы?**

Размещение осуществляется: на границе сети (между локальной сетью и Интернетом), внутри корпоративной сети или в распределенной архитектуре с несколькими точками.

**30. Что такое каскадная настройка прокси-серверов? Объясните каскадный режим работы прокси-сервера.**

Каскадная настройка — это последовательное соединение нескольких прокси-серверов, через которые проходит трафик. Каскадный режим обеспечивает дополнительную фильтрацию, анонимизацию и балансировку нагрузки.

### **31. Что такое DNS?**

DNS (Domain Name System) — это система, преобразующая символьные доменные имена в IP-адреса и обратно.

### **32. По каким причинам может понадобиться сервер имен?**

Для локального разрешения доменных имен, ускорения работы сети, обеспечения доступности внутренних ресурсов и повышения отказоустойчивости.

### **33. Какая программа в ОС FreeBSD отвечает за работу системы DNS?**

Программа BIND (Berkeley Internet Name Domain) отвечает за работу DNS в FreeBSD.

### **34. Как запустить BIND? Что это такое?**

BIND — это серверная программа для реализации службы DNS. Чтобы запустить BIND, необходимо установить его пакет, настроить конфигурационные файлы (named.conf) и запустить службу (service named start).

### **35. Какой файл используется для настройки DNS клиента?**

Файл /etc/resolv.conf используется для настройки DNS клиента в FreeBSD.

### **36. Что такое зона в понятии DNS? Типы зон.**

Зона DNS — это часть пространства имен, управляемая отдельным администратором. Типы зон:

- Зона прямого разрешения (Forward Zone)
- Зона обратного разрешения (Reverse Zone)

### **37. Когда используется кэширующий сервер имен?**

Кэширующий сервер имен используется для уменьшения времени ответа на DNS-запросы и снижения нагрузки на внешние DNS-серверы.

### **38. Какая программа используется для управления сервером имен?**

Для управления сервером имен используется утилита rndc (Remote Name Daemon Control) в связке с BIND.