МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра вычислительной техники

УТВЕРЖДАЮ
Проректор по учебной работе
_____ О.Г. Локтионова
«_20_» _09_____ 2022 г.

АДМИНИСТРИРОВАНИЕ ОПЕРАЦИОННЫХ СИСТЕМ

Методические указания к выполнению лабораторных работ для студентов направления подготовки 09.04.01 Информатика и вычислительная техника

Курск 2022

УДК 004

Составитель Д.О. Бобынцев

Рецензент: к.т.н., доцент Ватутин Э.И.

Администрирование операционных систем: методические указания к выполнению лабораторных работ / Юго-Зап. гос. ун-т; сост.: Д.О. Бобынцев. Курск, 2022. 47 с. Библиогр.: с. 47.

Содержит методические указания к выполнению лабораторных работ по дисциплине «Администрирование операционных систем». Указывается порядок выполнения работ, контрольные вопросы. Предназначено для студентов направления подготовки «Информатика и вычислительная техника».

Текст печатается в авторской редакции

Подписано в печать . Формат 60х84 1/16.
Усл.печ. л. 2,73. Уч.-изд. л. 2,47. Тираж 100 экз. Заказ Бесплатно.
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

Содержание

Для выполнения данного курса работ Вам понадобится виртуальная машина с установленной серверной операционной системой Ред ОС 7.3 на базе ОС Linux. Для создания машины рекомендуем пользоваться бесплатной платформой Oracle VM Virtual Box. Поскольку в список известных операционных систем у этой платформы не входит Ред ОС, выбирайте тип операционной системы на этапе создания Other Linix соответствующей разрядности. Виртуальный жёсткий диск создавайте на том физическом диске, где для него будет достаточно места в соответствии с его объёмом!

Скачать образ операционной системы Ред ОС Вы можете с официального сайта http://red-soft.ru/ Не забудьте указать на этапе установки, что Вам нужна серверная версия с графическим интерфейсом, иначе по умолчанию будет установленная версия для рабочей станции, а также провести базовую настройку, в частности, задать пароль верховного администратора root, и создать одну простую учётную запись с правами администратора для обычного входа в систему (можно без пароля).

Знакомство с командной оболочкой, корневой файловой системой и файлами устройств в Ред ОС

Цель работы: получение первичных навыков пользования файловой системой и командной оболочкой операционной системы Linux на примере Ред ОС.

Теоретический материал

Для управления ОС используется командные интерпретаторы (shell). Зайдя в систему, пользователь увидит приглашение - строку, содержащую символ «$» (далее этот символ будет обозначать командную строку). Программа ожидает ввода команд. Роль командного интерпретатора – передавать команды пользователя операционной системе. При помощи командных интерпретаторов можно писать небольшие программы - сценарии (скрипты). Оболочкой по умолчанию в РЕД ОС является «**Bash**» (Bourne Again Shell) Чтобы проверить, какая оболочка используется, необходимо выполнить команду:

echo $shell

В bash имеется несколько приемов для работы со строкой команд. Например, используя клавиатуру, можно:

- **Ctrl + A** - перейти на начало строки.
- **Ctrl + U** - удалить текущую строку.
- **Ctrl + C** - остановить текущую задачу.

Можно использовать «;» для того, чтобы ввести несколько команд одной строкой. Клавиши «вверх» и «вниз», позволяют вам перемещаться по истории команд.

Для того чтобы найти конкретную команду в списке набранных, не пролистывая всю историю, необходимо набрать:

Ctrl + R

Команды, присутствующие в истории, отображаются в списке пронумерованными. Для того, чтобы запустить конкретную команду, наберите:

! <номер команды>

Если ввести два восклицательных знака, запустится последняя из набранных команд. Иногда имена программ и команд слишком длинны. Bash сам может завершать имена. Нажав клавишу [TAB], можно завершить имя команды, программы или каталога. Например, предположим, что необходимо использовать программу декомпрессии bunzip2. Для этого нужно набрать bu, затем нажать [TAB]. Если ничего не происходит, то, вероятно, существует несколько возможных вариантов завершения команды. Нажав клавишу [TAB] еще раз, пользователь получит список имен, начинающихся с «bu».

Задание: *выведите описанным образом варианты команд, начинающихся на первые две буквы вашей фамилии, имени или отчества и покажите результат преподавателю.*

Если набрать столько начальных букв команды, сколько достаточно для её идентификации, то после нажатия Tab интерпретатор допечатает её полное имя. Программу, вызываемую из командной строки, Bash ищет в каталогах, определяемых в системной переменной PATH.

По умолчанию, в этот перечень каталогов не входит текущий каталог, обозначаемый «./» (точка слэш), поэтому для запуска программы prog из текущего каталога, надо дать команду:

./prog

Базовые команды оболочки Bash

Все команды, приведенные ниже, могут быть запущены в режиме консоли. Для получения более подробной информации используйте команду man <имя команды>. Пример:

man ls

Команда su позволяет получить права администратора. Когда пользователь набирает su, оболочка запрашивает пароль суперпользователя (root). Необходимо ввести пароль и нажать Enter. Чтобы вернуться к правам основного пользователя, необходимо набрать exit.

Команда cd позволяет сменить каталог. Она работает как с абсолютными, так и с относительными путями. Предположим, что, находясь в своем домашнем каталоге, пользователь хочет перейти в его подкаталог docs/. Для этого нужно ввести относительный путь:

cd docs/

Чтобы перейти в каталог /usr/bin, нужно набрать (абсолютный путь):

cd /usr/bin/

Некоторые варианты команды:

cd ..

позволяет сделать текущим родительский каталог,

cd -

позволяет вернуться в предыдущий каталог. Команда cd без параметров переводит в домашний каталог.

Команда ls (list) выдает список файлов в текущем каталоге. Синтаксис:

ls

Две основные опции:

-a - просмотр всех файлов, включая скрытые,

-l - отображение более подробной информации.

Команда rm используется для удаления файлов. Синтаксис:

rm <имя_файла>

У данной программы существует ряд параметров. Самые часто используемые:

-i - запрос на удаление файла,

-r - рекурсивное удаление (т.е. удаление, включая подкаталоги и скрытые файлы).

Команда mkdir позволяет создать каталог в текущем каталоге, тогда как rmdir удаляет каталог, при условии, что он пуст. Синтаксис:

mkdir <имя_каталога>

rmdir <имя_каталога>

Команда rmdir часто заменяется командой rm –rf <имя_каталога>, которая позволяет удалять каталоги, даже если они не пусты.

Задание: *создать в папке Документы домашнего каталога папку с именем, соответствующим Вашему имени и фамилии. В папке создать по одному файлу с расширениями jpg, pdf, docx, pptx, bmp. Имена задать произвольные. Вывести в терминале подробную информацию по всем файлам, после чего стереть данную папку из терминала. Результаты показать преподавателю.*

Команда less позволяет постранично просматривать текст. Синтаксис:

less <имя_файла>

Для выхода нужно нажать q.

Если Вы хотите просмотреть содержимое файла, не выходя из главного окна терминала, можно воспользоваться командой cat.

Команда grep много опций и предоставляет возможности поиска символьной строки в файле. Синтаксис:

grep <шаблон_поиска> <файл>

Команда ps отображает список текущих процессов. Колонка команд указывает имя процесса, колонка PID (идентификатор процесса) - номер процесса (этот номер используется, для операций с процессом, например, чтобы «убить» его командой kill). Синтаксис:

ps <аргументы>

Аргумент -u предоставляет больше информации, а -x позволяет просмотреть те процессы, которые не принадлежат пользователю (такие как те, что были запущены во время процесса загрузки).

Для просмотра полной информации обо всех процессах, воспользуйтесь командой sudo ps –aux.

Если программа перестала отвечать или зависла, необходимо использовать данную kill, чтобы её завершить. Синтаксис:

kill <PID_номер>

Иногда необходимо будет использовать kill -9 (когда обычная команда kill не дает желательного эффекта). Номер PID выясняется при помощи команды ps.

В соответствии с концепцией Linux «Всё есть файл» не только программы и данные, но и устройства, входящие в состав компьютера, имеют свои файлы по назначению. Например, информацию о центральном процессоре можно посмотреть в файле cpuinfo каталога proc.

Задание: *вывести на экран информацию о Вашем центральном процессоре.*

Для удобства запоминания сочетания клавиш сгруппированы по действию. Обратите внимание, что **в комбинациях следует использовать левую клавишу Alt**, т.к. правая Alt Gr используется как клавиша Compose (специальная клавиша, позволяющая вводить символы с помощью определённых комбинаций клавиш). Перечень приведен в справочном руководстве операционной системы.

Контрольные вопросы

1. Что означает постулат «Всё есть файл»?
2. Какой командный интерпретатор используется в Ред ОС по умолчанию?
3. Каким образом можно упростить набор имени команды?
4. Как посмотреть справку о команде?
5. Что делает команда su?
6. Что делает команда cd?
7. Что произойдёт, если применить cd без параметров?
8. Что делает команда ls?
9. Какими командами можно создавать и удалять каталоги? Просматривать содержимое файлов?
10. Назовите сочетания клавиш для очистки экрана интерпретатора и выхода из него.

Пользователи и права доступа в Ред ОС

Цель работы: освоение инструментария управления учётными записями пользователями в операционной системе Ред ОС.

Теоретический материал

РЕД ОС, как и любой UNIX – многопользовательская операционная система. Также имеются группы пользователей, основное предназначение которых - облегчить управление большим количеством пользователей, а также более точно распределить права доступа к различным объектам системы. Пользователи и группы внутри системы обозначаются цифровыми идентификаторами - UID и GID, соответственно.

Пользователь может входить в одну или несколько групп. По умолчанию он входит в группу, совпадающую с его именем. Чтобы узнать, в какие еще группы входит пользователь, введите команду id, вывод ее может быть примерно следующим:

uid=1000(test) gid=1000(test) группы=1000(test), 10(wheel) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

Такая запись означает, что пользователь test (цифровой идентификатор 1000) входит в группы test и wheel. Разные группы могут иметь разный уровень доступа к тем или иным каталогам; чем в большее количество групп входит пользователь, тем больше прав он имеет в системе.

Утилита passwd работает с паролями пользователей и поддерживает традиционные опции passwd и утилит shadow.

Возможные опции:

-d,- -delete - удалить пароль для указанной записи.

-f,- -force - форсировать операцию.

-k,- -keep-tokens - сохранить не устаревшие пароли.

-l,- -lock - блокировать указанную запись.

--stdin - прочитать новые пароли из стандартного ввода.

-S,- -status - дать отчет о статусе пароля в указанной записи.

-u,- -unlock - разблокировать указанную запись.

-?,- -help - показать справку и выйти.

--usage - дать короткую справку по использованию.

-V,- -version - показать версию программы и выйти.

При успешном завершении passwd заканчивает работу с кодом выхода 0. Код выхода 1 означает, что произошла ошибка. Текстовое описание ошибки выводится на стандартный поток ошибок.

Для добавления нового пользователя используйте команды useradd и passwd. В результате описанных действий в системе появится пользователь с некоторым паролем. Если пароль окажется слишком слабым с точки зрения системы, она об этом предупредит. Пользователь в дальнейшем может поменять свой пароль при помощи команды passwd, но, если он попытается поставить слабый пароль, система откажет ему (в отличие от root) в изменении.

В дистрибутивах ОС для проверки паролей на слабость используется модуль PAM passwdqc.

Программа useradd имеет множество параметров, которые позволяют менять её поведение по умолчанию. Например, можно принудительно указать, какой будет UID или какой группе будет принадлежать пользователь. Синтаксис:

useradd [-u <идентификатор> [-o] [-i]] [-g <группа>] [-G <группа>[[,<группа>] . . .]] [-d <каталог>] [-s <shell>] [-c <комментарий>] [-m [-k <skel_dir>]] [-f <inactive>] [-e <expire>] [-p <passgen>] [-a <событие>[, . . .]]

Рассмотрим опции данной команды.

-u <идентификатор> - идентификационный номер пользователя (UID). Этот номер должен быть неотрицательным целым числом, не превосходящим MAXUID, определенный в sys/param.h. По умолчанию используется следующий доступный (уникальный) не устаревший UID, больший 99. Эта опция игнорируется, если новое регистрационное имя будет администрироваться сетевой информационной службой (NIS).

-o - эта опция позволяет сдублировать UID (сделать его не уникальным). Поскольку защита системы в целом, а также целостность контрольного журнала (audit trail) и учетной информации (accounting information) в частности, зависит от однозначного соответствия каждого UID определенному лицу,

использовать эту опцию не рекомендуется (чтобы обеспечить учет действий пользователей).

-i - позволяет использовать устаревший идентификатор UID.

-g <группа> - целочисленный идентификатор или символьное имя существующей группы. Эта опция задает основную группу (primary group) для нового пользователя. По умолчанию используется стандартная группа, указанная в файле /etc/default/useradd. Эта опция игнорируется, если новое регистрационное имя будет администрироваться сетевой информационной службой (NIS).

-G <группа>[[,<группа>] . . .] - один или несколько элементов в списке через запятую, каждый из которых представляет собой целочисленный идентификатор или символьное имя существующей группы. Этот список определяет принадлежность к дополнительным группам (supplementary group membership) для пользователя. Повторения игнорируются. Количество элементов в списке не должно превосходить NGROUPS_MAX - 1, поскольку общее количество дополнительных групп для пользователя плюс основная группа не должно превосходить NGROUPS_MAX. Эта опция игнорируется, если новое регистрационное имя будет администрироваться сетевой информационной службой (NIS).

-d <каталог> - начальный каталог (home directory) нового пользователя. Длина этого поля не должна превосходить 256 символов. По умолчанию используется HOMEDIR/a, где HOMEDIR - базовый каталог для начальных каталогов новых пользователей, a - регистрационное имя нового пользователя.

-s <shell> - полный путь к программе, используемой в качестве начального командного интерпретатора для пользователя сразу после регистрации. Длина этого поля не должна превосходить 256 символов. По умолчанию это поле - пустое, что заставляет систему использовать стандартный командный интерпретатор /usr/bin/sh. В качестве значения shell должен быть указан существующий выполняемый файл.

-c <комментарий> - любая текстовая строка. Обычно, это краткое описание регистрационного имени и используется сейчас для указания фамилии и имени реального пользователя. Эта

информация хранится в записи пользователя в файле /etc/passwd. Длина этого поля не должна превосходить 128 символов.

-m - создает начальный каталог нового пользователя, если он еще не существует. Если каталог уже существует, добавляемый пользователь должен иметь права на доступ к указанному каталогу.

-k <skel_dir> - копирует содержимое каталога <skel_dir> в начальный каталог нового пользователя, вместо содержимого стандартного "скелетного" каталога, /etc/skel. Каталог <skel_dir> должен существовать. Стандартный "скелетный" каталог содержит стандартные файлы, определяющие среду работы пользователя. Заданный администратором каталог <skel_dir> может содержать аналогичные файлы и каталоги, созданные для определенной цели.

-f <inactive> - максимально допустимое количество дней между использованиями регистрационного имени, когда это имя еще не объявляется недействительным. Обычно в качестве значений используются положительные целые числа.

-e <expire> - дата, начиная с которой регистрационное имя больше нельзя будет использовать; после этой даты никакой пользователь не сможет получить доступ под этим регистрационным именем. (Эта опция удобна при создании временных регистрационных имен.) Вводить значение аргумента expire (представляющего собой дату) можно в любом формате (кроме Julian date). Например, можно ввести 10/6/99 или October 6, 1999.

-p <passgen> - указывает, что поле FLAG в файле /etc/shadow должно быть установлено в указанное значение. К этому полю обращается команда passwd, чтобы определить, действует ли для данного пользователя генератор паролей.

Если опция -p явно не задана, проверяется запись FORCED_PASS в файле /etc/default/useradd, чтобы определить значение для соответствующего поля в /etc/shadow.

Если записи FORCED_PASS нет в /etc/default/useradd, в соответствующем поле записи в /etc/shadow значения не будет.

Если значение FORCED_PASS равно 1, запись в /etc/shadow получает значение 1.

Если значение passgen не пустое и не является печатным символом ASCII, выдается диагностическое сообщение.

-a <событие> - список типов или классов событий через запятую, образующих маску аудита (audit mask) для пользователя. Сразу после установки системы стандартной маски аудита для пользователя нет, но ее можно задать в файле /etc/default/useradd с помощью команды defadm. Эту опцию можно использовать, только если установлены утилиты аудита (Auditing Utilities).

<рег_имя> - строка печатных символов, задающая регистрационное имя для нового пользователя. В ней не должно быть двоеточий (:) и символов перевода строки (\n). Она также не должна начинаться с прописной буквы.

Задание: *получите права суперпользователя root и создайте учётную запись пользователя с логином из Вашего имени и фамилии на латинице при помощи команды useradd, затем задайте пароль командой passwd. Попробуйте переключиться на созданную учётную запись. Если всё прошло успешно, покажите результат преподавателю.*

Для модификации уже имеющихся пользовательских записей применяется утилита usermod. Её синтаксис usermod [Параметры] учётная_запись. Со списком параметров Вы можете ознакомиться самостоятельно, посмотрев описание команды утилитой man.

Задание: *изучите опции команды usermod и измените логин созданной вами учётной записи, поменяв местами имя и фамилию. Покажите результат преподавателю.*

Просмотреть списки всех пользователей системы и всех групп пользователей можно командами

cat /etc/passwd

cat /etc/group

Для просмотра групп, в которых состоит пользователь, используется команда groups учётная_запись. Удаление пользователей выполняется командой userdel учётная_запись. Если будет дополнительно задан параметр -r, то будет уничтожен и домашний каталог пользователя. Нельзя удалить пользователя, если в данный момент он ещё работает в системе.

Утилиты vigr и vipw используются для ручного редактирования файлов /etc/passwd и /etc/group, в которых хранятся основные записи о пользователях и группах в системе.

Не рекомендуется создавать пользователей с правами сверх необходимых. Предпочтительнее создать серию новых групп и включить в них требуемого пользователя. А для данных групп установить соответствующие права на объектах файловой системы (утилиты chmod и chown).

/etc/passwd – файл, содержащий в текстовом формате список пользовательских учётных записей (аккаунтов).

Является первым и основным источником информации о правах пользователя операционной системы. Существует в большинстве версий и вариантов UNIX-систем. Обязан присутствовать в POSIX-совместимой операционной системе.

Принцип:

login : password : UID : GID : GECOS : home : shell

Каждая строка файла описывает одного пользователя и содержит семь полей, разделённых двоеточиями:

- регистрационное имя или логин;
- хеш пароля;
- идентификатор пользователя;
- идентификатор группы по умолчанию;
- информационное поле GECOS;
- начальный (он же домашний) каталог;
- регистрационная оболочка, или shell.

Пример записи:

bin:x:1:root,bin,daemon

Здесь сообщается, что группа bin имеет GID=1, а входят в неё пользователи root, bin и daemon.

Поле GECOS хранит вспомогательную информацию о пользователе (номер телефона, адрес, полное имя и так далее). Оно не имеет чётко определённого синтаксиса.

Тем не менее, демон (служебная программа, работающая в фоновом режиме) fingerd предполагает, что в нём содержатся следующие элементы, разделённые запятыми:

- полное имя;
- адрес офиса или домашний адрес;
- рабочий телефон;
- домашний телефон.

С помощью утилиты chfn можно изменять эту информацию, а с помощью finger — узнать, например, полное имя любого пользователя в системе (или даже на другом компьютере сети).

Пример строки с заполненным полем GECOS:

tester:x:210:8:Edward Chernenko,Marx Street 10,4554391,5454221:/home/ed:/bin/bash

После входа в систему пользователь оказывается в своём домашнем каталоге. Исторически сложилось так, что домашний каталог пользователя root называется /root, а остальные имеют вид /home/. Но могут применяться и другие схемы.

Если на момент входа в систему домашний каталог отсутствует, то система выдаёт сообщение об ошибке и отказывается допустить пользователя к командной строке. Такое поведение НЕ характерно для GNU/Linux; в большинстве дистрибутивов этой ОС просто выводится предупреждение, после чего пользователь попадает в каталог «/». Это можно изменить посредством установки параметра DEFAULT_HOME в файле /etc/login.defs в значение «no».

Следует отметить, что при использовании графического интерфейса (KDE, GNOME) пользователь не увидит предупреждения или сообщения об ошибке, но просто будет выведен из системы безо всяких объяснений (так как оконный менеджер не сможет выполнить запись в нужный каталог, такой как ~/.gnome).

В поле регистрационной оболочки задаётся shell, то есть интерпретатор командной строки. Здесь может быть указана любая программа, и пользователь может сам выбирать для себя наиболее подходящую при помощи команды chsh. Тем не менее, некоторые системы в целях безопасности требуют, чтобы суперпользователь явно разрешил использовать приложение в качестве интерпретатора командной строки. Для этого используется специальный файл /etc/shells, содержащий список «допустимых» оболочек.

vipw - запускает текстовый редактор, указанный в переменной среды EDITOR (или редактор по умолчанию, обычно vi), загружая в него копию файла /etc/passwd. После закрытия редактора

переносит временную копию в сам файл. Не позволяет двум пользователям выполнять редактирование одновременно.

В файле /etc/shadow хранятся хеши паролей всех пользователей в системе. Процессы суперпользователя могут читать его напрямую, а для остальных создана специальная библиотека PAM. Она позволяет непривилегированным приложениям спрашивать у неё, правильный ли пароль ввёл пользователь, и получать ответ. Библиотека PAM как правило действует с привилегиями вызвавшего процесса. Таким образом, хеш не попадает «в чужие руки».

В ранних UNIX пароль шифровался с помощью одного из вариантов DES, теперь используется MD5-хеширование или blowfish-хеширование (bcrypt). MD5-хеши всегда записываются после префикса «1».

Перед хешированием к паролю добавляются случайные символы - «salt» (соль, от англ. add salt to something — сделать что-либо более интересным; в русскоязычных источниках иногда используется термин «затравка»). Salt также приписывается к началу полученного хеша. Благодаря salt нельзя при простом просмотре файла обнаружить пользователей с одинаковыми паролями.

Кроме имени (первое поле каждой строки) и хеша (второе поле) в файле /etc/shadow также хранятся:

- дата последнего изменения пароля;
- через сколько дней можно будет поменять пароль;
- через сколько дней пароль устареет;
- за сколько дней до того, как пароль устареет, начать напоминать о необходимости смены пароля;
- через сколько дней после того, как пароль устареет, заблокировать учётную запись пользователя;
- дата, при достижении которой учётная запись блокируется;
- зарезервированное поле.

Даты обозначаются как число дней с 1 января 1970 года (начало эпохи UNIX).

Обязательная регулярная смена паролей — это популярная административная мера, призванная сделать учётные записи более

защищёнными. К сожалению, многие пользователи после принудительного изменения возвращают себе старый пароль.

Для активации автоматического входа пользователя (автовход) в GDM без ввода пароля при загрузке ОС, необходимо в файл /etc/gdm/custom.conf в секцию [daemon] добавить 2 строки:

AutomaticLogin=<имя_пользователя>

AutomaticLoginEnable=True

При использовании этой конфигурации, при завершении сеанса пользователя, произойдет обратный автоматический вход в текущего пользователя.

Чтобы иметь возможность переключиться в сеанс другого пользователя, приведите в файле /etc/gdm/custom.conf секцию [daemon] к виду:

[daemon]

WaylandEnable=false

TimedLoginEnable = true

TimedLogin = <имя_пользователя>

TimedLoginDelay = 10

Здесь в параметре TimedLoginDelay указывается время (в секундах) ожидания до автоматического входа в пользователя. За это время можно успеть войти в рабочий стол другого пользователя.

Контрольные вопросы

1. Что такое UID?
2. Что такое GID?
3. Как просмотреть список групп, в которые входит пользователь?
4. Как просмотреть список всех пользователей системы?
5. Опишите формат смены имени пользователя.
6. Опишите формат добавления пользователя в группу.
7. Можно ли удалить пользователя, авторизованного в системе в данный момент?
8. Что такое аутентификация?
9. Что такое авторизация?
10. Как активировать автовход пользователя в систему?

Настройка DHCP-сервера Ред ОС

Цель работы: научиться проводить базовую настройку сервера DHCP на базе серверной версии операционной системы Ред ОС.

Теоретический материал

DHCP – протокол динамической конфигурации хоста, представляет собой протокол сетевого управления, посредством которого сервер DHCP динамически назначает IP-адрес и другие параметры конфигурации сети для каждого устройства в сети.

Включенный в состав РЕД ОС DHCP-сервер, соответствующий RFC 2131, включает фирменные расширения BOOTP (RFC 2132). Так же поддерживает протокол конфигурации динамического хоста для IPv4 (DHCPv4), который может использоваться для обмена информацией о полном доменном имени клиента DHCPv4, описанный в RFC 4702.

Поддерживается стандарт DHCPv6, описанный в RFC 3315. Так же поддерживается контроль и настройка параметров доступа к беспроводным точкам доступа (CAPWAP) DHCP, описанные в RFC 5417.

Задачей данной работы является базовая настройка DHCP-сервера на базе серверной операционной системы Ред ОС. Для этого Вам понадобится виртуальная машина с данной операционной системой.

Для установки DHCP откройте терминал. Установку можно проводить только с правами верховного администратора, поэтому вначале выполните команду su -, после чего введите пароль верховного администратора. Если авторизация прошла успешно, выполните команду dnf install dhcp для Ред ОС 7.3 или yum install dhcp для более старой версии. Следуйте указаниям консольного установщика.

Теперь откроем на редактирование конфигурационный файл:

nano /etc/dhcp/dhcpd.conf

Подсети обозначаются блоками, пример такого блока представлен ниже:

subnet 192.168.0.0 netmask 255.255.255.0 {

```
  range 192.168.0.0 192.168.0.100;
  range 192.168.0.120 192.168.0.254;
  option domain-name-servers 192.168.0.10, 192.168.0.11;
  option domain-name "redos.test";
  option routers 192.168.0.1;
  option broadcast-address 192.168.0.255;
  default-lease-time 600;
  max-lease-time 7200;
 }
```

* *где*

- ***subnet*** *обозначает сеть, в области которой будет работать данная группа настроек;*
- ***range*** *- диапазон, из которого будут браться IP-адреса, диапазон адресов 192..168.0.101-192.169.0.120 выдаваться не будет;*
- ***option domain-name-servers*** *- через запятую перечисленные DNS-сервера;*
- ***option domain-name*** *- суффикс доменного имени;*
- ***option routers*** *- шлюз по умолчанию;*
- ***option broadcast-address*** *- адрес сети для широковещательных запросов;*
- ***default-lease-time***, ***max-lease-time*** *- время и максимальное время в секундах, на которое клиент получит адрес, по его истечению будет выполнено продление срока.*

Задание: *добавьте подсеть с начальным адресом N.N/2.1.1. и конечным N.N/2.1.N+10, где N – ваш номер по списку, умноженный на 16. Сохраните файл, переведя курсор на строку команд и выбрав нужную, и покажите преподавателю.*

Добавляем правило в **firewalld**:

```
firewall-cmd --permanent --add-service=dhcp
firewall-cmd --reload
```

Рассмотрим некоторые дополнительные параметры.

option domain-name *string***;** - параметр задает доменное имя, которое клиенты используют при запросах к DNS, при разрешении имен.

option netbios-name-servers <*ip-address*> [**,** *ip-address...*]**;** - задает список серверов имен **NetBIOS** (NBNS),

соответствующих **RFC 1001/1002.** Сервера должны быть перечислены в порядке предпочтительности. Сервера имен **NetBIOS** также известны как сервера **WINS**.

option netbios-dd-server <*ip-address*> [**,** *ip-address*...]**; -** опция сервера распределения данных **NetBIOS** (NBDD) указывает список серверов **RFC 1001/1002 NBDD**. Серверы должны быть перечислены в порядке предпочтения.

option netbios-node-type INT**; -** параметр позволяет сконфигурировать тип узла, т.е. способ разрешения имен клиентами **NetBIOS** поверх **TCP/IP**.

Возможные значения параметра:

1 B-node (Broadcast): разрешение имен с помощью широковещательных запросов, WINS не используется.

2 P-node (Peer): используется только WINS.

4 M-node (Mixed): смешанный тип, сначала используется широковещательный запрос, затем в случае неудачи - WINS

8 H-node (Hybrid): смешанный наоборот. WINS, а затем broadcast.

option netbios-scope *string***;**

Опция области **NetBIOS** указывает параметр **NetBIOS** через **TCP/IP** для клиента, как указано в **RFC 1001/1002.** См. RFC1001, RFC1002 и RFC1035 для ограничения набора символов.

Каждый клиент **BOOTP** может быть настроен отдельно в файле **dhcpd.conf**. Самая простая конфигурация состоит из адреса сетевой карты и IP-адреса, назначаемого этому клиенту. Если клиент должен быть загружен с сервера, указывается загрузочный образ. Простая конфигурация для клиента **BOOTP** может выглядеть так:

```
host haagen {
        hardware ethernet 08:00:2b:4c:59:23;
        fixed-address 239.252.197.9;
        filename "/tftpboot/haagen.boot";
       }
```

Включение и отключение bootp

```
allow bootp;
deny bootp;
```

Параметр bootp сообщает серверу dhcp обрабатывать или нет bootp-запросы. По умолчанию bootp-запросы разрешены.

Параметр dynamic-bootp в секции range указывается в случае, если предполагается назначать адреса из диапазона клиентам по протоколу BOOTP.

Необходима хотя бы одна секция host для каждого BOOTP-клиента, обслуживаемого сервером. Так же host может быть указана для DHCP-клиентов, хотя это и не обязательно, если только не требуется раздача адресов только определенным клиентам.

Если требуется обеспечить выдачу фиксированных адресов и конфигурацию клиентов по протоколам DHCP или BOOTP в более, чем одной подсети, то можно указать несколько адресов с помощью параметра fixed-address или указать несколько секций host.

Если клиентские параметры меняются в зависимости от сети, к которой подключен клиент, то необходимо использовать множественные записи host.

<имя_хоста> - имя, идентифицирующее хост. Если в описании хоста опция hostname не указана, то используется значение <имя_хоста>.

Объявления Host сопоставляются с реальными DHCP или BOOTP-клиентами путем сравнения опции dhcp-client-identifier, указанной в секции host со значением, предоставленным клиентом, или если клиент не предоставляет dhcp-client-identifier, то путем сравнения опции hardware и аппаратного (mac) адреса клиента. Клиенты BOOTP нормально не предоставляют dhcp-client-identifier, так что, при работе по протоколу BOOTP, необходимо использовать mac-адреса.

hardware <тип_железа> <аппаратный_адрес>;

Для того что бы BOOTP-клиент был опознан сервером, его mac-адрес должен быть объявлен с помощью параметра hardware в секции host параметр <тип_железа> - тип физического интерфейса. В настоящее время используются только ethernet и token-ring и, возможно, скоро будут реализованы другие типы, особенно fddi.

<аппаратный_адрес> - записывается как набор шестнадцатеричных значений (от 0 до ff), разделенных двоеточиями. Параметр hardware можно использовать и для DHCP-клиентов.

Параметр dynamic-bootp-lease-cutoff

dynamic-bootp-lease-cutoff <дата>;

Параметр dynamic-bootp-lease-cutoff устанавливает момент времени, в который все адреса, назначенные клиентам BOOTP, должны быть освобождены. Так как клиенты BOOTP не имеют возможности продлить срок использования полученных адресов и не могут определить, что срок аренды истек, то по умолчанию клиентам BOOTP адреса выделяются на неограниченный срок. Однако в некоторых случаях оказывается полезным установить параметр dynamic-bootp-lease-cutoff, например в учебном заведении при окончании семестра или в ночное время, когда предприятие не работает, а все компьютеры выключены.

<дата> - срок окончания действия всех адресов, выделенных BOOTP-клиентам. Дата указывается в следующем формате: W ГГГГ/ММ/ДД ЧЧ:ММ:СС .

W - день недели в виде числа от 0 (воскресенье) до 6 (суббота);

ГГГГ - год;

ММ - месяц 1..12;

ДД - день месяца 1..31;

ЧЧ - часы 0..23 ММ - минуты 0..59;

СС - секунды 0..59.

Время указывается в GMT, а не местное.

Параметр dynamic-bootp-lease-length

dynamic-bootp-lease-length <время_в_секундах>;

Параметр dynamic-bootp-lease-length используется для указания срока, на который выделяется адрес BOOTP-клиенту. В некоторых организациях возможна ситуация, что по прошествии определенного времени можно быть уверенным, что выделенный адрес уже не используется. Период задается числом секунд. Если клиент перезагружается в момент, когда срок, указанный в этом параметре еще не истек, то период аренды адреса заново устанавливается в указанное значение. Таким образом, часто

перезагружаемые BOOTP-клиенты могут постоянно удерживать свой адрес. Необходимо заметить, что при настройке параметра следует быть осторожным.

always-reply-rfc1048 <on/off>;

Некоторые BOOTP-клиенты ожидают ответа от сервера в стиле RFC1048, но сами не следуют RFC1048 при посылке запросов. Если в вашей сети имеется такой проблемный клиент, который не воспринимает параметры, передаваемые ему сервером, и если в логах появляется сообщение "(non-rfc1048)" , то можно воспользоваться этим параметром.

Если у вас все клиенты требуют работы в стиле RFC1048, то можно использовать параметр always-reply-rfc1048. Параметр может быть указан в любом объявлении, в этом случае - воздействует только на тех клиентов, которые соответствуют указанной зоне действия.

Резервирование IP-адреса за клиентом

Хост с именем myhost, у которого сетевая карта имеет MAC 08:45:32:00:00:23 должен иметь постоянный адрес 192.168.1.121.

```
host myhost {
    hardware ethernet 08:45:32:00:00:23;
    fixed-address 192.168.1.121;
}
```

Определённый интерфейс для работы

Если в системе присутствует несколько сетевых адаптеров, а сервер DHCP должен работать только для определенных, открываем на редактирование следующий файл:

vi /etc/sysconfig/dhcpd

И добавляем в него следующее:

DHCPDARGS=enp0s8

* в данном примере сервер будет работать только для интерфейса enp0s8.

Перед запуском убедитесь, что сетевой адаптер enp0s8 настроен для работы в создаваемой сети. Для этого создайте файл /etc/sysconfig/network-scripts/ifcfg-enp0s8 со следующим содержимым.

TYPE="Ethernet"

BOOTPROTO="none"
DNS1="192.168.0.1"
IPADDR0="192.168.0.1"
PREFIX0=24
GATEWAY0=192.168.0.1
DEFROUTE="yes"
PEERDNS="yes"
PEERROUTES="yes"
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes"
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_PEERDNS="yes"
IPV6_PEERROUTES="yes"
IPV6_FAILURE_FATAL="no"
IPV6_ADDR_GEN_MODE="stable-privacy"
NAME="enp0s8"
DEVICE="enp0s8"
ONBOOT="yes"
Разрешаем автозапуск сервиса:
systemctl enable dhcpd
и запускаем его:
systemctl start dhcpd

Настройка логов

По умолчанию, сервер dhcp ведет лог в файле /var/log/messages, что не очень удобно, так как это общий лог-файл, в котором может находиться много записей.

Для того чтобы сервер сохранял записи в отдельный файл, открываем на редактирование rsyslog.conf:

vi /etc/rsyslog.conf
И добавляем следующее:
local6.* /var/log/dhcp.log
Далее открываем конфигурационный файл dhcp:
nano /etc/dhcp/dhcpd.conf
И добавляем:

log-facility local6;

Перезапускаем сервисы:

systemctl restart dhcpd

systemctl restart rsyslog

Настройка DHCP-сервера для работы с сервером PXE.

Отредактируйте файл /etc/dhcp/dhcpd.conf ,

vi /etc/dhcp/dhcpd.conf

Добавьте строку в конец файла до фигурной скобки.

filename "pxelinux.0";

}

next-server <имя_сервера>;

Параметр next-server используется для указания клиенту адреса сервера, с которого должен быть получен загрузочный файл (тот самый файл, что указан в параметре filename). <Имя_сервера> может быть IP-адресом или доменным именем. Если этот параметр не указан, то используется адрес DHCP-сервера.

Контрольные вопросы

1. Для чего предназначена служба DHCP?
2. Что такое пул адресов?
3. Как создать пул адресов в операционной системе Ред ОС?
4. Какие параметры определяют время аренды IP-адреса?
5. Чем отличается IP-адрес от MAC-адреса?
6. Как настроить сервер для определённых сетевых адаптеров?
7. Как зарезервировать IP-адрес за клиентом?

Настройка DNS в Ред ОС

Цель работы: освоение настройку службы DNS в серверной версии операционной системы Ред ОС

Теоретический материал

Служба DNS (bind)

Если локальная сеть не подключена к сети Интернет, вполне возможно, что внутренний DNS-сервер в ней не нужен. За преобразование доменного имени в IP-адрес и обратно в Linux отвечает несколько разных механизмов, лишь один из которых базируется на службе доменных имён. В самом простом случае имена всех компьютеров вместе с их адресами можно записать в файл /etc/hosts. Порядок просмотра различных пространств имён указывается в файле /etc/nsswitch.conf. Строка hosts: files nisplus nis dns этого файла предписывает приложениям, пользующимся стандартной функцией gethostbyname() сначала заглянуть в /etc/hosts, затем попытаться получить таблицы hosts службы nisplus или nis, а только затем обращаться к DNS-серверу. Это вполне разумное правило, учитывая время выполнения запроса в каждом случае.

Если задачу преобразования имён в адреса взял на себя провайдер, собственный DNS-сервер можно тоже не заводить. В этом случае на всех компьютерах в качестве сервера имён указывается сервер провайдера (поле nameserver в файле /etc/resolv.conf), к которому и идут все запросы. Даже если внутренняя сеть организована согласно RFC1918 (интранет) и адреса компьютеров в ней не могут отображаться в сети Интернет, DNS-запросы во внешний мир работать будут. Между собой компьютерам предлагается общаться с помощью /etc/hosts или IP-адресов.

Однако уже здесь очевидны две задачи, для решения которых можно организовать собственную службу доменных имён.

Первая задача – уменьшение времени ответа на DNS-запрос абонентов внутренней сети. Если канал подключения к Интернет обладает большим временем задержки (скажем, четверть секунды), то работа с данными, включающими в себя много доменных имён

(например, с www-страницами) может стать весьма медленной, хотя общий объём трафика будет невелик. Система доменных имён – распределённая база данных, замечательно поддерживающая механизм кеширования запросов. Первое обращение к кэширующему DNS-серверу приводит к выполнению рекурсивного запроса: опрашивается сервер более высокого уровня, который, если не знает ответа, передаст запрос дальше. Результат запроса оседает в кэше, так что все последующие обращения именно к этой записи дальше кэширующего сервера не уйдут. Время жизни (Time To Live, TTL) записи в кэше определяется хозяином запрошенного доменного имени. По истечении TTL запись из кэша удаляется.

Вторая задача – именование компьютеров в интранет-сети. Это бывает важно, если среди компьютеров внутренней сети есть свои серверы (например, корпоративный WWW-сервер), к которым другие компьютеры обращаются по доменному имени. Поскольку адреса такой сети не пойдут дальше межсетевого экрана, Вы можете использовать имя какого угодно – в том числе несуществующего – домена и составить табличку в /etc/hosts. Однако раздавать эту табличку лучше всё-таки средствами DNS.

Обе эти задачи можно решить, воспользовавшись Bind - мощным полнофункциональным DNS-сервером (обратите внимание, пакет и сервис называются bind, а сама служба - named). При необходимости запуска bind с использованием chroot необходимо установить пакет bind-chroot. В этом случае будет использоваться каталог /var/named/chroot/.

Для того, чтобы запустить named в кеширующем режиме, достаточно раскомментировать и заполнить раздел настройки forwarders (вышестоящие серверы) в файле /var/named/chroot/etc/named.conf. Обратите внимание на возможные ограничения на право обращаться к серверу с обычными и рекурсивными запросами (настройки allow-query и allow-recursion). Можно раскомментировать находящиеся в этом файле разумные установки по умолчанию. Эти настройки открывают доступ только абонентам локальных сетей, т.е. сетей, к которым компьютер подключён непосредственно.

```
grep allow- /var/named/chroot/etc/named.conf
allow-query     { localhost; any; };
```

allow-query-cache { localhost; any; };

Использование Bind для полноценного именования компьютеров в локальной сети требует создания двух зон (т.е. прямой и обратной), содержащих в виде записей определённого формата информацию о доменных именах компьютеров и об их роли в этих доменах. Каждая зона должна включать запись типа SOA (State Of Authority, сведения об ответственности). В этой записи определяются основные временные и административные параметры домена, в том числе электронный адрес лица, ответственного за домен (администратора) и серийный номер зоны. Серийный номер - число в диапазоне от 0 до 4294967295 (2**32); каждое изменение, вносимое в зону, должно сопровождаться увеличением этого номера. Обнаружив увеличение серийного номера, кеширующие и вторичные серверы признают все закэшированные записи из этой зоны устаревшими. Структура номера может быть любой, лишь бы он постоянно увеличивался. Удобно использовать формат «<год><месяц><число><версия>», где все числа, кроме года, двузначные, а версия может обнуляться раз в день, соответствовать времени (например, по формуле 100*(часы*60+минуты)/(60*24) или иметь сквозную нумерацию (в этом случае появляется сложность с переходом от версии 99 к версии 100, то есть 0). Даже если серийный номер генерируется автоматически, рекомендуется пользоваться этим форматом, наглядно отражающим время создания зоны. Пример зоны, не содержащей ничего, кроме записи SOA и обязательной записи типа NS, находится в файле /usr/share/doc/bind/sample/var/named /named.localhost.

Кроме записи типа SOA, в каждой зоне должна быть хотя бы одна запись типа NS (Name Server), указывающая адрес DNS-сервера, авторитетного в этом домене (как минимум - адрес сервера, на котором запущен named). Несколько зон включаются в настройку Bind автоматически (файл /var/named/chroot/etc/named.rfc1912.zones). Они нужны для обслуживания сети, привязанной к сетевой заглушке (127.0.0.1/8). Стоит обратить внимание на то, что имя домена, который обслуживается зоной, задаётся в файле настроек, а в самом файле зоны можно использовать относительную адресацию (без . в конце

имени), так что операция переименования домена делается редактированием одной строки. Рекомендуется добавлять описания зон в конфигурационный файл /var/named/chroot/etc/named.rfc1912.zones.

Прямая зона нужна для преобразования доменного имени в IP-адрес – операции, необходимой многим программам постоянно. Большинство записей в прямой зоне – типа A (Address) – предназначены именно для этого. Другие часто встречающиеся типы записей – это CNAME (Canonical Name, настоящее имя), позволяющий привязать несколько дополнительных имён к одному, и MX (Mail eXchange, обмен почтой), указывающий, куда пересылать почтовые сообщения, в поле адресат которых встречается определённое доменное имя. Вот пример прямой зоны для воображаемого домена internal.domain.net (незначащие поля соответствующих файлов заменены на . . .):

```
cat /var/named/chroot/etc/named.rfc1912.zones
 . . .
zone "example" IN {
        type master;
        file "internal.domain.net";
        allow-update { none; };
};
 . . .
cat /var/named/chroot/var/named/internal.domain.net
$TTL    1D
@       IN      SOA server root.server (
2004082202      ; serial
12H             ; refresh
1H              ; retry
1W              ; expire
1H              ; ncache
                    )
IN      NS      server
MX 10   server
Server          A       10.10.10.1
www             CNAME   server
mail            CNAME   server
```

jack A 10.10.10.100

В этом примере совпадение имени файла и имени зоны осмысленно – так проще разбираться с содержимым каталога /var/named/chroot/var/named/. Используются правила умолчания: если в записи некоторое поле опущено, оно наследуется от предыдущей. Так, вместо A можно было бы всюду написать IN A, а вместо MX - @ IN MX (@ означает имя домена, указанное в конфигурационном файле). Как видно из примера, всю работу в сети делает компьютер с адресом 10.10.10.1, он же server.internal.domain.net, он же www.internal.domain.net и mail.internal.domain.net. Несмотря на наличие среди CNAME этого сервера имени mail, MX-запись указывает всё же не на него, а на действительный адрес - так рекомендовано RFC. Распространённая практика заводить запись типа A непосредственно на имя зоны (т. е. присваивать адрес имени internal.domain.net) приводит, помимо незначительных выгод, к мелким, неочевидным и весьма трудно преодолимым трудностям, которые мы здесь описывать не будем, ограничившись советом не использовать этот - пусть и законный - трюк.

Для того чтобы преобразовывать IP-адреса в доменные имена, у каждой сети должна быть обратная зона. Если такой зоны нет, и в файле /etc/hosts тоже ничего не написано, операция не выполнится. Такое преобразование нужно гораздо реже и в основном по соображениям административным: для того, чтобы выяснить принадлежность компьютера (с которого, допустим, пытаются атаковать сервер) по его IP-адресу. Некоторые почтовые серверы проверяют, содержится ли IP-адрес машины, передающей сообщение, в обратной зоне и похоже ли полученное доменное имя на то, что указано в сообщении, и при несовпадении отказываются принимать письмо. К сожалению, поскольку неудобства, связанные с отсутствием обратной зоны, понятны лишь грамотному администратору, а таковых на просторах Интернета не слишком много, обратные зоны во множестве сетей либо отсутствуют, либо дают неверную информацию. В случае внутренней сети обратная зона необязательна, но желательна - для простоты администрирования и удовлетворения потребностей разных программных продуктов, которые ею пользуются.

Обратная зона состоит почти целиком из записей типа PTR (Pointer, указатель). Чтобы не умножать сущностей, решено было не вводить новый способ работы сервера имён и представить обратное преобразование IP-адреса как прямое преобразование доменного имени специального вида. Например, чтобы выяснить доменное имя компьютера с адресом 1.2.3.4, необходимо запросить информацию о доменном имени 4.3.2.1.in-addr.arpa. Таким образом, каждой подсети класса C (или выше) соответствует определённый домен, в котором можно найти ответ. Вот как выглядит обратная зона для нашего воображаемого домена:

```
cat /var/named/chroot/etc/named.rfc1912.zones
zone "12.11.10.in-addr.arpa" { type master;
file "12.11.10.in-addr.arpa";
};
 cat /var/named/chroot/var/named/12.11.10.in-addr.arpa
$TTL 1D
@ IN SOA server.internal.domain.net. root.server.internal.domain.net (
2004082201 ; serial
12H ; refresh
1H ; retry
1W ; expire
1H ; ncache
)
IN NS server.internal.domain.net
0 PTR internal.domain.net.
1 PTR server.internal.domain.net
100 PTR jack.internal.domain.net.
```

Обратите внимание, что относительные адреса, использованные в левой части записей PTR, раскрываются в полные вида <адрес>.12.11.10.in-addr.arpa, а в правой части используются полные (которые вполне могут указывать на имена в разных доменах).

Проверить синтаксическую правильность конфигурационного файла и файла зоны можно с помощью утилит named-checkconf и named-checkzone, входящих в пакет bind. Они же используются при запуске службы командой service bind start. Однако при

интенсивной эксплуатации сервера, когда остановка службы нежелательна, можно попросить службу перечитать файлы настроек или определённые зоны с помощью утилиты rndc - вот тогда не стоит забывать о проверке синтаксиса.

Стоит иметь в виду, что, в отличие от прямых зон, обратные описывают административную принадлежность компьютеров, но сами принадлежат хозяину сети (как правило, провайдеру). Возникает особого рода затруднение, связанное с работой DNS-сервера уже не во внутренней сети, а в сети Интернет. Дело в том, что подсети класса C (т. н. сети /24, в которых сетевая маска занимает 24 бита, а адрес компьютера - 8) выдаются только организациям, способным такую подсеть освоить (в сети класса C 254 абонентских IP-адреса, один адрес сети и один широковещательный адрес). Чаще всего выдаются совсем маленькие подсети - от /30 (на два абонентских адреса) до /27 (на 30 адресов) - или другие диапазоны, сетевая маска которых не выровнена по границе байта. Таких подсетей в обратной зоне получится несколько, а возможности просто разделить её, отдав часть адресов в администрирование хозяевам, нет. Грамотный провайдер в таких случаях пользуется RFC2317 (есть в пакете bind-docs), предписывающем в обратной зоне заводить не записи вида PTR, а ссылки CNAME на адреса в «классифицированных» братных зонах специального вида. Обратное преобразование становится двухступенчатым, зато администрирование каждой классифицированной зоны можно отдать хозяину.

DNS-сервер, отвечающий на запросы из сети Интернет, должен быть зарегистрирован в родительском домене. В самом деле, единственный способ узнать, кто обслуживает домен internal.domain.net – спросить об этом у DNS-сервера домена domain.net. Так что о выделении поддомена необходимо договариваться с провайдером. Регистрацией в доменах первого уровня занимаются выделенные организации. Например, домен в зоне .ru необходимо регистрировать в компании АНО «Региональный Сетевой Информационный Центр» (RU-CENTER).

Правила требуют, чтобы при регистрации домена было указано не менее двух DNS-серверов, которые будут его обслуживать. Из всех зарегистрированных серверов (записей типа

NS в родительской зоне) только одна соответствует т.н. первичному (master) серверу, а остальные – т.н. вторичным (slave). Для внешнего пользователя вторичный сервер не отличается от первичного: он столь же авторитетен в ответе на запрос об именах его домена. Отличия только в способе администрирования. Все изменения вносятся в зоны первичного сервера, а вторичный только кеширует эти зоны, целиком получая их по специальному межсерверному протоколу. Полученная зона складывается в файл, редактировать который бессмысленно: первичный сервер при изменении зоны рассылает всем своим вторичным указание скачать её заново. Право на скачивание зоны можно ограничить настройкой allow-transfer (как правило, в ней перечисляются адреса вторичных серверов). Вот слегка отредактированная цитата из руководства по Bind, описывающая задание вторичного сервера в файле настроек:

```
// We are a slave server for eng.example.com
zone "eng.example.com" { type slave;
file "slave/eng.example.com";
// IP address of eng.example.com master server masters { 192.168.4.12; };
};
```

Вторичный сервер хорошо размещать где-нибудь подальше от первичного, во всяком случае, в другой сети - так повышается надёжность обработки запроса (если один сервер недоступен, возможно, ответит второй) и возрастает скорость распространения записей по кешам промежуточных серверов.

Проверку работоспособности, доступности и вообще самочувствия DNS-сервера лучше всего делать утилитой dig из пакета bind-utils. Это очень мощная утилита, выдающая максимум информации о том, что происходило с запросом (запрашивая обратное преобразование, не забудьте добавить ключ -x). Более лаконична, но не менее мощна утилита host из того же пакета. Упоминаемую старыми руководствами по UNIX утилиту nslookup использовать не рекомендуется.

```
dig jack.internal.domain.net
<<>> DiG 9.2.4rc5 <<>> jack.internal.domain.net
;; global options:   printcmd
```

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32751

;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:

;jack.internal.domain.net. IN A

;; ANSWER SECTION:

jack.internal.domain.net. 86400 IN A 10.11.12.100

;; AUTHORITY SECTION:

internal.domain.net. 86400 IN NS server.internal.domain.net.

;; ADDITIONAL SECTION:

server.internal.domain.net. 86400 IN A 10.11.12.1

;; Query time: 37 msec

;; SERVER: 127.0.0.1#53(127.0.0.1)

;; WHEN: Sun Aug 22 16:07:17 2004 ;; MSG SIZE rcvd: 95

Наконец, для выяснения административной принадлежности тех или иных доменов и сетей можно воспользоваться утилитой whois, которая обращается к специальной сетевой базе данных (не имеющей отношения к DNS).

В состав репозитория РЕД ОС включен DNS-сервер bind, поддерживающий стандарты RFC 1034,1035, 2136. Поддерживает реализацию хост-систем на базе стека протоколов Internet, описанных в RFC 1123. Поддерживает механизм инкрементального переноса зон IXFR, описанный в RFC 1995 и механизм извещения ведомых серверов NOTIFY, RFC 1996.

Подготовка сервера

Устанавливаем все обновления:

Если вы используете РЕД ОС 7.1 или 7.2, выполните команду:

yum update

Если вы используете РЕД ОС 7.3 и старше, выполните команду:

dnf update

Устанавливаем утилиту для синхронизации времени, отключаем cronyd и запускаем ntpd:

для РЕД ОС 7.1 или 7.2:

yum install ntp
systemctl disable chronyd
systemctl enable ntpd
systemctl stop chronyd
systemctl start ntpd
для РЕД ОС 7.3 и старше:
dnf install ntp
systemctl disable chronyd
systemctl enable ntpd
systemctl stop chronyd
systemctl start ntpd
Настраиваем временную зону:
cp /usr/share/zoneinfo/Europe/Moscow /etc/localtime
* в данном примере выбрано московское время.
И синхронизируем время с внешним сервером:
ntpdate -s time.yandex.ru

Установка и запуск BIND

Устанавливаем DNS-сервер, если он ещё не установлен, следующей командой:
для РЕД ОС 7.1 или 7.2:
yum install bind
для РЕД ОС 7.3 и старше:
dnf install bind
Разрешаем автозапуск:
systemctl enable named
Запускаем сервис имен:
systemctl start named
И проверяем, что он работает корректно:
systemctl status named

Задание: *выполните указанные действия и покажите результат преподавателю!*

Базовая настройка DNS-сервера

Открываем на редактирование конфигурационный файл bind консольным редактором vi:
vi /etc/named.conf
и редактируем следующее в блоке options (переход редактора vi в режим вставки – клавиша i):

listen-on port 53 { 127.0.0.1; 192.168.0.1; };
listen-on-v6 port 53 { none; };
allow-query { any; };
forward first;
forwarders { 77.88.8.8; };

Здесь 192.168.0.1 – IP-адрес нашего DNS-сервера, на котором он будет принимать запросы. allow-query разрешает выполнять запросы всем, но из соображений безопасности можно ограничить доступ для конкретной сети, например, вместо any написать 10.10.1.0/24. listen-on-v6 port 53 присвоим значение none, тем самым отключив ipv6. forward с параметром first указывает, DNS-серверу пытаться разрешать имена с помощью DNS-серверов, указанных в параметре forwarders, и лишь в случае, если разрешить имя с помощью данных серверов не удалось, то будет осуществлять попытки разрешения имени самостоятельно. forwarders перенаправляем запросы, которые сами не обрабатываем, на сервер Яндекса.

После завершения вставки нажмите Esc для возврата в командный режим, и наберите команду :wq для сохранения и закрытия файла.

Создание локальных зон DNS

Создайте папку с мастер-зонами:

mkdir /var/named/master

Задание: *создайте прямую зону с именем следующего формата Ваше_имя-Ваша-фамилия.ru аналогично описанному далее примеру. В именах, используемых в этой зоне, должно фигурировать её имя. Используйте следующие IP-адреса:*

1) *Для основной записи A и www – адрес 192.168.10.N, где N – Ваш номер по списку;*
2) *Для серверов имён адреса 192.168.10.(N+10), 192.168.10.(N+20), 192.168.10.(N+30);*
3) *Для имени test адрес 192.168.10.N+5.*

Создайте прямую DNS-зону консольным редактором nano.

nano /var/named/master/example.org

Пример её содержимого с описанием ниже.

$TTL 86400 ;
example.org. IN SOA ns01.example.org. root.example.org.

```
(
                1               ; Serial
                600             ; Refresh
                3600            ; Retry
                1w              ; Expire
                360             ; Minimum TTL
                )
        IN      NS      ns01.example.org.
        IN      NS      ns02.example.org.
        IN      NS      ns03.example.org.
        IN      A       192.168.10.20
ns01    IN      A       192.168.10.50
ns02    IN      A       192.168.10.60
ns03    IN      A       192.168.10.70
www     IN      A       192.168.10.20
test    IN      A       192.168.10.12
```

Рассмотрим подробнее написанное:

$TTL 3600 – Time to live время жизни, по умолчанию 1 день. По достижении установленного времени, кеширующий сервер запрашивает DNS сервер, содержащий доменную зону, информацию о зоне. И при необходимости обновляет записи.

example.org IN SOA ns01.example.org. root.example.org. зона обслуживания, адрес корневого сервера для зоны, акаунт её админа.

1; Serial – её серийный номер DNS записи.

600; Refresh – указывает подчинённым DNS серверам как часто им обращаться, для поиска изменений к master-серверу.

3600; Retry – говорит о том, сколько Slave-сервер должен подождать, прежде чем повторить попытку.

1w; Expire – максимальный срок жизни записей, после которой они потеряют актуальность (1 неделя).

300; Minimum TTL -минимальный срок жизни записи 5 мин.

NS ns01.example.org. – NS сервер который обслуживает эту зону.

NS ns02.example.org. – NS сервер который обслуживает эту зону.

NS ns03.example.org. – NS сервер который обслуживает эту зону.

A 192.168.10.20 – если требуется попасть по адресу example.org, то клиенту будет выдан этот IP-адрес.

ns01 A 192.168.10.50 – записи для поиска наших NS серверов.

ns02 A 192.168.10.60

ns03 A 192.168.10.70

test A 192.168.10.12 – если клиент запрашивает адрес test.example.org, DNS выдаст IP 192.168.10.12.

Чтобы сохранить созданный редактором nano файл, воспользуйтесь командами в нижнем меню (CTRL+горячая клавиша).

Задание. *Назначьте владельца и права по схеме ниже*.

chown -R root:named /var/named/master

chmod 0640 /var/named/master/*

В конфигурационный файл named.conf добавьте следующее:

```
zone "example.org" {
 type master;
 file "master/example.org";
 };
```

* где example.org — имя зоны, которую будет обслуживать наш DNS-сервер. Это и есть домен, для которого bind будет хранить записи. Должен совпадать по имени с файлом зоны в папке master.

Описание опций настройки зоны:

type <тип зоны> (в нашем случае первичная – значит master). Другие варианты – slave, stub, forward.

file <путь_к_файлу> с записями зоны. В данном примере указан относительный путь – то есть файл находится по пути master/test.local, который начинается относительно рабочей директории (по умолчанию - /var/named/). Таким образом, полный путь до файла - /var/named/master/test.local. Блок zone создаётся для каждой зоны.

Чтобы настройки применились, необходимо перезапустить службу командой systemctl restart named. Если Вы всё сделали без ошибок, интерпретатор не выдаст никаких сообщений, в

противном случае предложит просмотреть статус службы и прочитать сообщения об ошибках.

Для проверки работоспособности сервера с другого компьютера сети (например, на Windows) выполняем команду:

> nslookup test.example.org 192.168.0.1

Данной командой мы пытаемся узнать IP-адреса сайта test.example.org через сервер 192.168.0.1.

Должно получиться, примерно, следующее:

```
Server:      192.168.0.1
Address:     192.168.0.1#53

Name:        test.example.org
Address:     192.168.10.12
```

Если другого компьютера сети нет, можно сделать проверку и с самого сервера, использовав в этом случае его первый адрес в файле named.conf – 127.0.0.1.

Задание. *Проверьте работоспособность сервера указанным способом и покажите результат преподавателю.*

Создание обратной зоны

Задание. *Создайте обратную зону для созданной ранее прямой зоны описанным ниже способом и проверьте работу обратного DNS-запроса командой nslookup <IP-адрес в зоне прямого просмотра> <IP-адрес сервера>. Покажите результат преподавателю.*

В конфигурационный файл named.conf добавьте следующее:

```
zone "10.168.192.in-addr.arpa" {
   type master;
   file "master/10.168.192.zone";
};
```

Создайте обратную DNS-зону.

nano /var/named/master/10.168.192.zone

Пример её содержимого с описанием ниже.

```
$TTL    3600
@  IN      SOA     example.org. root.example.org (
                   20060204        ; Serial
                   3600            ; Refresh
```

```
                900             ; Retry
                3600000         ; Expire
                3600 )          ; Minimum
@  IN      NS      localhost.
50 IN      PTR     ns01
60 IN      PTR     ns02
70 IN      PTR     ns03
20 IN      PTR     www
12 IN      PTR     test
```

Чтобы настройки применились, необходимо перезапустить службу.

systemctl restart named

Журналирование Bind

Способ 1

По умолчанию, сервер Bind хранит журналы в файле:

/var/named/data/named.run

Для его непрерывного просмотра вводим следующую команду:

tail -f /var/named/data/named.run

Степень детализации журнала можно настроить в конфигурационном файле:

```
logging {
  channel default_debug {
    file "data/named.run";
  severity dynamic;
  };
};
```

* где file – путь к log-файлу; severity – уровень чувствительности к возникающим событиям.

Возможны следующие варианты для severity:

critical — критические ошибки;

error – ошибки и выше (critical);

warning – предупреждения и выше. Предупреждения не говорят о наличии проблем в работе сервиса, однако это такие событтия, которые могут привести к ошибкам, поэтому не стоит их игнорировать;

notice – уведомления и выше;
info — информация;
debug — отладка (подробный лог);
dynamic — тот же debug.
Способ 2
Создайте каталог для логов:
mkdir /var/log/bind

В разделе logging задаются 2 параметра channel (можно и больше двух – на Ваше усмотрение), эти параметры дословно можно назвать "канал" записи. Каждый канал определяет имя канала и настройки параметров записи (что записывать, а что - нет и куда писать).

Директива category задает какую категорию сообщений в какой канал отправлять.

Исходя из этого, мы имеем: запись стандартной информации в канал misc, а приходящие запросы посылаются в канал query. При этом, если файлы журнала достигают 4Мб (size 4m), он переименовывается добавлением к имени .1 и начинается запись в новый журнал, числа в конце других журналов увеличиваются. Журналы с номером, более указанного в version (в нашем случае 4) удаляются.

Параметры print* определяют заносить ли в журнал время появления, важность и категорию информации. Более подробно про настройки раздела logging можно почитать в man (5) named.conf.

Добавьте конфигурацию логирования в файл /etc/named.conf:

```
// настройки логирования
logging {
        channel "misc" {
                file "/var/log/bind/misc.log" versions 4 size 4m;
                print-time yes;
                print-severity yes;
                print-category yes;
        };

        channel "query" {
                file "/var/log/bind/query.log" versions 4 size 4m;
```

```
        print-time yes;
        print-severity no;
        print-category no;
    };

    category default {
        "misc";
    };

    category queries {
        "query";
    };
};
```

Проверка работоспособности

Для проверки работоспособности и правильной конфигурации DNS-сервера, нужно сделать к нему несколько запросов. Для этого можно воспользоваться консольными утилитами ping, dig или nslookup.

ВАЖНО!

Клиент, на котором выполняются запросы должен быть настроен на использование вашего DNS сервера, подробнее смотрите в настройках сетевого адаптера.

Утилита ping, при указании доменного имени, разрешает доменное имя с помощью DNS. Если пинг к хосту, указанному в DNS, выполняется, то A запись данного хоста корректна.

С помощью dig или nslookup можно проверить обратные записи или иные специфические записи DNS-сервера. Например, dig SRV _ldap._tcp.ad.test сделает запрос SRV-записи на сервере в домене ad.test. Должен вывести ip-адрес хоста с LDAP сервисом.

nslookup 192.168.10.50

Запрос должен вывести имя узла для введенного IP. Тем самым проверив обратную (PTR) запись.

Подготовка master сервера

Открываем на редактирование конфигурационный файл bind:

vi /etc/named.conf

и редактируем следующее в блоке options:

allow-recursion { none; };

Находим и закомменитуем строку

dnssec-validation yes;

Рассмотрим подробнее то, что мы написали:

allow-recursion { none; } – отключаем использование рекурсивных запросов т.к. это сильно снижает скорость работы, да и нее имеет смысла опрашивать вышестоящие DNS сервера по поводу зоны, которую сами же и обслуживаем.

Добавьте строки в блоки ваших локальных зон

allow-transfer {<ip адрес реплики>;};

allow-update {<ip адрес реплики>;};

notify yes;

Давайте разберемся, что мы туда добавили

allow-transfer {<ip адрес реплики>;} – разрешаем передачу данной зоны на остальные наши DNS сервера.

allow-update {<ip адрес реплики>;} – разрешаем обновление.

notify yes – включаем автоматическое уведомление подчиненных серверов об обновлении файла настроек DNS зоны.

Перезагрузите bind.

systemctl restart bind

Настройка slave сервера

Устанавливаем все обновления.

Если Вы используете РЕД ОС версии 7.1 или 7.2, выполните команду:

yum update

Если Вы используете РЕД ОС версии 7.3 и старше, выполните команду:

dnf update

Устанавливаем утилиту для синхронизации времени, отключаем cronyd и запускаем ntpd:

для РЕД ОС версии 7.1 или 7.2:

yum install ntp

systemctl disable chronyd

systemctl enable ntpd

systemctl stop chronyd

systemctl start ntpd

для РЕД ОС версии 7.3 и старше:

dnf install ntp

systemctl disable chronyd

systemctl enable ntpd

systemctl stop chronyd

systemctl start ntpd

Синхронизируем время с внешним сервером:

ntpdate ntp.ipa.test

Устанавливаем DNS-сервер, если не установлен, следующей командой:

для РЕД ОС версии 7.1 или 7.2:

yum install bind

для РЕД ОС версии 7.3 и старше:

dnf install bind

Разрешаем автозапуск:

systemctl enable named

Запускаем сервис имен:

systemctl start named

И проверяем, что он работает корректно:

systemctl status named

Открываем на редактирование конфигурационный файл bind:

vi /etc/named.conf

И редактируем следующее в блоке options:

listen-on port 53 { 127.0.0.1; <ip данного сервера>;};

listen-on-v6 port 53 { none; };

allow-query { any; };

forward first;

forwarders {8.8.8.8;};

allow-recursion { none; };

Находим и закомментируем строку

dnssec-validation yes;

Добавляем туда запись, только на этот раз она будет немного отличаться:

```
zone "example.org" IN {
        type slave;
        file "/var/named/slaves/slave.example.org";
```

masters { <ip адрес мастера>; };
allow-transfer {"none";};
};

Рассмотрим подробнее изменения:

type slave; - тип зоны подчинённая

file /var/named/slaves/slave.example.org" – путь к файлу с настройками, в этот раз в имени файла указано slave.example.org чтобы было понятно на каком сервере вы находитесь.

masters { <ip адрес мастера>; } – IP адрес Мастер сервера, откуда будет производиться запрос файлов с настройками DNS зон.

allow-transfer {none;} – отключаем передачу зоны другим серверам, чтобы нельзя было получить все записи в домене

Впишите вышеописанные блоки для каждой зоны. Перезагрузите DNS сервер.

systemctl restart named

Теперь должны появиться файлы DNS зон по пути, который указан.

ВАЖНО!

Каждый раз, когда в настройки зоны на мастере вносятся изменения, к серийному номеру прибавляется единица, это делается для того чтобы подчиненные сервера увидели изменения в записях и приняли обновленный файл зоны, если после внесения изменений в файл, серийный номер остался прежним или уменьшился, то подчиненные DNS сервера НЕ станут подтягивать обновления, считая, что на мастер сервере изменений не было!

Для редактирования серийного номера DNS зоны, откройте её файл и отредактируйте параметр serial.

Пример файла зоны:

```
...
@       IN      SOA     ns0.ipa.test. postmaster.ipa.test. (
                2018042804  ;serial
...
```

Это значение нужно увеличивать каждый раз при редактировании файла доменной зоны. Оно записано в формате YY-MM-DD + наращиваемый порядковый номер.

Для проверки работоспособности сервера с другого компьютера сети, если он имеется, выполняем команду:

nslookup test.example.org <ip адрес реплики>

* данной командой мы пытаемся узнать IP-адреса сайта test.example.org через сервер репликации DNS.

Должно получиться, примерно, следующее:

Server: 192.168.0.1
Address: 192.168.0.1#53
Name: test.example.org
Address: 192.168.10.12

Контрольные вопросы

1. Что такое DNS, и зачем она нужна?
2. Что понимается под разрешением имён?
3. Чем отличается рекурсивный запрос от итеративного?
4. Чем отличается прямой запрос от обратного?
5. Что такое DNS-зона?
6. Какие виды зон Вы знаете?
7. Какой сервер называется авторитетным?
8. Что такое ресурсная запись? Какие виды записей Вы знаете?
9. Какая ресурсная запись является основной?
10. Какая ресурсная запись используется для обратного запроса?

Список литературы

1. Кобылянский, В. Г. Операционные системы, среды и оболочки : учебное пособие / В. Г. Кобылянский. – Новосибирск : Новосибирский государственный технический университет, 2018. – 80 с. – URL: https://biblioclub.ru/index.php?page=book&id=576354 (дата обращения: 07.03.2022). – Режим доступа: по подписке. – Текст : электронный.
2. Курячий, Г.В. Операционная система Linux : учебник / Г.В. Курячий, К.А. Маслинский. – 2-е изд., исправ. – Москва : Национальный Открытый Университет «ИНТУИТ», 2016. – 451 с. - URL: https://biblioclub.ru/index.php?page=book&id=578058 (дата обращения: 02.02.2021). – Режим доступа: по подписке. – Текст : электронный.
3. Куль, Т. П. Операционные системы : учебное пособие / Т. П. Куль. – Минск : РИПО, 2019. – 312 с. - URL: https://biblioclub.ru/index.php?page=book&id=599951 (дата обращения: 05.03.2022). Режим доступа: по подписке. – Текст : электронный.
4. Основы администрирования информационных систем : учебное пособие / Д. О. Бобынцев, А. Л. Марухленко, Л. О. Марухленко [и др.]. – Москва ; Берлин : Директ-Медиа, 2021. – 201 с. - :URL: https://biblioclub.ru/index.php?page=book&id=598955 (дата обращения: 05.03.2022). – Режим доступа: по подписке. – Текст : электронный.