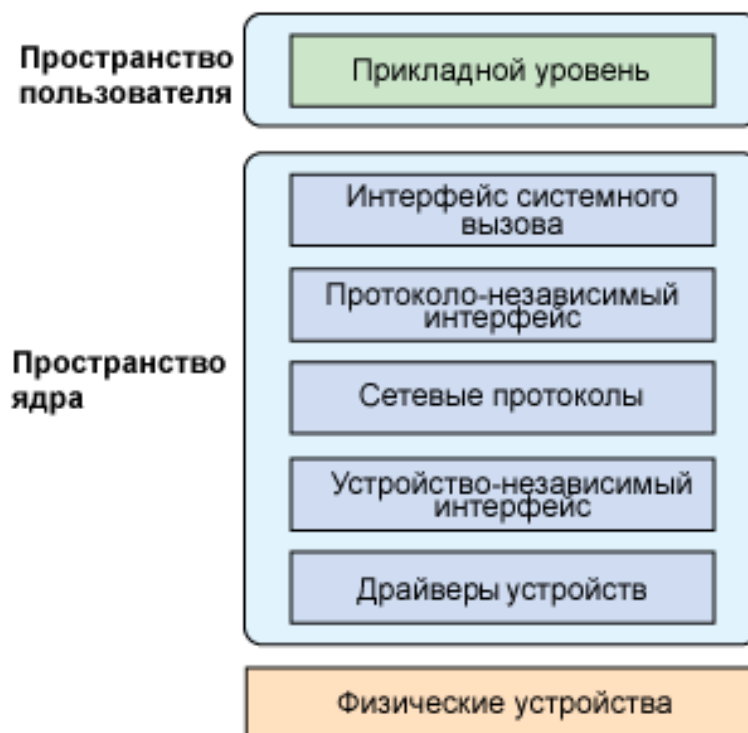


А.С. Ватьян, А.Д. Береснев

АДМИНИСТРИРОВАНИЕ В ИНФОРМАЦИОННЫХ СИСТЕМАХ



Санкт-Петербург

2019

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

УНИВЕРСИТЕТ ИТМО

А.Д. Береснев, А.С. Ватьян
АДМИНИСТРИРОВАНИЕ В ИНФОРМАЦИ-
ОННЫХ СИСТЕМАХ

УЧЕБНОЕ ПОСОБИЕ

РЕКОМЕНДОВАНО К ИСПОЛЬЗОВАНИЮ В УНИВЕРСИТЕТЕ ИТМО
по направлению подготовки 09.03.02 Информационные системы и тех-
нологии

в качестве учебного пособия для реализации основных профессиональных об-
разовательных программ высшего образования бакалавриата,

 УНИВЕРСИТЕТ ИТМО

Санкт-Петербург
2019

Береснев А.Д., Ватьян А.С., Администрирование в информационных системах – СПб: Университет ИТМО, 2019. – 45 с.

Рецензент(ы):

Маятин Александр Владимирович, кандидат педагогических наук, доцент, доцент факультета информационных технологий и программирования, Университета ИТМО.

В учебном пособии содержится информация о инструментальных средствах администрирования ОС Linux, настройки компонент сетевого стека и типовых сервисов.



Университет ИТМО – ведущий вуз России в области информационных и фотонных технологий, один из немногих российских вузов, получивших в 2009 году статус национального исследовательского университета. С 2013 года Университет ИТМО – участник программы повышения конкурентоспособности российских университетов среди ведущих мировых научно-образовательных центров, известной как проект «5 в 100». Цель Университета ИТМО – становление исследовательского университета мирового уровня, предпринимательского по типу, ориентированного на интернационализацию всех направлений деятельности.

© Университет ИТМО, 2019
© Береснев А.Д., Ватьян А.С., 2019

Оглавление

1. ПОНЯТИЕ ОБ АДМИНИСТРИРОВАНИИ СИСТЕМ	6
2. БАЗОВАЯ КОНФИГУРАЦИЯ И НАСТРОЙКИ.....	8
2.1. Настройка /etc/hosts File	8
2.2. Сетевая диагностика	8
2.3. Команда traceroute	10
2.4. Команда mtr	10
3. ДИАГНОСТИКА СИСТЕМЫ.....	12
3.1. Проверка текущего использования памяти	12
3.2. Использование ввода-вывода с помощью vmstat	13
3.3. Мониторинг процессов, памяти и использования ЦП с помощью htop	13
4. УПРАВЛЕНИЕ ФАЙЛОВОЙ СИСТЕМОЙ	15
4.1. Загрузка файлов на удаленный сервер.....	15
4.2. Защита файлов на удаленном сервере.....	16
4.3. Символические ссылки.....	16
5. УПРАВЛЕНИЕ В LINUX СИСТЕМАХ	17
5.1. Базовые команды.....	17
5.2. Управление пакетами.....	17
5.3. Поиск имен пакетов и информации	19
6. ОБРАБОТКА ТЕКСТА.....	23
6.1. Поиск строки в файлах с помощью grep.....	23
6.2. Поиск и замена по всей группе файлов	24
6.3. Редактирование текста.....	25
7. ВЕБ-СЕРВЕРЫ И HTTP-ПРОБЛЕМЫ	26
7.1. Веб-серверы	26
7.2. Как выбрать веб-сервер?	26
7.3. Журналы Apache.....	27
8. ПРАВА ПОЛЬЗОВАТЕЛЯ И ГРУППЫ	28
8.1. Что такое права пользователя и группы?	28
8.2. Чтение, запись и выполнение.....	28
8.3. Просмотр разрешений файлов	28
8.4. Работа с пользователями, группами и справочниками	29
8.5. Использование Sudo.....	31
8.6. Работа с группами	31
8.7. Создание и удаление каталогов	32
8.8. Изменение прав доступа на каталог и файл	33
9. НАСТРОЙКА СРЕДСТВ СЕТИ В LINUX.....	34
9.1. Устройство сетевого стека в Linux	34
9.2. Сетевые настройки ядра Linux.....	36

9.3. Настройка параметров сетевого интерфейса	37
9.4. Настройка маршрутов по умолчанию	38
9.5. Настройка DNS	40
9.6. Настройка через конфигурационные файлы	40
Литература	42

1. Понятие об администрировании систем

В настоящее время развитие информационных и телекоммуникационных технологий приводит к существенному изменению способов организации предприятий и управления. Эти тенденции получили объединены под еще неустоявшимся термином – «цифровизация».

В основу этого явления легло развитие сетевых технологий и технологий обработки данных. Развитие первой группы технологий связано с повсеместным распространением глобальной сети Интернет, внедрением новых протоколов, таких как IPv6, QUIC, 5G, а так же интернета вещей – IoT. Следует отметить, что практически завершена замена цифровой передачей традиционных аналоговых решений, таких как видеонаблюдение и телефония.

Современные технологии позволяют строить распределенные системы, использующие уделанные вычислительные ресурсы, быстро строить сложные конвергентные решения, использовать сторонние технологические системы как поставщики данных и сервисов их обработки. Новые возможности реализуются в бизнесе, дистанционном обучении, государственном управлении, медицине и т.п. И главным свойством систем, делающим возможным все это, является сетевая связанность.

Изменился и подход к пониманию роли информационных технологий в современном бизнесе. Традиционно понятие администрирование информационной системы сводилось к обеспечению поддержки существующих бизнес-процессов специфичными компьютерными средствами. Современный подход, в духе ITSM, сосредоточен на предоставлении бизнесу *сервиса*. То есть не только технологических но и организационных и бизнес решений, дающих эффект за счет или устранения ограничений или создания новых возможностей с одновременным обеспечением доступности, непрерывности, безопасности и производительности.

Однако, даже если мы имеем дело с облачными решениями, предоставляемыми как IaaS, SaaS или PaaS, в их основе лежат конкретные узлы с установленными операционными системами и настроенным сетевым стеком. Основным понятиям, принципам и приемам настройкам операционной системы Linux и сетевого стека посвящено это учебное пособие.

В конце пособия приводится список рекомендуемых источников литературы. В результате освоения материалов пособия обучающийся приобретает следующие профессиональные компетенции, а также умения и навыки:

- знание основных архитектур современных информационных коммуникационных систем, принципов работы распространенных технологий и стандартов информационных сетей,
- способность анализировать информацию о конфигурации модулей сетевого стека,

- умение осуществлять начальное конфигурирование основных компонентов телекоммуникационных сетей,

- умение осуществлять диагностику конфигураций телекоммуникационных сетей, пользоваться типовыми техническими средствами для конфигурирования и диагностики компонентов телекоммуникационных сетей.

Распределение трудозатрат студентов в аудитории и в процессе СРС представлено в соответствии с программами изучаемой дисциплины. Методическое пособие рекомендуется применять как дополнительный источник информации при подготовке к и выполнению практических работ.

Для дисциплины «Телекоммуникационные системы и технологии» на подготовку к практическим работам рекомендуется отводить по 20 часов, а для дисциплины «Администрирование в операционной системе Linux» по 25 часов из 47 запланированных для СРС в каждом модуле.

Материалы пособия используются студентами в качестве источника справочной информации при выполнении практических работ и подготовке к защите отчетов (для курса «Телекоммуникационные системы и технологии» - работы 1,3,6,7, для курса «Администрирование в операционной системе Linux» - работы 2,3,4).

2. Базовая конфигурация и настройки

2.1. Настройка /etc/hosts File

Файл /etc/hosts содержит список IP-адресов с соответствующими именами хостов. Это позволяет указывать имена хостов для IP-адреса в одном месте на локальном компьютере, а затем подключать несколько приложений к внешним ресурсам через их имена хостов. Система файлов хостов предшествует DNS, а файлы hosts всегда проверяются до запроса DNS. В результате /etc/hosts могут быть полезны при поддержке небольших «внутренних» сетей, для разработки и для управления кластерами.

Некоторые приложения требуют, чтобы машина правильно идентифицировала себя в файле /etc/hosts. В результате мы рекомендуем настроить файл /etc/hosts вскоре после развертывания. Вот пример файла:

/etc/hosts

```
127.0.0.1 localhost.localdomain localhost
103.0.113.12 username.example.com username
```

Вы можете указать количество имен хостов в каждой строке, разделенных пробелами. Каждая строка должна начинаться с одного и только одного IP-адреса. В приведенном выше примере замените 103.0.113.12 на IP-адрес вашего устройства.

Рассмотрим несколько дополнительных /etc/hosts записей:

```
198.51.100.30 example.com
192.168.1.1 mysrv.example.com
```

В этом примере все запросы для hostname или домена example.com будут разрешаться на IP-адрес 198.51.100.30, который обходит записи DNS для example.com и возвращает альтернативный веб-сайт.

Вторая запись указывает системе посмотреть на 192.168.1.1 для домена mysrv.example.com. Такие записи хоста полезны для использования сетей «частного» или «обратного канала» для доступа к другим серверам в кластере без необходимости отправки трафика в общедоступную сеть.

2.2. Сетевая диагностика

В этом разделе мы рассмотрим некоторые основные команды Linux, которые помогут вам оценить и диагностировать сетевые проблемы. Если вы предполагаете проблемы с подключением, добавление результата из соответствующих команд в лист поддержки может помочь диагностировать вашу проблему. Это особенно полезно в случаях, когда проблемы с сетью являются нерегулярными и плохо воспроизводятся.

Команда ping

Команда ping проверяет соединение между локальной машиной и удаленным адресом или машиной. Следующие команды «ping»:

```
ping de.ifmo.ru
ping 77.234.213.242
```

Эти команды отправляют небольшое количество данных (ICMP-пакет) на удаленный хост и ждут ответа. Если система может установить соединение, она будет сообщать о «времени прохождения в оба конца» для каждого пакета. Вот пример вывода четырех писем на google.com:

```
ping de.ifmo.ru
Обмен пакетами с cde.ifmo.ru [77.234.213.242] с 32 байтами данных:
Ответ от 77.234.213.242: число байт=32 время=24мс TTL=56
Ответ от 77.234.213.242: число байт=32 время=22мс TTL=56
Ответ от 77.234.213.242: число байт=32 время=27мс TTL=56
Ответ от 77.234.213.242: число байт=32 время=22мс TTL=56
```

Поле времени (time) указывает в миллисекундах продолжительность поездки в оба конца для отдельного пакета. Когда вы собрали необходимое количество информации, используйте Control + C, чтобы прервать процесс. После прекращения процесса вам будет предоставлена некоторая статистика. Это выглядит так:

```
Статистика Ping для 77.234.213.242:
  Пакетов: отправлено = 4, получено = 4, потеряно = 0
  (0% потерь)
Приблизительное время приема-передачи в мс:
  Минимальное = 22мсек, Максимальное = 27 мсек, Среднее = 23
мсек
```

Следует отметить несколько важных параметров.

Пакетная потеря или расхождение между количеством отправленных пакетов и количеством успешно завершенных пакетов. Это число показывает процент упавших пакетов.

Время в минутах (rtt) статистики в итоговой строке сообщает информацию обо всех ответах на пинг. Для этого пинга мы видим, что самый быстрый пакет в оба конца (min) составил 22 миллисекунды. Средний кругооборот (средний) составил 23 миллисекунды. Самый длинный пакет (max) занял 27 миллисекунды.

Команда ping полезна как неформальный инструмент диагностики для измерения латентности между точками сети и как инструмент, обеспечивающий простое подключение к удаленному серверу.

2.3. Команда traceroute

Команда traceroute расширяет функциональность команды ping. Она предоставляет отчет о пути, который пакеты проходят при переходе с локального компьютера на удаленный компьютер. Каждый шаг (промежуточный сервер) в пути называется прыжком. Информация о маршруте полезна при устранении проблемы с сетью: если в одном из первых нескольких переходов есть потеря пакетов, проблема часто связана с локальной сетью пользователя (LAN) или поставщиком услуг Интернета (ISP). Напротив, если есть потеря пакетов в конце маршрута, проблема может быть вызвана проблемой с подключением сервера.

Ниже приведен пример вывода команды tracert (ее аналог traceroute):

```
tracert de.ifmo.ru

Трассировка маршрута к cde.ifmo.ru [77.234.213.242]
с максимальным числом прыжков 30:

 1    5 ms    3 ms    3 ms    192.168.0.1
 2   48 ms   51 ms   23 ms   172.31.254.26
 3    6 ms   10 ms    7 ms   5x18x6x245.static-business.spb.ertelecom.ru [5.1
8.6.245]
 4   17 ms   15 ms   15 ms   ertelecom.msk.ru [194.190.254.142]
 5   17 ms   19 ms   15 ms   m9-3-gw.msk.runnet.ru [194.190.254.141]
 6   27 ms   23 ms   24 ms   msk-m9-1-gw.RUNNet.ru [194.85.40.205]
 7   27 ms   28 ms   26 ms   vuztc.spb.runnet.ru [194.190.255.170]
 8   25 ms   23 ms   23 ms   77.234.193.6
 9   24 ms   22 ms   28 ms   77.234.212.246
10   44 ms   20 ms   24 ms   77.234.212.245
11   38 ms   34 ms   38 ms   cde.ifmo.ru [77.234.213.242]
```

Часто имена хостов и IP-адреса с каждой стороны неудачного перехода полезны при определении того, кто управляет машиной, на которой происходит ошибка маршрутизации. Неисправные прыжки обозначаются строками с тремя звездочками (* * *).

Добавление вывода traceroute к методам поддержки Linode иногда полезно при попытке диагностики сетевых проблем. Вы также можете перенаправить информацию traceroute своему интернет-провайдеру (ISP), если вы подозреваете, что проблема с подключением связана с сетью вашего интернет-провайдера. Запись информации traceroute особенно полезна, если вы столкнулись с проблемой прерывистой проблемы.

2.4. Команда mtr

Команда mtr, подобно инструменту traceroute, предоставляет информацию о маршруте, по которому идет интернет-трафик между локальной системой и удаленным хостом. Однако mtr предоставляет дополнительную информацию о времени прохождения пакета для пакета. В некотором смысле вы можете думать о mtr как о комбинации traceroute и ping.

Ниже приведен пример вывода команды mtr:

```

mtr --report lib.ru
HOST: gate
Loss%  Snt  Last   Avg   Best  Wrst  StDev
 1. 77.234.215.129    0.0%   10    0.5   0.8   0.4   1.4   0.4
 2. 77.234.193.5     0.0%   10    0.7  29.0   0.7  151.7  58.8
 3. s14-1-gw.spb.runnet.ru 0.0%   10    1.5   1.5   0.7   3.9   0.9
 4. msk-m9-1-gw.RUNNet.ru 0.0%   10   11.1  11.2  10.7  11.8   0.4
 5. rostelecom.msk.runnet.ru 0.0%   10   17.0  15.0  13.0  17.0   1.4
 6. msk-bgw1-xe-2-0-0-0.rt-comm. 0.0%   10   10.6  10.3   9.8  11.0   0.4
 7. msk-dsr8-tg9-1.rt-comm.ru 0.0%   10   12.6  12.8  11.6  16.9   1.5
 8. msk-dsr8-tg9-1.rt-comm.ru 0.0%   10   12.2  15.0  12.1  32.8   6.4
 9. lib.ru           10.0%   10   11.5  16.0  11.5  21.2   4.2

```

Как и команда `ping`, `mtr` отслеживает скорость соединения в реальном времени до выхода из программы с помощью `CONTROL + C`. Чтобы автоматически остановить `mtr` и сгенерировать отчет после десяти пакетов, используйте флаг `--report`:

```
mtr --report
```

Отметим, что `mtr` будет приостанавливаться в течение нескольких минут при генерации вывода.

3. Диагностика системы

3.1. Проверка текущего использования памяти

Чтобы узнать, сколько памяти ваша система использует в настоящее время, введите команду:

```
free -m
```

На Linode 2GB при обычном использовании вывод в ответ на команду должен выглядеть следующим образом:

```
Mem:          total        used        free        shared    buffers    cached
-/+ buffers/cache:    105        1785
Swap:         5493           0         5493
```

Этот вывод требует тщательного чтения для интерпретации. Из общего объема памяти (RAM) в 1999 система использует 954 мегабайта и имеет 1044 мегабайта свободных. Тем не менее, система также имеет 703 мегабайта «устаревших» данных, буферизированных и хранящихся в кеше. Операционная система «отбрасывает» кеш, если возникает необходимость в дополнительной памяти, и сохраняет кеш, пока такой необходимости нет. Для системы Linux обычным методом является то, что она оставляет старые данные в ОЗУ до тех пор, пока пространство не понадобится, поэтому не беспокойтесь, если только небольшой объем памяти «свободен».

В приведенном выше примере имеется 1782 мегабайта памяти, которые на самом деле свободны. Это означает, что 1782 мегабайта доступны для вашей системы, когда вы запускаете дополнительный процесс, или работающему приложению требуется больше памяти.

3.2. Использование ввода-вывода с помощью vmstat

Инструмент `vmstat` предоставляет информацию о памяти, использовании подкачки, ожидании ввода-вывода и активности системы. Это особенно полезно для диагностики проблем, связанных с I/O.

Если вы считаете, что у вас проблема ввода-вывода, выполните следующую команду:

```
vmstat 1 20
```

Эта команда запускает `vmstat` двадцать раз с интервалом 1 секунда, показывая текущее состояние системы. Ниже приводится пример вывода результата.

```
procs -----memory----- ---swap-- -----io----- --system-- -----cpu-----
r  b   swpd   free   buff  cache   si   so   bi   bo   in   cs  us  sy  id  wa  st
0  0     0 1062648 190564 576072    0    0    0    2    1    1  0  0 99  0  0
0  0     0 1062632 190564 576072    0    0    0    0 2060  14  0  1 99  0  0
0  0     0 1062632 190564 576072    0    0    0    0 2059  14  0  0 100  0  0
0  0     0 1062632 190564 576072    0    0    0    0 2131  17  0  0 100  0  0
0  0     0 1062632 190564 576072    0    0    0    0 2163  14  0  1 99  0  0
0  0     0 1062632 190564 576072    0    0    0    0 2118  12  0  0 100  0  0
0  0     0 1062632 190564 576072    0    0    0   12 2105  25  0  1 99  0  0
0  0     0 1062756 190564 576072    0    0    0    0 2096  12  0  1 99  0  0
0  0     0 1062756 190564 576072    0    0    0    0 2128    8  0  0 100  0  0
0  0     0 1062756 190564 576072    0    0    0    0 2107  16  0  0 100  0  0
0  0     0 1062756 190564 576072    0    0    0    0 2165  14  0  1 99  0  0
0  0     0 1062756 190564 576072    0    0    0   24 2121  18  0  1 99  0  0
0  0     0 1062756 190564 576072    0    0    0   40 2127  26  0  1 98  1  0
0  0     0 1062756 190564 576072    0    0    0    0 2243  16  0  1 99  0  0
0  0     0 1062756 190564 576072    0    0    0    0 2169  16  0  1 99  0  0
0  0     0 1062756 190564 576072    0    0    0    0 2065  14  0  0 100  0  0
0  0     0 1062756 190564 576072    0    0    0    0 2118  14  0  1 99  0  0
0  0     0 1062756 190564 576072    0    0    0    0 2127  16  0  1 99  0  0
0  0     0 1062756 190564 576072    0    0    0    0 2088  15  0  0 100  0  0
0  0     0 1062756 190564 576072    0    0    0    0 2089  16  0  1 99  0  0
```

Столбцы памяти и свапа предоставляют такую же информацию, что и команда «`free -m`», хотя и в немного более сложном для понимания формате. Наиболее релевантной информацией, созданной этой командой, является столбец `wa`, который является последним столбцом в большинстве реализаций. В этом поле отображается количество времени, затрачиваемого центральным процессором, который ожидает завершения операций ввода-вывода.

Если это число будет постоянно и значительно выше 0, стоит принять меры для решения проблемы использования ИО. Вывод `vmstat` иногда помогает поддерживать диагностику проблем.

3.3. Мониторинг процессов, памяти и использования ЦП с помощью `htop`

Если требуется более организованное представление текущего состояния вашей системы в реальном времени, мы рекомендуем использовать инструмент `htop`. Он установлен по умолчанию для большинства систем. Чтобы установить `htop`, выполните одну из следующих команд, в зависимости от используемого вами дистрибутива:

```
apt-get install htop
yum install htop
pacman -S htop
```

```
emerge sys-process/htop
```

Чтобы запустить программу, выполните следующую команду:

```
htop
```

Вы можете завершить программу в любое время, нажав клавиши F10 или Q.

Существует специфика htop, на которую следует обратить внимание.

График использования памяти отображает используемую память, буферную память и кэшированную память. Номера, отображаемые в конце этого графика, отражают общий объем доступной памяти и общую память объема в системе, как сообщается ядру.

Конфигурация по умолчанию htop представляет все потоки приложений как независимые процессы, что может быть неочевидно. Вы можете отключить эту функцию, выбрав опцию «Настройка» с помощью F2, затем «Параметры отображения», а затем переключитесь на «Скрыть пользовательские потоки».

Вы можете переключить представление «Дерево» с помощью клавиши F5, которая отображает процессы в иерархии и показывает, какие процессы были созданы другими процессами в организованном формате. Это полезно при диагностике проблемы, когда у вас возникают проблемы с различиями между процессами.

4. Управление файловой системой

4.1. Загрузка файлов на удаленный сервер

Если вы используете FTP-клиент, OpenSSH (который включен и активен во всех версиях Linode) позволяет вам использовать FTP-подобный интерфейс по протоколу SSH. Многие клиенты, включая WinSCP для Windows, Cyberduck для Mac OS X и Filezilla для Linux, OS X и Windows, поддерживают этот протокол известный как SFTP.

Если вы привыкли к FTP, SFTP будет вам очень хорошо знаком. По умолчанию, независимо от доступа пользователя к файловой системе в командной строке, этот пользователь также будет иметь доступ к SFTP.

Следует учесть влияние прав доступа к файлам при настройке доступа пользователей.

Вы также можете использовать утилиты Unix, включая `scp` и `rsync`, для безопасной передачи файлов в Linode. На локальной машине команда для копирования `team-info.tar.gz` будет выглядеть так:

```
scp team-info.tar.gz username@hostname.example.com:/home/username/backups/
```

За командой `scp` следует путь к файлу в локальной файловой системе, подлежащей передаче. Далее следуют имя пользователя и имя хоста удаленной машины, разделенные знаком «at» (`@`). За именем хоста должно стоять сдвоенное двоеточие (`:`) и путь на удаленном сервере, где должен быть загружен файл. Используем более обобщенный пример:

```
scp [source] [/path/to/local/file] [remote-username]@[remote-hostname]:[/path/to/remote/file]
```

Эта команда доступна по умолчанию для компьютеров OS X и Linux. Вы можете использовать его для копирования файлов в Linode, а также между удаленными серверами. Если вы используете SSH-ключи, вы можете использовать команду `scp` без ввода пароля для каждой передачи.

Синтаксис `scp` должен соответствовать следующей форме:

```
scp [source] [destination].
```

Вы можете скопировать файлы с удаленного хоста на локальную машину, изменив порядок путей в приведенном выше примере.

4.2. Защита файлов на удаленном сервере

Поскольку серверы Linode являются доступными в сети и часто имеют несколько разных пользователей, важно поддерживать безопасность файлов.

Мы предлагаем следующие рекомендации по обеспечению безопасности:

Разрешайте пользователям делать то, что им нужно. Сюда входят пользователи приложений.

Используйте только сервисы на общедоступных интерфейсах. Один из распространенных источников уязвимостей безопасности – это неиспользуемые демоны. Сюда входят серверы баз данных, серверы разработки HTTP и FTP-серверы.

Используйте SSH-соединения, когда это возможно, для защиты и шифрования передачи конфиденциальной информации.

4.3. Символические ссылки

Символическое связывание (англ. *symlinking*) позволяет вам создать объект в вашей файловой системе, который указывает на другой объект в вашей файловой системе. Это полезно, когда вам необходимо предоставить пользователям и приложениям доступ к определенным файлам и каталогам без реорганизации ваших папок. Таким образом, вы можете предоставить ограниченным пользователям доступ к вашим каталогам, доступным через Интернет, не перемещая DocumentRoot в свои домашние каталоги.

Чтобы создать символическую ссылку, выполните команду в следующем формате:

```
ln -s /home/username/config-git/etc-hosts /etc/hosts
```

Это создает ссылку файла `etc-hosts` в месте расположения файла `/etc/hosts` системы. В более общем виде:

```
ln -s [/path/to/target/file] [/path/to/location/of/sym/link]
```

Обратите внимание на следующие функции команды `link`.

Конечный термин, т.е. расположение ссылки, является необязательным. Если вы опустите ссылку, ссылка будет создана в текущем каталоге с тем же именем, что и файл, на который вы ссылаетесь.

При указании местоположения ссылки убедитесь, что путь не имеет конечной косой черты. Вы можете создать сим-ссылку, которая указывает на каталог, но сим-ссылки не могут заканчиваться косой чертой.

Вы можете удалить сим-ссылку, не затрагивая целевой файл.

Вы можете использовать относительные или абсолютные пути при создании сим-ссылки.

5. Управление в Linux системах

5.1. Базовые команды

В этом разделе приведен список основных команд для управления содержимым вашей файловой системы.

Для копирования файлов:

```
cp /home/username/todo.txt /home/username/archive/todo.01.txt
```

Это копирует todo.txt в папку архива и добавляет номер к имени файла. Если вы хотите рекурсивно скопировать все файлы и подкаталоги в каталог в другой каталог, используйте параметр -R. Эта команда выглядит так:

```
cp -R /home/username/archive/ /srv/backup/username.01/
```

Чтобы переместить файл или каталог, используйте команду

```
mv /home/username/archive/ /srv/backup/username.02/
```

Вы также можете использовать команду mv для переименования файла.

Чтобы удалить файл, используйте команду

```
rm scratch.txt
```

Это приведет к удалению файла scratch.txt из текущего каталога.

5.2. Управление пакетами

В большинстве систем Linux используются инструменты управления пакетами для облегчения установки и обслуживания всего программного обеспечения в вашей системе.

Хотя эти инструменты предоставляют множество мощных функций, легко ознакомиться с преимуществами управления пакетами. Если вы устанавливаете программное обеспечение вручную без инструментов управления пакетами, становится сложно поддерживать вашу систему в актуальном состоянии и управлять зависимостями. По этим причинам мы рекомендуем устанавливать все программное обеспечение через инструменты управления пакетами, если другие средства не являются абсолютно необходимыми.

Ниже рассматривается задача поиска пакетов, установленных на вашей системе.

Кажущаяся легкость установки пакетов может повлечь за собой формирование множества неочевидных зависимостей, и можно легко потерять информацию о том, какое программное обеспечение установлено в вашей системе. Следующие команды предоставляют список установленных пакетов в вашей системе.

Для систем Debian и Ubuntu:

```
dpkg -l
```

В следующем примере представлены первые несколько строк вывода этой команды в производственной системе Debian Lenny.

```

||/ Name                Version                Description
+++-----
=====
ii adduser              3.110                 add and remove users and groups
ii apache2-mpm-itk     2.2.6-02-1+lenny2    multiuser MPM for Apache 2.2
ii apache2-utils       2.2.9-10+lenny4      utility programs for webservers
ii apache2.2-common    2.2.9-10+lenny4      Apache HTTP Server common
files
ii apt                  0.7.20.2+lenny1      Advanced front-end for dpkg
ii apt-utils           0.7.20.2+lenny1      APT utility programs
ii bash                 3.2-4                 The GNU Bourne Again SHell

```

Для систем CentOS и Fedora:

```
yum list installed
```

В следующем примере показано несколько строк вывода этой команды:

```

MAKEDEV.i386           3.23-1.2              installed
SysVinit.i386          2.86-15.el5           installed

```

Системы CentOS и Fedora предоставляют имя пакета (SysVinit), архитектуру, скомпилированную для (i386), и версию сборки, установленной в системе (2.86-15.el5).

Для системы Arch Linux

```
pacman -Q
```

Эта команда предоставляет общий список всех пакетов, установленных в системе. Arch также позволяет фильтровать эти результаты, чтобы отображать только те пакеты, которые были явно установлены (с параметром `-Qe`) или которые были автоматически установлены как зависимости (с опцией `-Qd`). Вышеприведенная команда представляет собой комбинацию вывода двух команд:

```
pacman -Qe
pacman -Qd
```

Ниже приведен пример вывода:

```
perl-www-mechanize 1.60-
```

```
perl-yaml 0.70-1
pkgconfig 0.23-1
procmail 3.22-2
python 2.6.4-1
rsync 3.0.6-1
```

Для систем Gentoo Linux:

```
emerge -evp --deep world
```

Ниже приведен пример этого вывода:

These are the packages that would be merged, in order:

```
Calculating dependencies... done!
[ebuild R ] sys-libs/ncurses-5.6-r2 USE="unicode -debug -doc -gpm -minimal -
nocxx -profile -trace" 0 kB
[ebuild R ] virtual/libintl-0 0 kB
[ebuild R ] sys-libs/zlib-1.2.3-r1 0 kB
```

Поскольку в любой Linux-системе часто установлено большое количество пакетов, результаты вывода могут быть весьма объемными. В связи с этим часто бывает полезно использовать такие инструменты, как `grep`, и `less`, чтобы сделать эти результаты более полезными. Например:

```
dpkg -l | grep "python"
```

Это вернет список всех пакетов со словом «python» в их имени или описании. Точно так же вы можете использовать «less»:

```
dpkg -l | less
```

Это вернет тот же список, что и обычный «`dpkg -l`»; однако результаты будут отображаться в меньшем пейджере, что позволяет легко и легко выполнять поиск и прокрутку.

5.3. Поиск имен пакетов и информации

Иногда имя пакета не является интуитивно понятным и не основывается на имени программного обеспечения. В соответствии с этим большинство инструментов управления пакетами предоставляют возможность поиска в базе данных пакетов. Эти инструменты поиска могут оказаться полезными, если вы ищете конкретную часть программного обеспечения, но не знаете, как это называется.

Для систем Debian и Ubuntu:

```
apt-cache search [package-name]
```

Эта запись будет искать базу данных локального пакета для данного термина и генерировать список с описаниями. Выдержка из вывода для `apt-cache search python` следующая:

```
txt2regex - A Regular Expression "wizard", all written with bash2
builtins
vim-nox - Vi IMproved - enhanced vi editor
vim-python - Vi IMproved - enhanced vi editor (transitional package)
vtk-examples - C++, Tcl and Python example programs/scripts for VTK
zope-plone3 - content management system based on zope and cmf
zorp - An advanced protocol analyzing firewall
groovy - Agile dynamic language for the Java Virtual Machine
python-django - A high-level Python Web framework
python-pygresql-dbg - PostgreSQL module for Python (debug extension)
python-samba - Python bindings that allow access to various aspects of
Samba
```

Обратите внимание, что поиск `apt-cache` запрашивает полные записи для всех пакетов, а не просто названия и описания, отображаемые здесь, следовательно, нужно включить `vim-nox` и `groovy`, которые упоминают `python` в своих описаниях. Чтобы просмотреть полную запись по определенному пакету, нужно записать:

```
apt-cache show [package-name]
```

Эта запись предоставляет информацию о поддерживающем устройстве, зависимостях, размере, домашней странице родительского проекта и описании программного обеспечения.

Для систем CentOS и Fedora:

```
yum search [package-name]
```

Эта запись создает список всех пакетов, доступных в базе данных пакетов, которые соответствуют данному термину. Ниже приведен пример вывода `yum search wget`:

```
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
* addons: centos.secsup.org
* base: centos.secsup.org
* extras: centos.secsup.org
* updates: styx.biochem.wfubmc.edu
===== Matched: wget
=====
  wget.i386 : A utility for retrieving files using the HTTP or FTP protocols.
```

Вы можете использовать инструменты управления пакетами, чтобы узнать больше информации о конкретном пакете. Используйте следующую команду, чтобы получить полную запись из базы данных пакета:

```
yum info [package-name]
```

Этот результат представляет более подробную информацию о пакете, его зависимостях, происхождении и цели.

Для систем Arch Linux:

```
расман -Ss [package-name]
```

Это приведет к поиску базы данных локального пакета. Вот выдержка из результатов поиска «python»:

```
extra/twisted 8.2.0-1
    Asynchronous networking framework written in Python.
community/emacs-python-mode 5.1.0-1
    Python mode for Emacs
```

Термины «extra» и «community» относятся к тому репозиторию, в котором находится программное обеспечение. Чтобы запросить дополнительную информацию о конкретном пакете, выполните команду в следующей форме:

```
расман -Si [package-name]
```

Запуск расман с параметром -Si генерирует запись пакета из базы данных. Эта запись включает зависимости, размер пакета и краткое описание.

Для систем Gentoo Linux:

```
emerge --search [package-name]
emerge --searchdoc [package-name]
```

Первая команда только ищет базу данных для имен пакетов. Вторая команда выполняет поиск по базе данных для имен и описаний пакетов. Эти команды позволят вам искать локальное дерево пакетов (Portage) для определенного имени или термина пакета. Результат любой команды аналогичен выводу ниже.

```
Searching...
[ Results for search key : wget ]
[ Applications found : 4 ]

* app-emacs/emacs-wget
  Latest version available: 0.5.0
  Latest version installed: [ Not Installed ]
  Size of files: 36 kB
  Homepage:      http://pop-club.hp.infoseek.co.jp/emacs/emacs-
wget/
  Description:   Wget interface for Emacs
  License:       GPL-2
```

Вывод, предоставленный командой `emerge --search`, является подробным, в отличие от других инструментов дистрибутива. Команда `emerge -search` позволяет использовать регулярные выражения, если вам нужно сузить результаты.

Поскольку при поиске пакетов часто возникает большое количество результатов, эти команды выводят большое количество текста. В результате часто бывает полезно использовать такие инструменты, как `grep` и `less`, чтобы облегчить прокрутку этих результатов. Например:

```
apt-cache search python | grep "xml"
```

Эта запись вернет подмножество списка пакетов, которые соответствуют поисковому термину «python», и которые упоминают `xml` в их имени или кратком описании. По аналогии:

```
apt-cache search python | less
```

Эта запись вернет тот же список, что и обычный `python` для поиска `apt-cache`, но результаты будут отображаться в меньшем пейджере. Это позволяет выполнять поиск и прокрутку более удобно.

Вы можете использовать `append | grep "[string]"` для любой из этих команд для фильтрации результатов поиска пакетов `or | less`, чтобы отображать результаты в меньшем пейджере, независимо от распределения.

6. Обработка текста

Среди Linux и UNIX-подобных систем почти вся информация о конфигурации системы хранится и обрабатывается в текстовой форме. В следующих разделах приведен список основных команд Linux и инструментов для управления текстовыми файлами.

6.1. Поиск строки в файлах с помощью `grep`

Средство `grep` позволяет вам искать поток текста, такой как файл или вывод команды, для выражения или шаблона регулярного выражения.

Чтобы использовать инструмент `grep`, давайте рассмотрим пример:

```
grep "^Subject:.*HELP.*" /home/username/mbox
```

Эта запись позволяет провести поиск в вашей почте для строк темы (начиная со слова «Subject:»), начиная с любого количества символов, содержащих слово «помощь» в верхнем регистре, и за ним следует любое количество дополнительных символов. Затем результат команды будет выведен в терминале.

Средство `grep` предоставляет дополнительные параметры, которые, если указано, заставляют программу выводить контекст для каждого соответствия (например, с `-C 2` для двух строк контекста). С `-n`, `grep` выводит номер строки совпадения. С `-l` `grep` печатает имя файла для каждого совпадения, что полезно, когда вы выполняете «`grep`» группы файлов или выполняете «`grep`» рекурсивно через файловую систему (используя `-r`). Используйте `grep --help` для получения дополнительных параметров.

Чтобы выполнить `grep` группы файлов, вы можете указать файл с помощью шаблона:

```
grep -i "morris" ~/org/*.txt
```

Эта запись найдет все записи, не содержащие слова «morris» (из-за опции `-i`). Средство `grep` будет искать все файлы в каталоге `~/org/` с расширением `.txt`.

Вы можете использовать `grep` для фильтрации результатов другой команды, отправляющей выходные данные на стандартный выход (`stdout`). Это делается путем «конвейера» вывода одной команды в `grep`. Например:

```
ls /home/username/data | grep "1257"
```

В этом примере мы предполагаем, что каталог `/home/username/data` содержит большое количество файлов с отметкой времени UNIX в именах файлов.

Вышеупомянутая команда будет фильтровать вывод, отображая только те файлы, которые имеют четыре цифры «1257» в именах. В этих случаях `grep` только фильтрует вывод `ls` и не смотрит в содержимое файла.

6.2. Поиск и замена по всей группе файлов

Хотя инструмент `grep` достаточно эффективен для фильтрации текста на основе регулярных выражений, если вам нужно отредактировать файл или иным образом манипулировать текстом, вы можете использовать `sed`. Инструмент `sed` или `Stream EDitor` позволяет вам искать шаблон регулярного выражения и заменять его другой строкой.

`sed` – чрезвычайно мощный инструмент, и мы рекомендуем вам создавать резервные копии файлов и тщательно тестировать команды `sed` перед их запуском. Вот очень простой однострочный `sed`, предназначенный для иллюстрации его синтаксиса:

```
sed -i `s/^good/BAD/` morning-star.txt
```

Это заменяет вхождения слова «good» в начале строки (отмечено символом `^`) строкой «BAD» в файле `morning-star.txt`. Опция `-i` сообщает `sed` выполнить замену «на месте». Команда `sed` может создавать резервные копии файлов, которые она редактирует, если вы укажете суффикс после опции `-i`, как в `-iBAK`. В приведенной выше команде этот параметр сохранит исходный файл как `morning-star.txt.BAK` перед внесением изменений.

Общий формат команды `sed`:

```
's/[regex]/[replacement]/'
```

Чтобы сопоставить буквенные косые черты (`/`), вы должны избегать их с помощью обратной косой черты (`\`). В результате, чтобы соответствовать символу `/`, вы должны использовать `\/` в выражении `sed`. Если вы ищете строку с несколькими слэшами, вы можете заменить слэши, которые имеют другой символ. Например:

```
's|r/e/g/e/x|regex|'
```

Это приведет к удалению косых черт из строки `r / e / g / e / x`, чтобы эта строка представляла собой регулярное выражение после запуска команды `sed` в файле, который содержит строку.

Следующий пример выполняет поиск и заменяет один IP-адрес другим. В этом случае `98.76.54.32` заменяется на `12.34.56.78`:

```
sed -i 's/98\.76\.54\.32/12\.34\.56\.78/'
```

В приведенном выше примере символы периода экранируются как `\.` В регулярных выражениях символ полной остановки (периода) соответствует любому символу, если он не экранирован.

6.3. Редактирование текста

Во многих документах Linode вам может быть предложено отредактировать содержимое файла. Для этого вам нужно использовать текстовый редактор. Большинство дистрибутивов, предоставляемых Linode, имеют реализацию текстового редактора `vi` / `vim` и текстового редактора. Это небольшие, легкие и мощные текстовые редакторы, которые позволяют вам манипулировать текстом файла из среды терминала.

Существуют и другие варианты текстовых редакторов, в частности `emacs` и «`zile`». Не стесняйтесь устанавливать эти программы с помощью диспетчера пакетов вашей операционной системы. Убедитесь, что вы выполняете поиск в базе данных пакетов, чтобы можно было установить версию, скомпилированную без компонентов графического интерфейса (например, X11).

Чтобы открыть файл, введите команду, начинающуюся с имени редактора, который вы хотите запустить, за которым следует имя файла, который вы хотите отредактировать. Вот несколько примеров команд, которые открывают файл `/etc/hosts`:

```
nano /etc/hosts
vi /etc/hosts
emacs /etc/hosts
zile /etc/hosts
```

Когда вы редактируете файл, вы можете сохранить и выйти из редактора, чтобы вернуться в командную строку. Эта процедура зависит от разных редакторов. В `emacs` и `zile` последовательность клавиш одна и та же: нажмите кнопку управления и введите `x` и `s` для сохранения. Эта операция обычно обозначается как «`C-x C-s`», а затем «`C-x C-c`», чтобы закрыть редактор. В `nano` нажмите `Control-O` (обозначено `\ ^ O`) и подтвердите имя файла, чтобы записать файл, и используйте `Control-X` для выхода из программы.

Работа редакторов `vi` и `vim` немного сложнее. После открытия файла в `vi` вы можете ввести режим «вставки», нажав клавишу «`i`»; это позволит редактировать текст обычным способом. Чтобы сохранить файл, вы должны выйти в «обычный» режим, нажав клавишу эвакуации (`Control-` [также отправляет `esc`]-выход), а затем введите: `wq`, чтобы записать файл и выйти из программы.

Выше рассмотрены только самые основные схемы использования этих текстовых редакторов, и есть множество внешних ресурсов, которые предоставят инструкции для более продвинутого использования этого программного обеспечения.

7. Веб-серверы и HTTP-проблемы

7.1. Веб серверы

Веб-серверы работают, прослушивая порт TCP, обычно порт 80 для HTTP и порт 443 для HTTPS. Когда посетитель делает запрос на контент, сервер отвечает, предоставляя запрашиваемый ресурс. Как правило, ресурсы указываются с URL-адресом, который содержит протокол, http или https; двоеточие и две слэши: //; имя хоста или домен, www.example.com или username.example.com; и путь к файлу, /images/avatar.jpg или index.html. Полный URL-адрес будет похож на http://www.example.com/images/avatar.jpg.

Чтобы предоставить эти ресурсы пользователям, в вашем Linode должен быть запущен веб-сервер. Существует множество различных HTTP-серверов и множество конфигураций для поддержки различных рамок веб-разработки. Три наиболее популярных веб-сервера общего пользования - это HTTP-сервер Apache, Lighttpd («Lighty») и nginx («Engine X»). У каждого сервера есть свои сильные и слабые стороны, и ваш выбор во многом зависит от вашего опыта и ваших потребностей.

После того, как вы выбрали веб-сервер, вам нужно решить, какую (если есть) поддержку скриптинга вам необходимо установить. Поддержка сценариев позволяет запускать динамический контент со сценариями на вашем сервере и на стороне сервера на таких языках, как Python, PHP, Ruby и Perl.

7.2. Как выбрать веб-сервер?

В большинстве случаев конечные пользователи не знают, какой веб-сервер вы используете. В результате выбор веб-сервера часто является личным решением, основанным на комфорте администратора и потребностях данного развертывания. Это может стать проблемой для нового системного администратора. В этом разделе приводятся некоторые рекомендации, предоставляя информацию и информацию о наиболее популярных веб-серверах.

HTTP-сервер Apache считается некоторым стандартным веб-сервером *de facto*. Это наиболее широко распространенный веб-сервер с открытым исходным кодом, его интерфейс конфигурации стабилен в течение многих лет, а его модульная архитектура позволяет ему функционировать во многих различных типах развертываний. Apache формирует основу стека LAMP и поддерживает интеграцию динамических серверных приложений в веб-сервер.

Напротив, веб-серверы, такие как Lighttpd и nginx, оптимизированы для эффективного обслуживания статического контента. Если в вашей системе ресурсы сервера ограничены и имеют большую нагрузку, рассмотрите один из

этих серверов. Они функциональны и стабильны с минимальными системными ресурсами. Lighttpd и nginx могут быть сложнее в конфигурировании при интеграции динамических интерпретаторов контента.

Выбор веб-сервера зависит от типа данных, потребности интеграции с программными средами и компетенций персонала.

7.3. Журналы Apache

Когда что-то не так с Apache, может быть трудно определить, какая причина ошибки связана с поведением веб-сервера. Существует ряд распространенных проблем, с помощью которых вы можете начать работу по устранению неполадок. Когда возникают более сложные проблемы, вам может потребоваться просмотреть журналы ошибок Apache.

По умолчанию журналы ошибок находятся в файле `/var/log/apache2/error.log` (в дистрибутивах на базе Debian). Вы можете отслеживать состояние этого журнала с помощью следующей команды:

```
tail -F /var/log/apache2/error.log
```

В настройках виртуального хоста по умолчанию, предложенных в наших установках Apache и LAMP, мы предлагаем добавить настраиваемый параметр журнала:

```
ErrorLog /var/www/html/example.com/logs/error.log CustomLog  
/var/www/html/example.com/logs/access.log combined
```

Здесь `example.com` представляет имя вашего виртуального хоста и местоположение его ресурсов. Эти директивы заставляют Apache создавать два файла журнала, которые содержат информацию о регистрации, относящуюся к этому виртуальному хосту. Это позволяет легко устранять ошибки на конкретных виртуальных хостах. Для отслеживания или закрытия журнала ошибок используем команду

```
tail -F /var/www/html/example.com/logs/error.log
```

Это позволит вам видеть новые сообщения об ошибках по мере их появления. Проблемы могут быть диагностированы с использованием определенных частей сообщения об ошибке из журнала Apache в качестве термина в веб-поиске. Общие ошибки для поиска включают:

- Отсутствующие файлы или ошибки в именах файлов

- Ошибки разрешений

- Ошибки конфигурации

- Выполнение или интерпретация динамического кода.

8. Права пользователя и группы

8.1. Что такое права пользователя и группы?

Операционные системы Linux/Unix позволяют работать в многозадачном режиме способом, подобным другим операционным системам. Однако существенным отличием Linux от других операционных систем является его способность иметь многочисленное количество пользователей на системном уровне. Linux был изначально разработан таким образом, чтобы позволить нескольким пользователям одновременно иметь доступ к системе. Чтобы этот многопользовательский режим работал правильно, должен существовать метод защиты пользователей друг от друга. Здесь и устанавливаются различные права доступа.

8.2. Чтение, запись и выполнение

Разрешения на права доступа – это «права» для работы с файлом или каталогом. Базовыми правами являются чтение (read), запись (write) и выполнение (execute).

- Чтение - доступное для чтения разрешение позволяет просматривать содержимое файла. Разрешение на чтение в каталоге позволяет вам перечислить содержимое каталога.
- Запись - разрешение на запись в файле позволяет вам изменять содержимое этого файла. Для каталога разрешение записи позволяет редактировать содержимое каталога (например, добавлять / удалять файлы).
- Выполнить - для файла исполняемое разрешение позволяет запускать файл и выполнять программу или скрипт. Для каталога разрешение на выполнение позволяет вам перейти на другой каталог и сделать его текущим рабочим каталогом. Обычно у пользователей есть группа по умолчанию, но они могут принадлежать нескольким дополнительным группам.

8.3. Просмотр разрешений файлов

Чтобы просмотреть разрешения на файл или каталог, выполните команду `ls -l <directory / file>`. Не забудьте заменить информацию в `<>` фактическим именем файла или каталога. Ниже приведен пример вывода команды `ls`:

```
-rw-r--r-- 1 root root 1031 Nov 18 09:22 /etc/passwd
```

Первые десять символов показывают разрешения доступа. Первый символ это признак каталога (d) или файла (s или -).

Следующие три тройки символов формируют маску прав. В каждой тройке выводится право на чтение (r), право на запись (w) и право на выполнение (x). Для каталога право на запись определяет возможность создание в нем полкаталогов и

файлов, а право выполнения означает возможность сделать этот каталог текущим. Первая тройка прав относится к пользователю – владельцу файла, вторая – к группе, которая владеет файлом, третья – ко всем остальными пользователями системы.

8.4. Работа с пользователями, группами и справочниками

Чтобы создать нового обычного пользователя, используйте команду `useradd`.

```
useradd <name>
```

Команда `useradd` использует множество переменных, некоторые из которых показаны в таблице ниже:

Опция	Описание	Пример
<code>-d <home_dir></code>	home_dir will be used as the value for the user's login directory	<code>useradd <name> -d /home/<user's home></code>
<code>-e <date></code>	the date when the account will expire	<code>useradd <name>** -e <YYYY-MM-DD></code>
<code>-f <inactive></code>	the number of days before the account expires	<code>useradd <name> -f <0 or -1></code>
<code>-s <shell></code>	sets the default shell type	<code>useradd <name> -s /bin/<shell></code>

Вам нужно будет установить пароль для нового пользователя, используя команду `passwd`. Обратите внимание: для изменения пароля пользователя вам понадобятся права `root`. Синтаксис команд выглядит следующим образом:

```
passwd <username>
```

Пользователь сможет изменить свой пароль в любое время, используя команду `passwd` с синтаксисом. Ниже приведен пример:

```
$ passwd
Changing password for Nivanov.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

Существует еще один способ создания учетных записей пользователей, которые могут быть проще для начинающих администраторов. Однако вам может потребоваться установить новый пакет. Команда установки для Debian / Ubuntu выглядит следующим образом:

```
apt-get install adduser
```

Команда `adduser` автоматически создает домашний каталог и устанавливает группу по умолчанию, оболочку и т. Д. Чтобы создать нового стандартного пользователя с помощью команды `adduser`, синтаксис выглядит следующим образом:

```
adduser <name>
```

После ввода команды вы получите серию подсказок; большая часть этой информации является необязательной. Однако вы должны указать хотя бы имя пользователя (для этого примера имя пользователя - `sjones`) и, конечно же, пароль.

```
root@localhost:~# adduser Pivanova
Adding user `Pivanova' ...
Adding new group `Pivanova' (1001) ...
Adding new user `Pivanova' (1001) with group `Pivanova' ...
Creating home directory `/home/Pivanova' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for Pivanova
Enter the new value, or press ENTER for the default
  Full Name []: Chuck Jones
  Room Number []: 213
  Work Phone []: 856-555-1212
  Home Phone []:
  Other []:
Is the information correct? [Y/n] Y
```

Чтобы удалить учетную запись пользователя, введите следующую команду:

```
userdel <name>
```

Выдача вышеуказанной команды приведет к удалению учетной записи пользователя. Их файлы и домашний каталог не будут удалены.

Чтобы удалить пользователя, домашнюю папку и файлы, используйте следующую команду:

```
userdel -r <name>
```

8.5. Понимание Sudo

Root является суперпользователем и имеет возможность делать в системе все, что угодно. Поэтому для защиты от нанесения системе возможного ущерба sudo используется вместо root. Sudo позволяет пользователям и группам получать доступ к командам, которые они обычно не могут использовать. Sudo позволит пользователю получать административные привилегии без входа в систему с правами root. Образец команды sudo выглядит следующим образом:

```
sudo apt-get install <package>
```

Прежде чем использовать sudo, его необходимо установить, если он не является частью вашего дистрибутива. Команда Debian выглядит следующим образом:

```
apt-get install sudo
```

Чтобы предоставить пользователю возможность sudo, их имя нужно будет добавить в файл sudoers. Этот файл очень важен и не должен редактироваться напрямую с помощью текстового редактора. Если файл sudoers отредактирован неправильно, это может привести к предотвращению доступа к системе.

Поэтому команда visudo должна использоваться для редактирования файла sudoers. В командной строке войдите в систему как пользователь root и введите команду visudo.

Ниже приведена часть файла sudoers, в которой отображаются пользователи с доступом sudo.

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
PIvanova ALL=(ALL:ALL) ALL
kbrown  ALL=(ALL:ALL) ALL
lmartin ALL=(ALL:ALL) ALL
```

После того, как вы предоставили свои привилегии sudo для учетной записи пользователя, сохраните файл sudoers и выйдите из него с правами root. Теперь войдите в систему как пользователь и проверьте привилегии как ваш пользователь с доступом sudo. Когда новому пользователю нужен доступ sudo, теперь вы сможете редактировать файл sudoers с помощью своего собственного входа, используя следующую команду:

```
sudo visudo
```

8.6. Работа с группами

Linux использует группы для организации пользователей. Группы организуют сбор счетов, прежде всего в качестве меры безопасности. Контроль членства в группе осуществляется через файл / etc / group, который показывает список групп и членов. У каждого пользователя есть группа по умолчанию или основ-

ная группа. Когда пользователь входит в систему, членство в группе устанавливается для их основной группы. Это означает, что когда пользователь запускает программу или создает файл, как файл, так и запущенная программа будут связаны с текущим членством в группе. Пользователь может обращаться к другим файлам в других группах, если они также являются членами этой группы и установлены права доступа. Чтобы запускать программы или создавать файл в другой группе, пользователь должен запустить команду `newgrp`, чтобы переключить свою текущую группу. Образец команды `newgrp` выглядит следующим образом:

```
$ newgrp <marketing>
```

Если пользователь, входящий в указанную выше команду, является членом маркетинговой группы в файле `/etc/group`, то текущее членство в группе изменится. Важно отметить, что любые созданные файлы теперь будут связаны с маркетинговой группой, а не с основной группой пользователя. Пользователи также могут изменять свою группу, используя команду `chgrp`. Синтаксис команды `chgrp` следующий:

```
$ chgrp <newgroup>
```

8.7. Создание и удаление каталогов

Чтобы создать каталог, используйте команду:

```
mkdir <directory name>
```

Чтобы создать каталог и установить права доступа одновременно, используйте следующий параметр и синтаксис:

```
mkdir -m a=rwx <directory name>
```

Параметр `-m` является коротким для режима, а `a = rwx` означает, что все пользователи имеют права на чтение, запись и выполнение в каталоге. Чтобы просмотреть полный список всех параметров команды `mkdir`, введите `man mkdir` в командной строке.

Чтобы удалить файл, используйте следующее:

```
rm <file>
```

Чтобы удалить каталог, используйте следующее:

```
rm -r <directory name>
```

Важно отметить, что если вы удалите каталог, все файлы внутри будут также удалены.

8.8. Изменение прав доступа на каталог и файл

Чтобы просмотреть права доступа к файлам и права собственности на файлы и каталоги, используйте команду `ls -al`. Параметр `a` - показывает скрытые файлы или все файлы, а параметр `l` - для длинного списка. Вывод будет похож на следующий:

```
drwxr-xr-x 2 user user 4096 Jan 19 11:41 documents
-rw-r--r-- 1 user user 675 Jan 17 13:45 .profile
drwxr-xr-x 4 user user 4096 Jan 17 15:45 public
```

Первый столбец с десятью буквами и тире показывает разрешения файла или каталога. Второй столбец (с единственным номером) указывает количество файлов или каталогов, содержащихся в каталоге. Следующий столбец указывает владельца, за которым следует имя группы, размер, дата и время последнего доступа и, наконец, имя файла. Например, используя первую строку из вышеприведенного вывода, приведена следующая информация:

```
drwxr-xr-x are the permissions
2 is the number of files or directories
user is the owner
user is the group
4096 is the size
Jan 9 10:11 is the date/time of last access
documents is the directory
```

9. Настройка средств сети в Linux

9.1. Устройство сетевого стека в Linux

Устройство обобщенного сетевого стека традиционно описывается моделью взаимодействия открытых систем (OSI — Open Systems Interconnection). В ней задана терминология и общие принципы сетевой коммуникации по многоуровневому сетевому стеку. Самый распространенный сетевой стек TCP/IP, хоть и унаследовал от OSI термины и принципы, но использует 4-х уровневый стек [Олифер, Олифер, 2017].

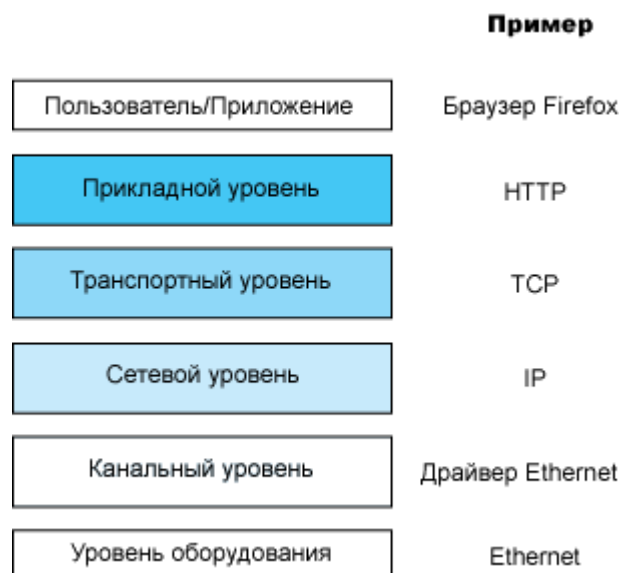


Рис.1 Обобщенная архитектура сетевого стека

В стеке TCP/IP верхний уровень — «Прикладной». На нем работают прикладные протоколы, предоставляющие или программный или командный интерфейс пользователю или приложению. Эти протоколы выполняют пользовательские задачи — передачу веб-страниц, удаленного терминала, передачу электронной почты и др. Как правило, эти протоколы оперируют символьным потоком.

Транспортный уровень обеспечивает связанность приложений через сетевой стек. Он может обеспечивать достоверность передачи (через протоколы TCP или QUIC), или не обеспечивать этого (протокол UDP). В качестве адресной информации используется двухбайтовый номер порта (как правило фиксированный и заранее известный для сервера и динамический для клиента).

Сетевой уровень обеспечивает взаимодействие между узлами в составной сети. Он осуществляет доставку пакета данных, поиск кратчайшего маршрута, поиск роутеров и ближайших соседей и т.п. задачи. К этому уровню относятся протоколы IPv4, IPv6, ARP, ICMPv4, ICMPv6, RIP, BGP и др. В качестве адресной информации используется IP адрес узла из 32 бит для IPv4 и 128 бит для IPv6.

Нижний уровень в стеке TCP/IP описывает интерфейсы инкапсуляции в кадры канального уровня, принципы взаимодействия протоколов сетевого уровня и типичных канальных протоколов, таких как Ethernet.

Для операционной системы Linux основным сетевым стеком является как раз TCP/IP. Архитектура реализации стека приведена на рис.2

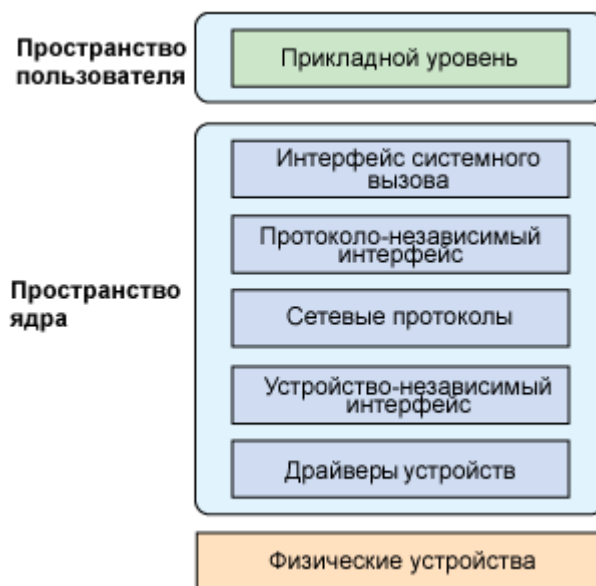


Рис.2. Высокоуровневая архитектура сетевого стека Linux

Основные компоненты сетевого стека функционируют в пространстве ядра. Верхний уровень стека – слой функционирования приложений, клиентов или серверов. Для доступа к сетевому стеку эти процессы используют соответствующие системные вызовы операционной системы.

В самом низу расположены драйвера устройств (устройство IEEE 802.1 или IEEE 802.11 или com-порта). Взаимодействие с драйверами устройств в Linux осуществляется через специальные файлы в каталоге /dev. Например ethXX или wlanXX, где XX – номер сетевого устройства (их может быть несколько).

Подробнее рассмотрим пространство ядра сетевого стека, а точнее сетевую подсистему, операционной системы Linux. Описание сетевого стека приводится по [Джонс, 2007].

Интерфейс системного вызова может быть реализован как нормальные файловые операции ввода – вывода через файл сокета. По сути такие операции порождают системный вызов к ядру

Ниже работает протоколо-независимый (protocol agnostic) уровень. Уровень сокетов является протоколо-независимым (protocol agnostic) интерфейсом, который предоставляет набор стандартных функций для поддержки ряда различных протоколов. Этот уровень не только поддерживает обычные TCP- и UDP-протоколы, но также и IP, raw Ethernet и другие транспортные протоколы, такие как Протокол управления передачей потоков данных (SCTP — Stream Control Transmission Protocol).

Взаимодействие через сетевой стек происходит посредством сокета. Структура сокета в Linux — `struct sock`, определенная в `linux/include/net/sock.h`. Эта большая структура содержит все необходимые состояния отдельного сокета, включая определенный протокол, используемый сокетом, и операции, которые можно над ним совершать [Джонс, 2007].

Слой сетевых протоколов содержит программные модули всех протоколов сетевого и транспортного уровня (IPv4, IPv6, TCP, UDP и др.) Эти программные модули активируются функцией `inet_init`.

Под уровнем протоколов располагается другой независимый уровень интерфейса, который связывает протоколы с различными драйверами физических устройств с разными возможностями. Этот уровень предоставляет стандартный набор функций, которые используются низко-уровневыми сетевыми устройствами, чтобы иметь возможность взаимодействовать с высоко-уровневым стеком протокола [Джонс, 2007].

9.2. Сетевые настройки ядра Linux

Настройки модулей сетевого стека из пространства ядра возможна через конфи конфигурационные файлы иклучи специальных утилит. Однако не все параметры можно установить подобным образом. В таком случае применяется прямое конфигурирование модулей ядра через файловую систему `/proc` напрямую или через специальные утилиты (`make confi`, `sysctl` и др.). Параметров более сотни, рассмотрим некоторые из них, пригодные для сценария организации сетевого шлюза на базе Linux (RHEL, CENTOS, FEDORA).

Для шлюза необходимо увеличить значения размеров буферов для приема и отправки данных через сокеты (параметры ядра `net.core.rmem_max`, `net.core.wmem_default`, `net.core.wmem_max`).

Одной из существенных опасностей для интернет шлюза является атака через «наводнение» пакетами с флагом `syn`. Включим параметр ядра, защищающий от них:

```
net.ipv4.tcp_syncookies = 1
```

Следует запретить ICMP редиректы и игнорировать широковещательные ICMP запросы:

```
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.icmp_echo_ignore_broadcasts=1
```

Для шлюза на котором работает NAT критически важен диапазон портов для трансляции. По умолчанию он достаточно мал. Увеличим его:

```
net.ipv4.ip_local_port_range = 16384 61032
```

Уменьшим время keep alive соединений:

```
net.ipv4.tcp_keepalive_time = 1900
```

Можно так же указать время, в течении которого следует ожидать приема сегмента с флагом FIN до закрытия сокета (в секундах):

```
net.ipv4.tcp_fin_timeout = 30
```

Наконец, для того, чтобы разрешить пересылку сетевых пакетов между сетевыми интерфейсами в системе установим параметр `ip_forward`:

```
net.ipv4.ip_forward = 1
```

9.3. Настройка параметров сетевого интерфейса

После подключения драйверов устройств нужно задать параметры протокола IP для интерфейсов. Настройка интерфейса заключается в присвоении IP-адреса. Так же в системе необходимо указывать адрес шлюза по умолчанию и параметры DNS.

Эти параметры можно установить через конфигурационные файлы, специфичные для системы и (или) с помощью специальных утилит.

Настройка может проводится вручную (подробно этот процесс рассмотрен далее), или автоматически с помощью протокола DHCP (утилиты `dhclient`).

Для назначения IP и маски используют программу `ifconfig`.

Эта команда позволяет выяснить какие интерфейсы зарегистрированы в системе:

```
ifconfig -a
```

В выводе этой команды должен присутствовать минимально один интерфейс с именем `lo` -так называемый "кольцевой интерфейс" или "обратная петля" - `Local Loopback`.

Он должен функционировать в любом случае, даже если на компьютере нет реальных сетевых интерфейсов, так как осуществляет взаимодействие приложений через сетевой стек в пределах хоста.

Обычные интерфейсы обозначаются как `ethN`, где `N` – номер интерфейса, начиная с 0. Заметим, что имена интерфейсов могут отличаться в разных версиях Linux.

Для того чтобы присвоить сетевому интерфейсу `eth0` адрес и маску (например IP – `172.21.21.10` и маску `255.255.0.0` следует выполнить команду:

```
ifconfig eth1 172.21.21.10 netmask 255.255.0.0 up
```

Если не указывать маску подсети, то по умолчанию устанавливается маска подсети `255.0.0.0`.

`ifconfig` позволяет менять и параметры канального уровня. Например MAC адрес:

```
ifconfig eth0 hw ether AC:AE:55:66:77:88
```

или даже адрес прерывания сетевой карты и режим работы карты:

```
ifconfig eth0 irq 5 media 10baseT.
```

9.4. Настройка маршрутов по умолчанию

Еще одним важным параметром сетевого стека является маршрут по умолчанию. Это параметр относится к механизму маршрутизации, реализуемым демоном `rotuted`. Правила маршрутизации определяют последовательность передачи пакетов IP. Эти правила маршрутизации хранятся в одной из таблиц ядра. Формировать таблицы маршрутизации можно статически и динамически. Статический маршрут - это маршрут, который задается вручную командой `route`. Динамическая маршрутизация подразумевает автоматическое формирование правил демоном `routed`. Маршрут по умолчанию определяет куда будут пересылаться пакеты для которых не задан маршрут явно. Для конечного хоста в общем случае это обязательный параметр.

Добавить маршрут с помощью команды `route` можно так:

```
route add default gw 172.21.0.1 eth0 ,
```

а посмотреть таблицу маршрутизации командой `netstat -nr`.

Отметим, что существующий маршрут по умолчанию не замещается автоматически и должен быть удален явно:

```
route del default gw eth0.
```

Использование команды `ip`

Утилита `ip` объединяет в себе возможности традиционных утилит `ifconfig`, `arp`, `route` в Linux.

Утилита `ip` поставляется во всех дистрибутивах по умолчанию и заменит старые утилиты со временем.

Рассмотрим синтаксис команды:

`ip [опции] объект команда [параметры]`

опции — это глобальные настройки, которые сказываются на работе всей утилиты

объект — объект или устройство, с которым будем работать или о котором будем узнавать информацию.

команды — какое-либо действие с объектом.

параметры — само собой, командам иногда нужно передавать параметры, они передаются в этом пункте.

Поддерживаются такие опции для настройки сети в Linux:

`-v` — только вывод информации об утилите и ее версии

`-s` — включает вывод статистической информации

`-f` — позволяет указать протокол, с которым нужно работать, если протокол не указан, то берется на основе параметров команды. Опция `f` должна принимать одно из значений: `bridge`, `dnet`, `inet`, `inet6`, `ipx` или `link`. По умолчанию используется `inet`, `link` — означает отсутствие протокола.

`-o` — выводить каждую запись с новой строки

`-r` — выполнять резолвинг и выводить символические имена хостов

Все остальные опции — ярлыки опции `-f`:

`-4` — ярлык для `-f inet`

`-6` — `-f inet6`

`-B` — `-f bridge`

`-0` — `-f link`

К основным объектам относятся:

`address` — сетевой адрес на устройстве

`link` — физическое сетевое устройство

`monitor` — мониторинг состояния устройств

`neigh` — ARP

`route` — управление маршрутизацией

`rule` — правила маршрутизации

tunnel — настройка туннелирования

Набор команд следующий: add, change, del или delete, flush, get, list или show, monitor, replace, restore, save, set, и update. Если команда не задана, по умолчанию используется show (показать).

Параметры зависят от объекта и указанной команды. К основным параметрам относятся:

dev имя_устройства — сетевое устройство

up — включить

down — выключить

llarr — MAC — адрес

initcwnd — размер окна перегрузки TCP при инициализации

window — размер окна TCP

cwnd — размер окна перегрузки TCP

type — тип

via — подключиться к роутеру

default — маршрут по умолчанию.

Приведем несколько примеров использования команды ip.

ip link show eth1 — отображает информацию о состоянии интерфейса eth1

ip link list up — выводит информацию о статусе всех включенных интерфейсов

ip link set eth1 up — включить eth1

ip link set eth1 down - выключить eth1

ip addr add 172.21.0.3/24 dev eth0 — установить ip адрес для интерфейса eth0

ip addr del 172.21.0.3/24 dev eth0 — удалить ip адрес интерфейса eth0

ip add flush dev eth0 — удалить все ip адреса интерфейса eth0

ip route get 192.168.10.0/24 — отобразить маршрут к этой сети.

9.5. Настройка DNS

Для клиента необходимо задать параметр IP-адрес сервера DNS который будет разрешать доменные имена в IP адреса по запросу клиента.

По умолчанию эти параметры задаются в файле /etc/resolv.conf.

В этом файле конфигурируются службы отображения сетевых адресов в имена хостов. В частности можно указать до трех разных DNS серверов.

Строки конфигурации будут начинаться с строки «nameserver», за которым следует IP-адрес сервера DNS. Можно указать и суффикс домена по умолчанию.

9.6. Настройка через конфигурационные файлы

Использование описанных утилит позволяет сразу изменить параметры сетевого стека «на ходу», но эти параметры нигде не сохраняются и работают до перезагрузки. Для того чтобы сделать их постоянными, нужно добавить вызов утилит в скрипты автозапуска системы или использовать конфигурационные

файлы сетевых интерфейсов, которые используются сетевой службой для их конфигурирования при запуске.

В семействе Linux RedHat эти файлы называются ifcfg-ethX (где X номер интерфейса) и располагаются в каталоге /etc/sysconfig/network-scripts/.

Рассмотрим основной синтаксис файлов ifcfg-eth. В текстовый файл с этим именем заносятся параметры в виде КЛЮЧ=ЗНАЧЕНИЕ. Порядок параметров не важен. Приведем основные ключи, сразу с примерами заполнения.

Имя интерфейса:

DEVICE=eth0

Протокол загрузки:

BOOTPROTO=none

Возможные значения none (конфигурация статическая) или bootp или dhcp (указывается протокол автоматического конфигурирования)

Включать при загрузке:

ONBOOT=yes

Задаем адрес и маску:

NETMASK=255.255.255.0

IPADDR=172.21.0.27

Задаем маршрут по умолчанию:

GATEWAY=172.21.0.1

Задаем параметр DNS

DNS1=172.21.0.1

DNS2=8.8.8.8.

.

Литература

- [Батура, 2012] Батура Т.В. Методы анализа компьютерных социальных сетей // Вестник НГУ. Серия: Информационные технологии. 2012. Том 10. Вып. 4. С. 13–28.
- [Берновский, 2012] Берновский М.М., Кузюрин Н.Н. Случайные графы, модели и генераторы безмасштабных графов // Труды Института системного программирования РАН. 2012. Том 22. С. 419–432.
- [БСЭ] Большая советская энциклопедия. Электронный ресурс. – Режим доступа: <http://dic.academic.ru/dic.nsf/bse/143029>
- [Губанов, 2009] Губанов Д.А., Новиков Д.А., Чхартишвили А.Г. Модели информационного влияния и информационного управления в социальных сетях // Проблемы управления. 2009. № 5. С. 28–35.
- [Губанов, 2010] Губанов Д.А., Новиков Д.А., Чхартишвили А.Г. Социальные сети: модели информационного влияния, управления и противоборства. – М.: Изд. физ.-мат. литературы, 2010. – 228 с.
- [Давыденко, 2005] Давыденко В.А., Ромашкина Г.Ф., Чуканов С.Н. Моделирование социальных сетей // Вестник Тюменского государственного университета. № 1, 2005. С. 68–79.
- [Иванов, 2001] Иванов В.Н. Основы социального управления. Учебное пособие / А. Г. Гладышев, В. Н. Иванов, В. И. Патрушев и др. Под ред. В. Н. Иванова. — М.: Высш. шк., 2001. — 271 с.
- [Лисьев, 2011] Лисьев Г.А., Попова И.В. Технологии поддержки принятия решений. — 2-е издание, стереотипное. — М.: Издательство «ФЛИНТА», 2011. — 133 с. — ISBN 978-5-9765-1300-6.
- [Новиков, Петраков, 1999] Новиков Д.А., Петраков С.Н. Курс теории активных систем М.: СИНТЕГ, 1999. – 104 с.
- [Общая социология, 2009] Общая социология: Учебное пособие / Под ред. проф. А.Г. Эфендиева. – М.: ИНФРА-М, 2009. – 654 с.
- [Рыбина, 2010] Рыбина Г.В. Основы построения интеллектуальных систем. — М.: Финансы и статистика; ИНФРА-М, 2010. — 432 с. — ISBN 978-5-279-03412-3.
- [Социология, 2009] Социология: Учебник для вузов / Под ред. проф. В.Н. Лавриненко. – М.: ЮНИТИ-ДАНА, 2009. – 448 с.
- [Charu, 2012] Charu C. Aggarwal. Social Network Data Analytics. 2011. 520 p.
- [Chi, 2014] Nai-Wen Chi, Ken-Yu Lin, Shang-Hsien Hsieh Using ontology-based text classification to assist Job Hazard Analysis // Advanced Engineering Informatics (2014)
- [Domingos, 2002] Domingos P., Richardson M. Mining the Network Value of Customers / Proceedings of the Seventh International Conference on Knowledge Discovery and Data Mining. 2002. P. 57-66.

[Олифер, Олифер, 2017] Компьютерные сети. Принципы, технологии, протоколы. Учебник / Олифер Виктор Григорьевич, Олифер Наталия Алексеевна. – Санкт-Петербург.: Питер, 2017. – 992 с. - ISBN 978-5-496-01967-5.

[Джонс, 2007] Анатомия сетевого стека в Linux. Электронная публикация. 2007. <https://www.ibm.com/developerworks/ru/library/l-linux-networking-stack/index.html>



Миссия университета – открывать возможности для гармоничного развития конкурентоспособной личности и вдохновлять на решение глобальных задач.

КАФЕДРА ИНТЕЛЛЕКТУАЛЬНЫХ ТЕХНОЛОГИЙ В ГУМАНИТАРНОЙ СФЕРЕ

Кафедра интеллектуальных технологий в гуманитарной сфере была организована в 1998 году и при образовании получила название «кафедра технологий профессионального обучения». Тогда же кафедру возглавил профессор Потеев Михаил Иванович, и с 2002 года кафедра стала выпускающей. В 2012 году кафедра была переименована в соответствие с основным направлением деятельности. В настоящее время кафедру возглавляет кандидат технических наук, доцент Наталия Николаевна Горлушкина.

Центральной идеей образовательных программ, реализуемых кафедрой, является участие студентов в выполнении работ, связанных с возможными направлениями будущей деятельности, и с задачами, решаемыми университетом. Научные исследования, проводимые на кафедре, связаны с интеллектуальным анализом данных, математическим моделированием и проектированием информационных систем. В этих областях много интересных, сложных и нерешенных задач. На старших курсах студенты имеют возможность выбрать то направление в рамках профиля, которое им наиболее интересно.

Существующие международные программы

Сотрудниками кафедры интеллектуальных технологий в гуманитарной сфере ведутся переговоры о внедрении программ двойных/совместных дипломов и реализации международных программ академического студенческого обмена с вузами стран Шанхайской организации сотрудничества, а также Чехии, Англии и Финляндии.

Компании, в которых осуществляется производственная и преддипломная практика, а также компании, трудоустраивающие выпускников

Доктор Web, Государственный Русский музей, ООО «ИНТЕРФОРУМ», Ростелеком, Интерзет, ООО «Интермедиа», «ВКонтакте» и др.

http://www.ifmo.ru/ru/viewdepartment/13/kafedra_intellektualnyh_tehnologiy_v_gumanitarной_sfere.htm#ixzz3byzWK4D7

Ватьян Александра Сергеевна, Береснев Артем Дмитриевич
АДМИНИСТРИРОВАНИЕ
В ИНФОРМАЦИОННЫХ СИСТЕМАХ
Учебное пособие

В авторской редакции

Редакционно-издательский отдел Университета ИТМО

Зав. РИО

Н.Ф. Гусарова

Подписано к печати ____

Заказ № ____

Тираж ____

Отпечатано на ризографе