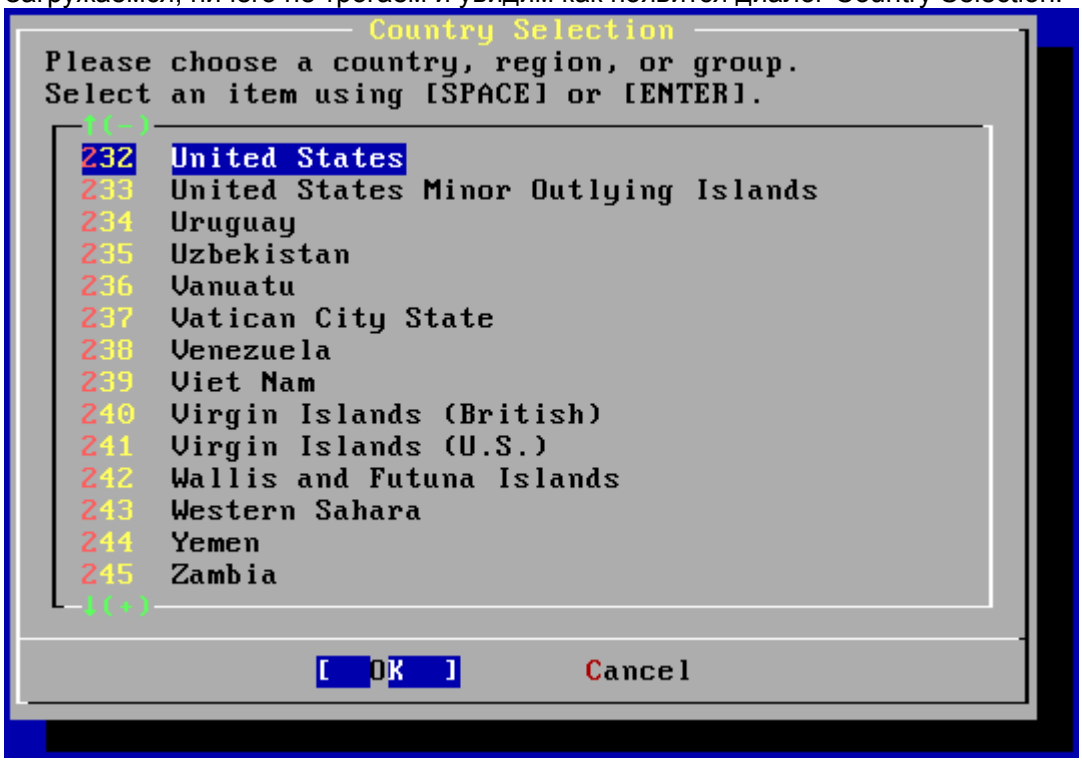
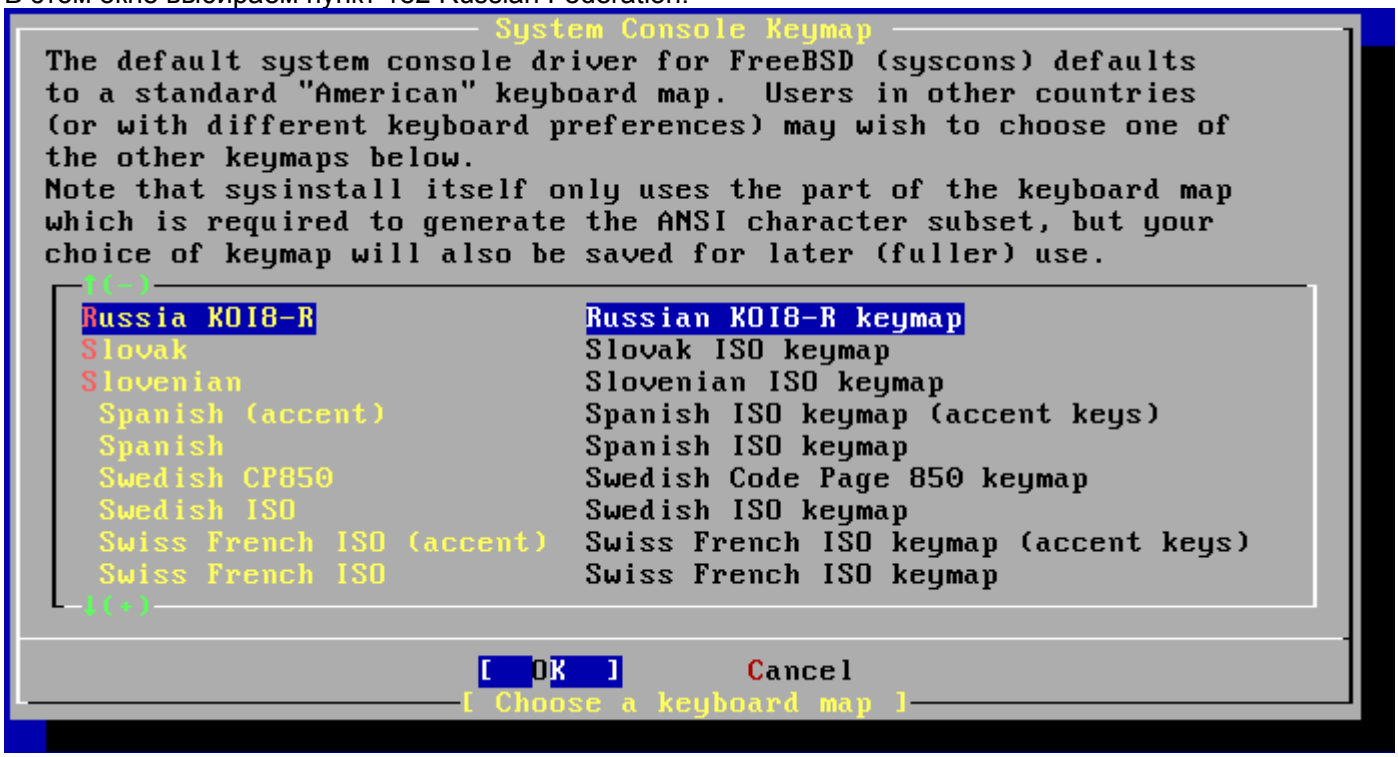
Установка freebsd

Статей таких конечно много, но решил и тут запостить, как ставлю FreeBSD я. Установка будет минимальая, ибо на серверах чем меньше всякой лабуды, тем лучше. Начнём с того что вам нужен всего лишь один первый диск, взять который можно на оффициальном сайте. Буду ставить с диска 7.1-RELEASE-i386-disc1, поэтому имеено с него и загружайтесь, поставив в БИОСе загрузку с привода cd-rom.

Загружаемся, ничего не трогаем и увидим как появится диалог Country Selection:

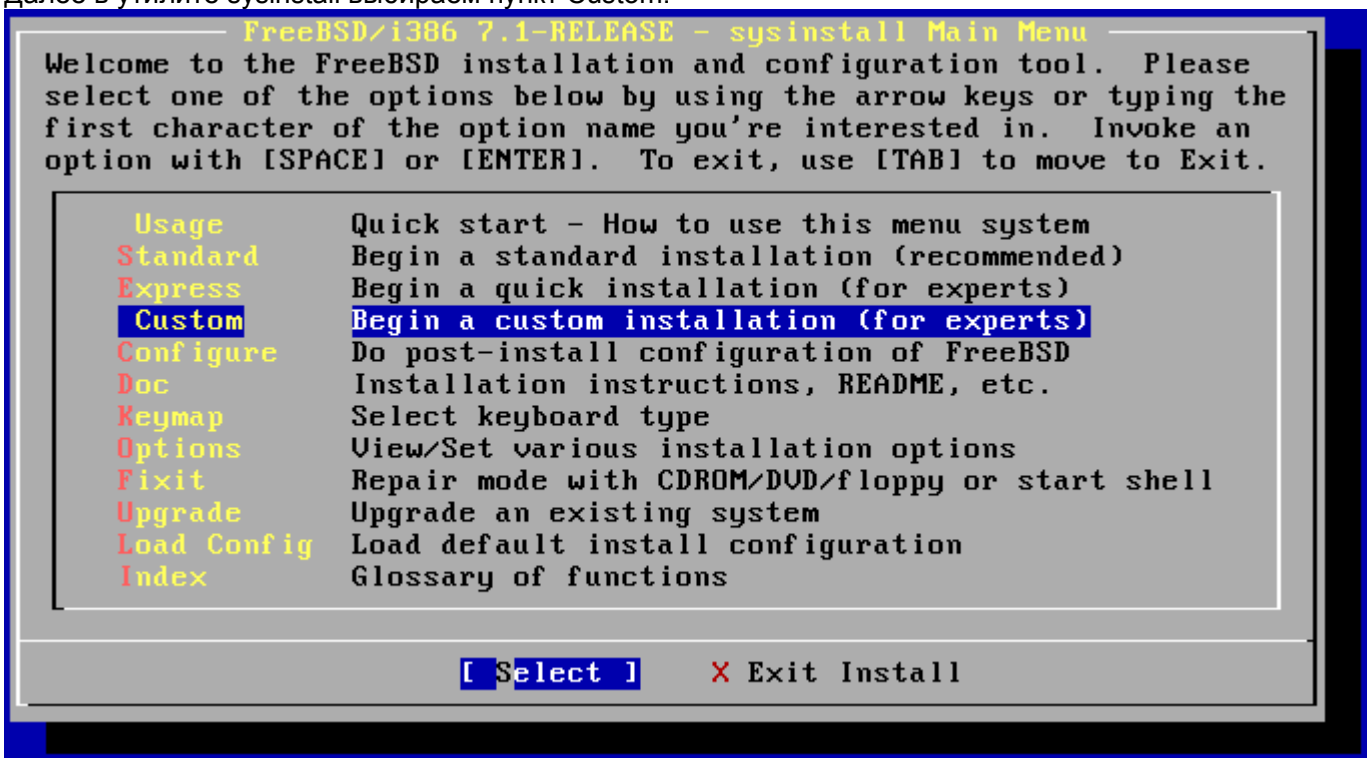


В этом окне выбираем пункт 182 Russian Federation.

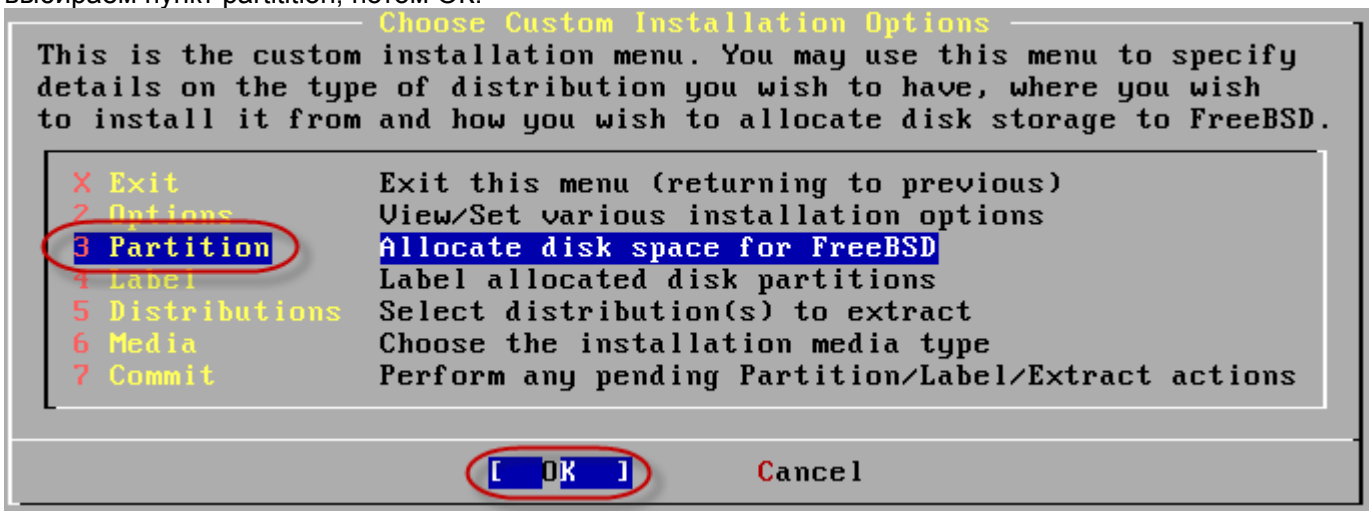


в появившемся окне выбираем пункт – russia KOI8-R

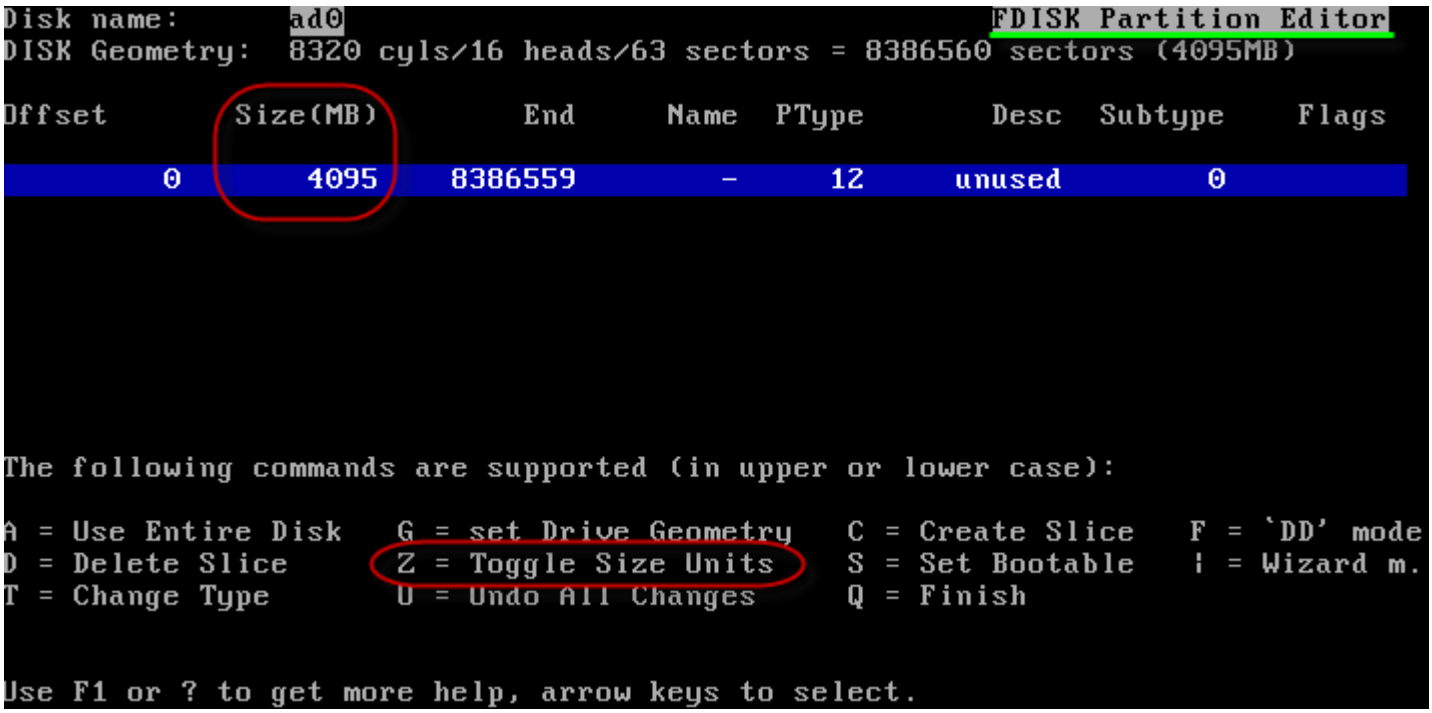
Далее в утилите sysinstall выбираем пункт Custom:



выбираем пункт partitition, потом OK:



Появится чёрное окно FDISK Partitition Editor, в котором мы и формируем слайсы (аналог разделов в m$). Размеры указаны в секторах, но нам удобнее видеть мегабайты, поэтому 2-а раза нажмём букву “z”.



Обьём винчестера, как видно на изображении выше, = 4Гб, их все и будем использовать.

Итак:

Создать слайс – “C”

Удалить слайс – “D”

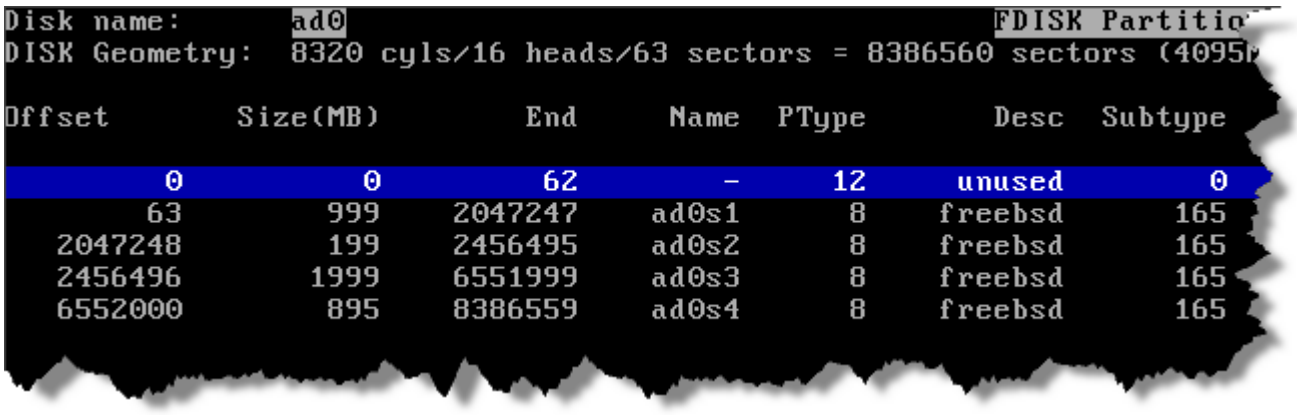
Закончить разбивку – “Q”

Размер слайсов должен зависеть от того, для чего вы ставите FreeBSD, если Вам всё равно, жмите «A». Мы разобьём в ручную следующим образом:

точка монтирования | размер / 1000 Mb swap 200 Mb /usr 2000 Mb /var остальное

Для просмотра скрытой части статьи вы должны авторизироваться [Login] или зарегистрироваться [Register] на сайте.

Должно получится что то вроде:



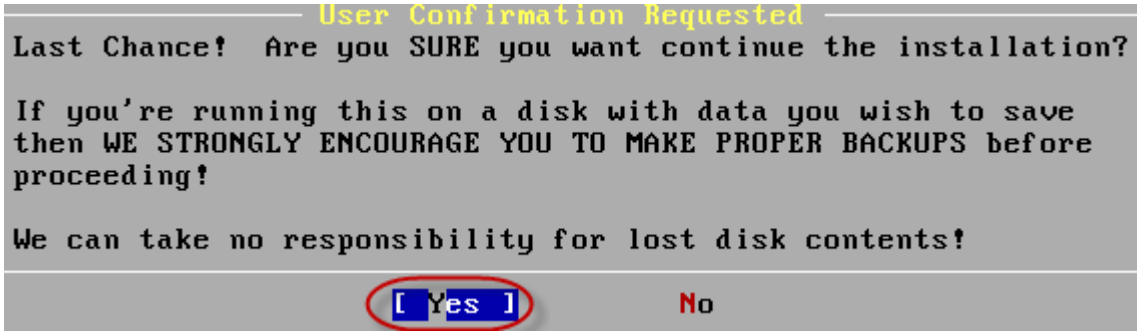
Заканчиваем разбивку нажатием на «Q» и во всплывшем окне выбираем пункт «BootMgr» -> «Install the FreeBSD Boot Manager», жмём «OK». Переходим к меню Label для формирования точек монтирования, появляется окно freebsd disklabel editor, там будем формировать наши точки монтирования, а имено точки: / , /usr , /var и swap . Тут тоже самое, что и со слайсами: Создать – “C”, Удалить – “D”, Закончить разбивку – “Q”.

Ставим выделение на первый слайс, который у нас получился 999MB (см 2-ой столбик скриншота выше), жмём «C», жмём «OK», выбираем пункт «FS A file system», опять «OK», в поле вписываем

посто слеш, так называемый корень / и снова «ОК». Со слайсом в 199Мб длаем тоже самое, только указываем не «FS A file system», а »Swap A swap partitition» (это подкачка). Со слайсом в 1999Мб делаем тоже самое, что и с первым слайсом, а именно выделяем его, жмём «С», жмём «ОК», выбираем пункт «FS A file system», опять «ОК», в поле вписываем /usr и снова «ОК». И последний слайс, размер которго получился 895Мб (это тот слайс, который мы решили, сколько останется столько и будем по обьёму). С ним всё тоже самое, что и с предыдущим только вместо /usr вписываем /var .Должно получится что то вроде этого:

И вновь завершаем всё нажатием кнопки «Q».

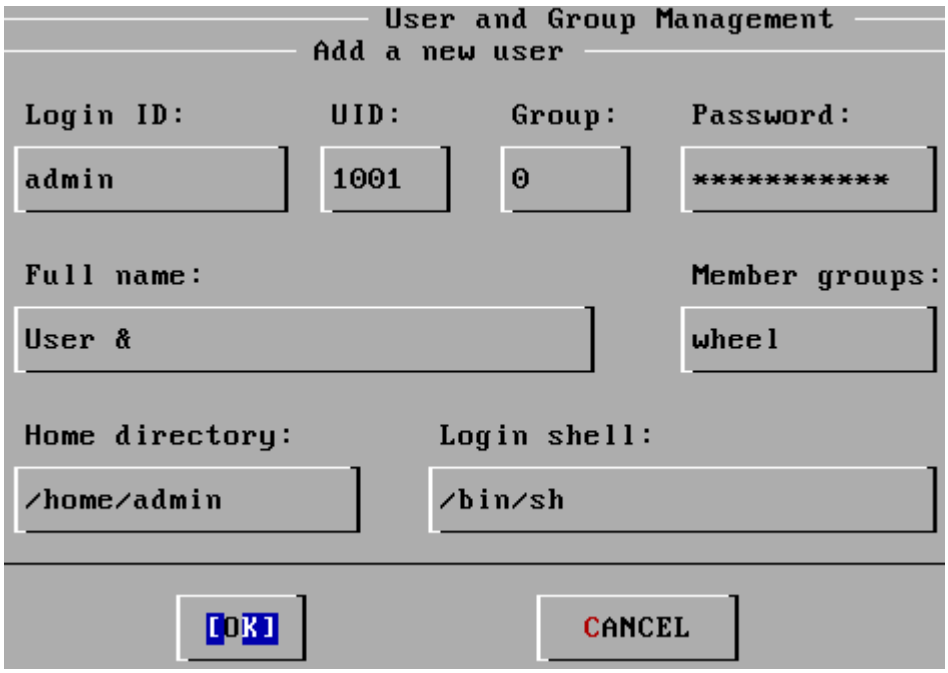
«ОК» выбираем во всех меню пока не окажитесь в меню «choose custom installation options» Убеждаемся, что в «media» указан тот источник, с которого мы будем ставить нашу ОС, а имено cd и жмём «commit». Жмём «YES»:



Ждём пока всё поставится, как правило это не занимает много времени. По окончании, опять «YES» и появляемся в меню «freebsd configuration menu». В общем то FreeBSD мы поставили, сместе с коллекцией портов и исходниками ядра. Нужна первоначальная настройка операционной системы, именно эти и займёмся.

Идём в «root password» где вписываем пароль суперпользователя 2-а раза. Это самый важный пароль, держите его в секрете.

Далее идём в «User Management» , выбираем «User» и заполняем поля:



«OK»

Переходим в пункт Console:

В пункте FONT -> выбираем IBM866

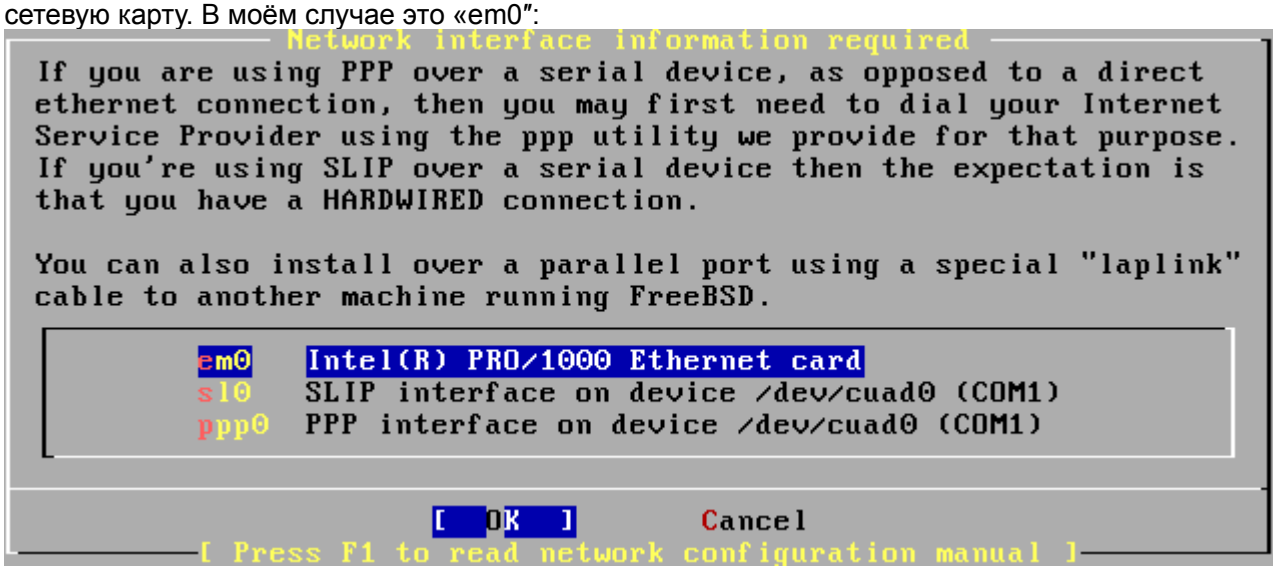
В пункте KEYMAP -> выбираем RUSSIA KOI8-R

В пункте screenmap -> выбираем koi8-r to ibm866

В пункте ttys -> выбираем const25r

<>

Теперь настроим сетевой интерфейс. Находясь в «FreeBSD Configuration Menu» выбираем «Networking», ставим крестик напротив «Interfaces», где выбираем наш интерфейс, т.е. нашу сетевую карту. В моём случае это «em0″:



Далее на вопрос с IPv6 жмём «No» , так же как и на вопрос о DHCP. Заполняем поля латинскими буквами. Учитываем, что ставим домашний сервер и вводиме данные, кроме IP, маски сети и шлюза не важны. В противном случае некоторые данные нужно уточнить у вашего провайдера.

хост – любое_имя

домен – любое название

IPV4 GETEWAY – IP адрес шлюза

name server – 217.24.112.2 (это IP адлес днс сервера, если знаете DNS физически поближе к вам и с лучшей связью, пишите его)

ipv4 address: IP адрес для сетевой карты

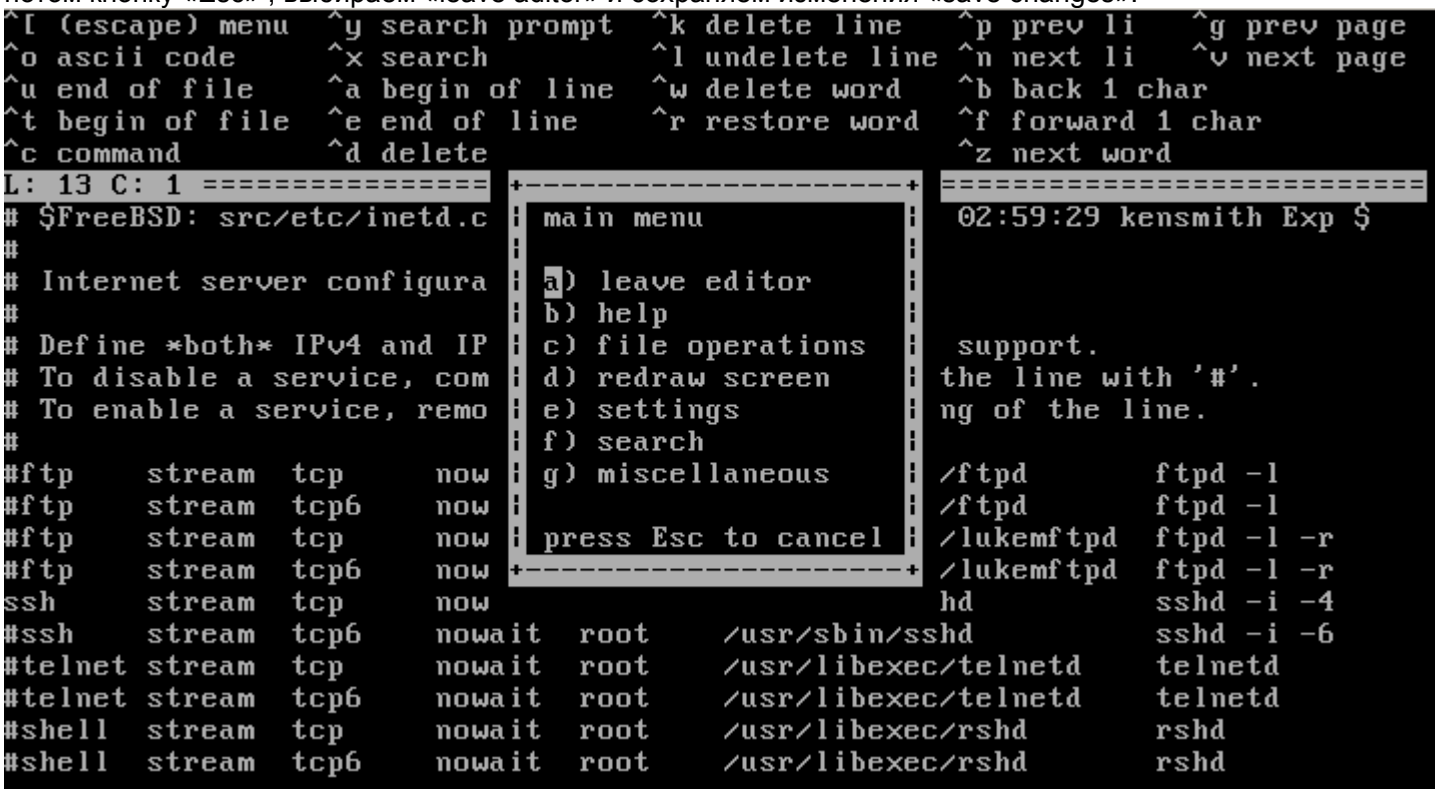
netmask: мадка сети, спишем 255.255.255.0

Остальные поля оставим пустыми и жмём «ОК»

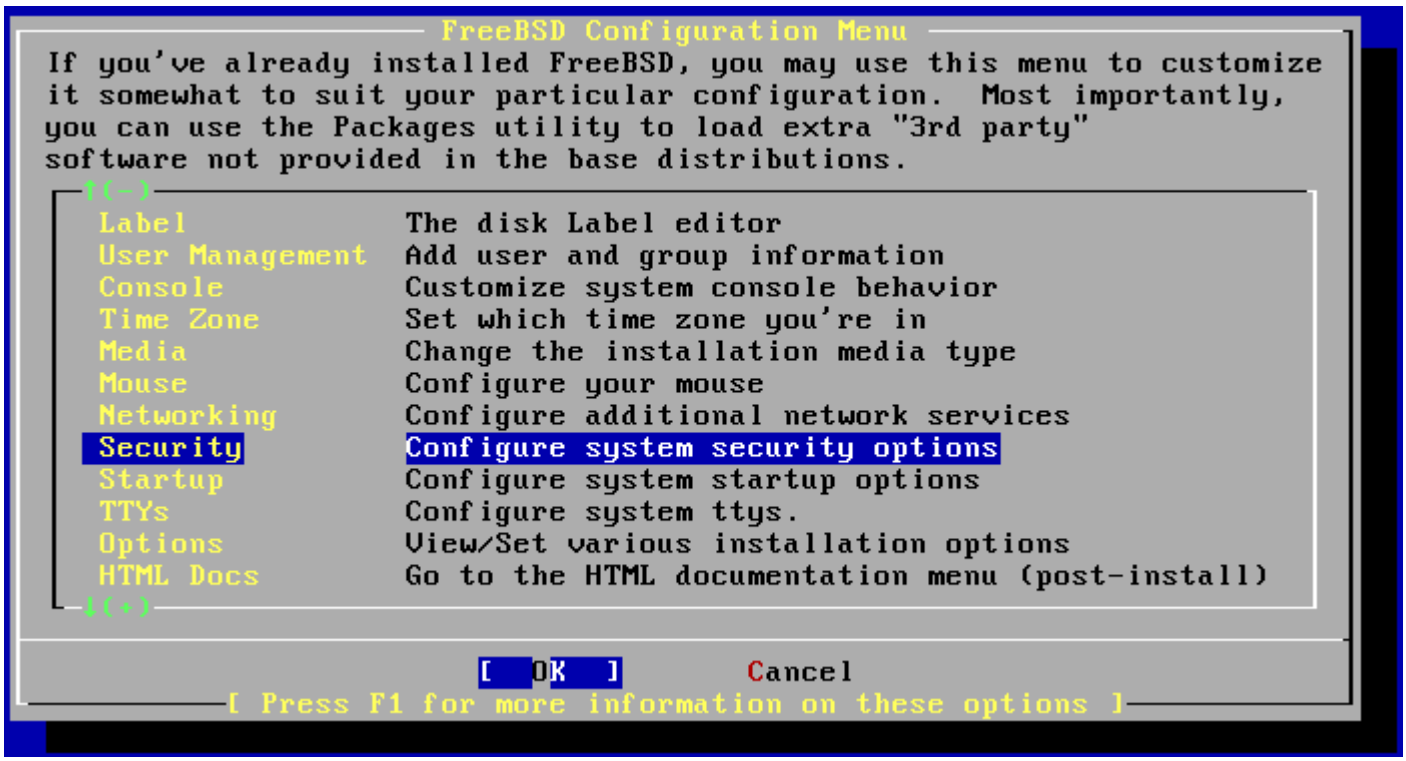
Вернувшись в окно «network interface information required» ставим крестики напротив AMD, inetd (два раза жмём yes); раскоментируем строчку:

ssh tcp

потом кнопку «Esc» , выбираем «leave aditor» и сохраняем изменения «save changes».



Возвращаемся в «FreeBSD Configuration Menu», где выбираем «security»:



Переходим в «securelevel» выбираем «disabled» и «exit». Настройка завершена, жмём опять «exit», «exit».

«exit install» выбираем «yes».

Установка и первоначальная настройка завершена. Перезагружаемся, вытаскиваем диск и всё готово, радуемся свежеустановленной FreeBSD. Думаю статья пригодится начинающим пользователям этой ОС и тем, кто хочет только ради интереса посмотреть на неё. Дерзайте

Простой способ раздачи интернета в сети

Задача довольно тривиальная: сервер на FreeBSD имеет 2-е сетевые карты; в одну вставлен RJ-45 от ADSL-модема, от второй сетевой карты кабель идёт в коммутатор, а уже посредством коммутатора сеть получают остальные ПК. Сервер FreeBSD соединяется с интернетом по PPPOE и будет раздавать его остальным. Реализовывать раздачу будем nat'ом. Опишу почти самый простой способ.

Сначала перекомпилим ядро, не пугайтесь, это не сложно, нем более мы лишь добавим туда несколько нужных параметров, а не будем пересматривать всё содержимое конфигурационного файла (это отдельная история).

1) Проверим, присутствуют ли тексты ядра:

```
1 ls /usr/src/sys
```

Если такого каталога нет, значит вставляем диск №1 с freebsd в cd-rom на сервере,

заходим в систему под root'ом:

```
1 su -l
2 Password:
```

запускаем:

```
1 sysinstall
```

выбраем Configure, далее Distributions, потом src, потом base и sys.

Жмём везде «ОК», ждём пока не спросят откуда устанавливать, выбираем CD/DVD

и ждём пока скопируются исходники.

Тем, кто устанавливал FreeBSD по этой инструкции, пункт 1 можно пропустить.

2) Тексты ядра скопировались, далее:

```
1 cd /usr/src/sys/i386/conf
2 cp GENERIC mykern
3 ee /usr/src/sys/i386/conf/mykern
```

Меняем строчку:

```
1 ident GENERIC
```

на:

```
1 ident mykern
```

и в конце файлика mykern, который мы редактируем, добавляем строки

Для просмотра скрытой части статьи вы должны авторизироваться [Login] или зарегистрироваться [Register] на сайте.

После этого делаем:

```
1 cd /usr/src
2 make buildkernel KERNCONF=mykern
3 make installkernel KERNCONF=mykern
```

ВНИМАНИЕ: Если вы выполнили успешно 2-е последние строчки и находитесь физически далеко от сервера, не перезагружайтесь, иначе фаервол начнёт действовать и блокировать все пакеты (выполняется правило по умолчанию)

3) Далее подредактируем:

```
ee /etc/rc.conf
```

Проверим наличие строк:

```
gateway_enable="YES"
natd_enable="YES"
#re0-сетевой интерфейс, к которому подключён ADSL (измените на свой)
natd_interface="re0"
natd_flags="-dynamic -f /etc/natd.conf"
firewall_enable="YES"
#файл с правилами фаервола-rc.firewall
firewall_script="/etc/rc.firewall"
```

Если вышеуказанных строк нет(а их скорее всего нет), добавьте их.

4) Теперь:

```
ee /etc/natd.conf
```

Содержимое должно быть следующим (исправьте если это не так):

```
log yes
deny_incoming yes
verbose no
log_denied yes
same_ports yes
use_sockets yes
unregistered_only yes
```

5) Добавим пару простейших правил, лишь бы заработал необходимый минимум.

```
mv /etc/rc.firewall /etc/rc.firewall_old
cp /etc/rc.firewall_old /etc/rc.firewall
ee /etc/rc.firewall
```

Стираем там всё, зажав сочетание кнопок Ctrl+K

и впишим туда следующее

Для просмотра скрытой части статьи вы должны авторизироваться [Login] или зарегистрироваться [Register] на сайте.

7) Клиенты.

Когда с сервером всё завершено успешно, на клиентах в настройках сетевого интерфейса, в качестве шлюза надо прописать внутренний адрес нашего сервера (в данном примере это ip адрес re1). И IP адреса и маска подсети у клиентов должны быть из той же подсети, что и re1, а так же не забудьте прописать dns сервера (скорее всего их вам сообщил провайдер). От настройки клиентов можно избавиться, настроив один раз dhcp-сервер и указав клиентам, автоматически получать настройки сети.

— ниже не обязательно для прочтения —

для тех, кто хочет понять хотябы коротко, что мы сделали:

Впаяли в ядро фаервол, в атозагрузку добавили nad и фаервол, создали правила, разрешающие ssh, www, ping, dns, перезапустили сервер и настроили клиентов. Таким образом обеспечили раздачу интернета через freebsd сервер. Более подробное конфигурирование ядра и правил фаервола-это уже темы отдельных статей.

Как сделать мост (bridge) на FreeBSD

Очередная вполне стандартная для оффисов ситуация, когда между интернетом и локальной сетью ставим мост(bridge) на FreeBSD для блокировки сайтов, мониторинга сети, да и мало ли ещё чем можно озадачит сервер.

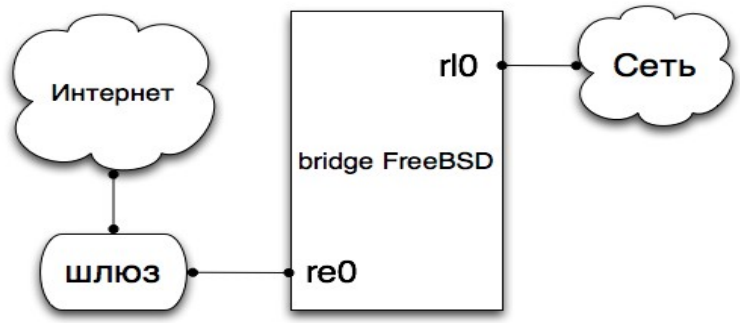
Мост на FreeBSD делается довольно просто, поэтому заметка будет короткой. В конце будет полезный бонус.

Итак действия будут разворачиватьс на:

```
1 uname -r -s
```

FreeBSD 9.0-RELEASE

Так же стоит 2-е сетевые карты (re0 и rl0), одна получает интернет, другая смотрит дальше в сеть. IP-адресА у двух этих сетевых карт находяться в одной подсети. Схематично это выглядит так:



Нам надо пересобрать ядро. Как это делается, написано здесь. Вам нужно выполнить там пункты 1 и 2 , но в пункте 2 в конфигурацию ядра добавить строку(ниже, ко всем там перечисленным):

```
1 device if_bridge
```

Это вобщем-то и всё, что нужно выполнить из статьи про простой способ раздачи интернета в сети.

После того как выполнили «make installkernel KERNCONF=…»,в /etc/rc.conf добавляем:

```
1 firewall_enable="YES"
2 firewall_script="/etc/rc.firewall"
3 cloned_interfaces="bridge0"
4 ifconfig_bridge0="addm re0 addm rl0 up"
```

здесь перечислены вышеописанные 2-е сетевые карты re0 и rl0.

Далее делаем:

mv /etc/rc.firewall /etc/rc.firewall_0

echo # > /etc/rc.firewall

файл /etc/rc.firewall пока что сделаем таким:

```
1 /sbin/ipfw -f flush
2 /sbin/ipfw -q add 00100 allow all from any to any via lo0
3 /sbin/ipfw -q add 00101 allow all from any to 127.0.0.0/8
4 /sbin/ipfw -q add 00102 allow all from 127.0.0.0/8 to any
5 /sbin/ipfw -q add 00103 allow all from any to any
```

и перегружаемся:

```
1 reboot
```

На этом всё, мост(физически железку) можно интегрировать в сеть и через него будет проходить весь интернет-трафик пользователей.

И вот тот самый бонус, а именно блокировка сайтов с помощью ipfw на FreeBSD. Тоесть довольно удобно никуда не ходить, никому ничего в браузеры не прописывать, а рубить сидя на месте.

dhcp-сервер на FreeBSD

dhcp (протокол динамической конфигурации узла) может понадобится как дома так и на предприятии, чаще конечно на предприятии, ибо ходить по всем компьютерам и настраивать сетевое подключение вручную не очень увеселительное мероприятие. Именно для раздачи сетевых настроек я и поднял dhcp-сервер. А теперь о том, как сделать dhcp-сервер на FreeBSD.

```
cd /usr/ports/net/isc-dhcp3-server
make install clean
```

В появившемся окне я оставил галочки напротив dhcp_paranoia, dhcp_jail, хотя и без них тоже всё прекрасно работать должно.

```
rehash
```

пример конфигурация находится в /usr/local/etc/dhcpd.conf.sample, но файлик должен называться dhcpd.conf, поэтому:

```
cp /usr/local/etc/dhcpd.conf.sample /usr/local/etc/dhcpd.conf
ee /usr/local/etc/dhcpd.conf
```

Содержимое файла /usr/local/etc/dhcpd.conf с небольшими пояснениями:

```
#доменый суффикс, который будет присвоен клиентам
#для примера домен homelan.org
option domain-name "homelan.org";
#список серверов DNS, которые должен использовать клиент,
#можно указать несколько, разделив их запятыми.
option domain-name-servers 217.24.112.2;
#Маска сети, выдаваемая клиентам.
option subnet-mask 255.255.255.0;
#На какое время выдавать клиентам настройки (в секундах) 259200сек=3дня
default-lease-time 259200;
# На какое максимальное время выдавать клиентам настройки (в секундах)
max-lease-time 259200;
#Параметр определяет, будет ли DHCP сервер
#пытаться обновить DNS при выдаче
#конфигурационной информации.
ddns-update-style none;
# включаем IP-Forwarding
option ip-forwarding on;

#диапазон IP-адресов, который будет выдаваться клиентам.
# и IP адрес шлюза (ниже адреса для примера, у вас могут быть другие)
#Если интернет у вас в сети раздаётся NAT'ом,
# то для раздачи интернета всем, шлюзом указывайте IP сервера nat
subnet 10.36.1.0 netmask 255.255.255.0 {
range 10.36.1.11 10.36.1.250;
option routers 10.36.1.10;
}

#Ниже строки для резервации IP-адреса,
#например, если вам нужно, чтобы у одного ПК был постоянный адрес,
#то стоки ниже вам пригодятся
host pc01 {
hardware ethernet 00:11:22:33:44:55;
fixed-address 10.36.1.11;
}
#здесь:
# pc01 - имя хоста
# 00:11:22:33:44:55 - MAC адрес ПК, для которого делаем резервацию.
# 10.36.1.11 - IP адрес, который будет получать ПК
# конец конфигурационного файла
```

MAC адрес можно узнать:

- под m$ wind0w$ выполнив в командной строке ipconfig -all
- под freebsd и многими unix'подобными, выполнив в консольке ifconfig

Теперь побеспокоимся об автозапуске демона dhcp, для этого в файл /etc/rc.conf , добавим строки:

```
1 dhcpd_enable="YES"
2 dhcpd_ifaces="rl1"
```

Здесь rl1 – сетевой интерфейс, вашего сервера, через который будут выдаваться адреса.

Теперь можно стартовать сервер DHCP:

```
1 cp /usr/local/etc/rc.d/isc-dhcpd /usr/local/etc/rc.d/isc-dhcpd.sh
2 /usr/local/etc/rc.d/isc-dhcpd.sh start
```

Всё, теперь клиентам указываем автоматически получать адреса и проверяем результат. По примеру выше, они должны получить адрес из 10.36.1.* подсети.

P.S.: не забудьте на фаерволе (если таковой имеется) открыть порт №68 и 67

BIND9 на FreeBSD в качестве кэширующего DNS-сервера

Делаем локальный кеширующий DNS-сервер (Domain Name System) на FreeBSD, используя BINDверсии 9, необходимый для того, чтобы «отвечать на запросы быстрее, чем это происходит при прямом опросе внешнего сервера имён».

Итак приступим к делу, всё выполняем от имени root'а.

Версия ОС:

```
1uname -a
2FreeBSD 7.1-RELEASE
```

Смотрим установленную версию DNS-сервера BIND:

```
1named -v
2BIND 9.4.2-P2
```

Забэкапимся на всякий случай:

```
1cp -r /etc/namedb/ /etc/namedb_0/
```

Правим основной конфигурационный файл:

```
1ee /etc/namedb/named.conf
```

Заполняем его так

Для просмотра скрытой части статьи вы должны авторизироваться [Login] или зарегистрироваться [Register] на сайте.

в /etc/rc.conf дописываем строку:

```
1named_enable="YES"
```

а в /etc/resolv.conf пишем только строку:

```
1nameserver     127.0.0.1
```

остальные ДНС-сервера, перечисленные в этом файле resolv.conf не нужны

стартуем BIND:

```
1/etc/rc.d/named start
2wrote key file "/var/named/etc/namedb/rndc.key"
3Starting named.
```

проверяем:

```
1nslookup mediaunix.com
```

видим:

```
1Server:     127.0.0.1
2Address:    127.0.0.1#53
3
4Non-authoritative answer:
5Name:   mediaunix.com
6Address: 78.36.111.40
```

и ещё так проверить можно:

```
1dig @10.36.1.10 google.com
```

где 10.36.1.10 – наш ДНС-сервер.

Если видим следующий ответ, значит всё в порядке:

```
1; &lt;&lt;&gt;&gt; DiG 9.6.1-P1 &lt;&lt;&gt;&gt; @10.36.1.10 google.com
2; (1 server found)
3;; global options: +cmd
4;; Got answer:
5;; -&gt;&gt;HEADER&lt;&lt;- opcode: QUERY, status: NOERROR, id: 11898
6;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 4, ADDITIONAL: 4
7
```

```
;; QUESTION SECTION:
;google.com.           IN  A

;; ANSWER SECTION:
google.com.     237 IN  A   74.125.232.50
google.com.     237 IN  A   74.125.232.52
google.com.     237 IN  A   74.125.232.49
google.com.     237 IN  A   74.125.232.48
google.com.     237 IN  A   74.125.232.51

;; AUTHORITY SECTION:
google.com.     147327  IN  NS  ns2.google.com.
google.com.     147327  IN  NS  ns1.google.com.
google.com.     147327  IN  NS  ns3.google.com.
google.com.     147327  IN  NS  ns4.google.com.

;; ADDITIONAL SECTION:
ns3.google.com.     277159  IN  A   216.239.36.10
ns2.google.com.     277159  IN  A   216.239.34.10
ns1.google.com.     277159  IN  A   216.239.32.10
ns4.google.com.     277159  IN  A   216.239.38.10

;; Query time: 70 msec
;; SERVER: 10.36.1.10#53(10.36.1.10)
;; WHEN: Sun Feb  6 19:36:05 2011
;; MSG SIZE  rcvd: 244
```

Если вы настраивали dhcp в сети и автоматически давали адреса dns-серверов как в этой статье, то подправьте в конфигурационном файле /usr/local/etc/dhcpd.conf строку:

```
option domain-name-servers 217.24.112.2;
```

указав вместо адреса 217.24.112.2 IP-адрес вашего кеширующего сервера и перестартуйте сервис dhcp:

```
/usr/local/etc/rc.d/isc-dhcpd.sh restart
```

Клиентам после этого нужно перезапустить сетевое соединение, чтобы получить новый адрес. Если же DHCP не используется, то в /etc/resolv.conf впишите перед остальными dns-серверами(если таковые имеются, но они уже станут не нужны-закомментируйте их) строку:

```
nameserver      10.36.1.10
```

В случае с клиентом на ms w!n пропишите адрес dns-сервера в настройках сетевого соединения. Но лучше 1 раз настроить dhcp-сервер и забыть про ручное прописывание параметров сетевого соединения. Вобщем то и всё, но есть небольшая плюшка.

Можно настроить rndc для управления нашим демоном сервера имён. Утилита позволяет контролировать работу демона named, в BIND8 она называлась ndc, а в BIND9 переименована в rndc. Она не обязательна, но кто хочет, настраиваем.

Выполняем:

/usr/sbin/rndc-confgen > /etc/namedb/rndc.conf

смотрим создавшийся /etc/namedb/rndc.conf:

```
cat /etc/namedb/rndc.conf
```

видим:

```
# Start of rndc.conf
key "rndc-key" {
algorithm hmac-md5;
secret "куча_символов";
};

```

```
options {
default-key "rndc-key";
default-server 127.0.0.1;
default-port 953;
};
# End of rndc.conf

# Use with the following in named.conf, adjusting the allow list as needed:
# key "rndc-key" {
#   algorithm hmac-md5;
#   secret "куча_символов";
# };
#
# controls {
#   inet 127.0.0.1 port 953
#       allow { 127.0.0.1; } keys { "rndc-key"; };
# };
# End of named.conf
```

Исходя из содержимого rndc.conf в начало /etc/namedb/named.conf дописываем:

```
key "rndc-key" {
algorithm hmac-md5;
secret "куча_символов";
};

controls {
inet 127.0.0.1 port 953
allow { 10.36.1.0/24; 127.0.0.1; } keys { "rndc-key"; };
};
```

Это те самые закомментированные строки из /etc/namedb/rndc.conf, только я добавил туда ещё свою подсеть(10.36.1.0/24) в параметр allow, но это делать не обязательно. Перестартуем bind9:

```
/etc/rc.d/named reload
server reload successful
```

проверим работоспособность rndc:

```
rndc status
number of zones: 11
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is OFF
recursive clients: 0/0/1000
tcp clients: 0/100
server is up and running
```

Для очистки кэша BIND9 достаточно выполнить:

```
rndc flush
```

Полный список команд можно посмотреть запустив rndc без параметров.

P.S.: Не забываем открыть в фаерволе порты 53(DNS) и 953(rndc) иначе rndc не заработает выдав:

```
rndc: connection to remote host closed
This may indicate that
* the remote server is using an older version of the command protocol,
* this host is not authorized to connect,
* the clocks are not syncronized, or
* the key is invalid.
```