

Руководство по Asterisk Издание 2

Марк Спенсер
Мэк Эллисон
Кристофер Родз

Коллектив авторов по разработке документации
по Asterisk

последняя редакция: 30.03.03

Перевод Copyright © 2006 Сафонова Т.П.

Руководство по Asterisk

Руководство по Asterisk
Издание 2

О книге

Авторы:

Марк Спенсер

Мэк Эллисон

Кристофер Родз

Коллектив авторов по разработке документации по Asterisk

Выражаем особую благодарность всем пользователям, помощникам и разработчикам, которые помогли Asterisk стать реальностью.

Copyright © 2003 Digium, Inc. *Этот документ запрещается тиражировать, копировать или распространять в любом виде, электронном или физическом, без заранее полученного письменного разрешения от Digium.*

Последнее издание этого документа можно бесплатно скачать с <http://www.digium.com>.

Эта книга была создана с использованием OpenOffice, о котором можно получить информацию на сайте <http://www.openoffice.org>.

Перевод Copyright © 2006 Сафонова Т.П.

Содержание

1. Глава 1: Введение.....	5
1.1 Что такое Asterisk?.....	5
1.2 Приобретение Asterisk.....	6
1.3 Лицензирование.....	6
1.4 Поддерживаемые технологии.....	7
1.4.1 Интерфейсы Zaptel Pseudo TDM.....	7
1.4.2 Интерфейсы Non-Zaptel hardware.....	8
1.4.3 Пакет голосовых протоколов.....	8
1.5 Вклад в создание Asterisk.....	8
1.5.1 Вклад в развитие кодов.....	9
1.5.2 Вклад в создание документации.....	9
1.5.3 Канал IRC Asterisk и почтовый список.....	10
1.5.4 Поддержка спонсоров Asterisk.....	10
1.5.5 Предложения для разработчиков ядра.....	10
2. Глава 2: Архитектура Asterisk.....	11
2.1 Обзор по архитектуре Asterisk.....	11
2.2 Подробное описание архитектуры Asterisk.....	11
2.3 Образцы сетей.....	12
2.3.1 The Mythical 1x1 PBX.....	12
2.3.2 An 8x16 Small Office PBX.....	13
2.3.3 SME с удаленными офисами.....	14
2.3.4 Высокий уровень плотности IVR и конференций.....	14
2.4 Устройство файловой системы.....	15
2.5 Наименование каналов.....	17
2.5.1 Каналы Zap: Zaptel TDM.....	18
2.5.2 Начало работы каналов протокола.....	19
2.5.3 Каналы внутреннего обмена Asterisk.....	19
3. Глава 3: Запуск Asterisk.....	21
3.1 Аргументы командной строки Asterisk.....	21
3.2 Интерфейс командной строки Asterisk.....	23
4. Глава 4: План набора номера в Asterisk.....	25
4.1 Введение в расширенные контексты.....	25

4.1.1	Использование расширенных контекстов.....	25
4.1.2	Базовый расширенный контекст.....	26
4.1.3	Образец голосового меню.....	26
4.1.4	Образец подстановки расширения.....	27
4.1.5	контекстное присоединение.....	28
4.2	Полный набор контекстов.....	30
4.3	Определение расширений.....	30
4.3.1	Пример базового расширения.....	30
4.3.2	Набор номера по телефону.....	31
4.3.3	Маршруты ID звонящего.....	31
4.3.4	Последовательный набор номеров.....	32
4.3.5	Базовое голосовое меню.....	33
4.3.6	Использование переменных величин.....	33
4.3.7	Включенные контексты.....	34
4.3.8	Режимы работы День/ночь.....	35
4.3.9	Несвязанный набор номера.....	36
4.3.10	Сбой на главной линии и LCR.....	37
4.3.11	Использование Macros.....	38
5. Глава 5:	Конфигурация файлов.....	40
5.1	Введение в конфигурацию файлов.....	40
5.2	Грамматика конфигурации файлов.....	40
5.2.1	Простые группы.....	41
5.2.2	Унаследованные опции объекта (например, zapata.conf).....	42
5.2.3	Полное единство объекта (iax.conf).....	43
5.3	Интерфейсы каналов.....	43
5.3.1	zapata.conf.....	43
5.3.2	sip.conf.....	56
5.3.3	iax.conf.....	60
5.4	Конфигурации приложений.....	68
5.4.1	voicemail.conf	68

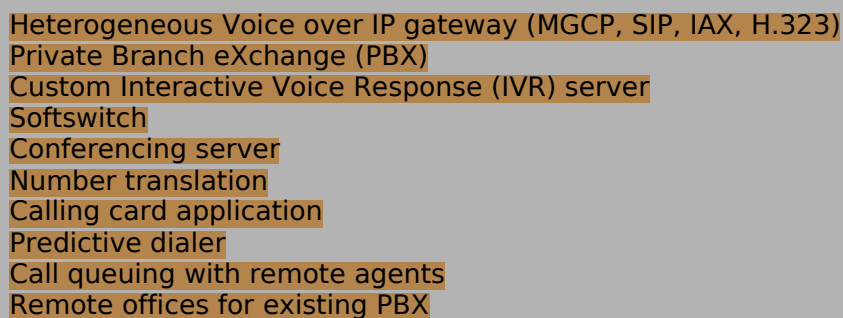
Глава 1: Введение

1.1 Что такое Asterisk?

Официально считается, что Asterisk- это гибрид OpenSource TDM, голосового пакета PBX и платформы IVR с функциональностью ACD. Неофициально Asterisk можно назвать, практически, самым сильным, гибким и распространенным интегрированным телекоммуникационным программным продуктом. Его имя произошло от символа asterisk, *, который в UNIX (включая LINUX) и средах DOS означает любую карту, подходящую к любому файловому имени. Подобно Asterisk PBX разработан под интерфейс любого телефонного «железа» или программного обеспечения и совместим с любым телефонным приложением без состыковки.

Обычные телефонные продукты разрабатываются под специальные технические нужды в сети. Многие приложения по телефонии имеют дело с большим количеством технологий Asterisk, воспользовавшись всеми существующими плюсами, создает единую среду, которую можно настраивать под любое приложение или сборник приложений.

Asterisk, помимо всего прочего, может быть использован в любом из этих приложений



- Heterogeneous Voice over IP gateway (MGCP, SIP, IAX, H.323)
- Private Branch eXchange (PBX)
- Custom Interactive Voice Response (IVR) server
- Softswitch
- Conferencing server
- Number translation
- Calling card application
- Predictive dialer
- Call queuing with remote agents
- Remote offices for existing PBX

Также он может выполнять все эти роли одновременно и без состыковок с интерфейсами.

1.2 Приобретение Asterisk

Вышедшую версию Asterisk можно бесплатно скачать с <ftp://ftp.asterisk.org> через анонимный FTP. При получении доступа к Asterisk, в большинстве случаев, предпочтение отдается анонимному хранилищу, расположенному на cvs.digium.com с CVSROOT :pserver:anoncvs@cvs.digium.com. Более подробно этот вопрос рассматривается в разделе «Скачивание и установка».

1.3 Лицензирование

Asterisk распространяется на условиях GNU General Public License (GPL)- общедоступная Лицензия. Эта лицензия даст вам право свободно распространять Asterisk в качестве источника и в бинарной форме, с модификациями или без них. В любом случае Asterisk распространяется с исходным кодом (включая любые изменения, сделанные вами) и без дальнейших ограничений по использованию или распространения кода. За более подробной информацией обращайтесь к тексту GNU General Public License (дается в приложении).

Общедоступная лицензия (GPL) не распространяется на «железо» и программное обеспечение, с которыми связан Asterisk. Например, если вы используете SIP soft phone в качестве клиента Asterisk, то это совсем не означает, что данная программа поставляется на условиях GPL. Дополнительно к Asterisk даются AGI приложения, которые с ним неразрывно связаны.

По приложениям без условий GNU GPL (например, из-за авторских прав на интеллектуальную собственность) обращайтесь к Digium, которая является единственной организацией, занимающейся лицензированием Asterisk. Более подробно об этом вам расскажут на сайте sales@digium.com.

1.4 Поддерживаемые технологии

Asterisk разработан для того, чтобы новые интерфейсы и технологии легко взаимодействовали. Его основная цель- поддерживать все возможные виды телефонных технологий. Последние данные о «железе» и список совместимости протоколов можно найти на <http://www.digium.com> или <http://www.asterisk.org> . Обычно интерфейсы делятся на три категории: Zaptel hardware, non-Zaptel hardware и голосовой пакет.

1.4.1 Интерфейсы Zaptel Pseudo TDM.

Данные интерфейсы обеспечивают интеграцию с традиционными цифровыми и аналоговыми телефонными интерфейсами (включая соединение с общественной телефонной сетью). Zaptel-совместимые интерфейсы поддерживают псевдо-TDM подключения между собой, что позволяет остаться негласным участником, чего практически невозможно сделать при настоящих TDM звонках, конференциях и т.д. Zaptel интерфейсы доступны Digium (<http://www.digium.com>) для разных сетевых интерфейсов, в том числе PSTN, POTS, T1, E1, PRI, PRA, E&M, Wink, а также Feature Group D. Доступное «железо» на момент написания этого руководства:

T100P - Single span T1 or PRI connection (mixed data/voice permitted)
E100P . Single span E1 or PRA connection (mixed data/voice permitted)
T400P . Quad span T1 or PRI connection (mixed data/voice permitted)
E400P . Quad span E1 or PRA connection (mixed data/voice permitted)
X100P . Single analog PSTN connection
S100U . Single analog POTS connection (USB)
S400P . Single to Quad analog POTS connection (PCI)

Помните, что по техническим причинам, у вас должен быть установлен, по меньшей мере, один интерфейс с Zaptel (любого вида) в вашей системе Asterisk, если вы хотите использовать конференцию.

1.4.2 Не-Zaptel интерфейсы

Эти интерфейсы обеспечивают связь с традиционными и доставшимися в наследство телефонными услугами, но не поддерживают подключения псевдо-TDM. Сюда входят

ISDN4Linux . Basic Rate ISDN interface for Linux
OSS/Alsa . Sound card interfaces
Linux Telephony Interface (LTI) . Quicknet Internet
Phonejack/Linejack
Dialogic hardware¹ . Full-duplex Intel/Dialogic hardware

1.4.3 Пакет голосовых протоколов

Это стандартные протоколы для коммуникации помимо сетевых пакетов (IP и Frame Relay). Они являются единственными интерфейсами, которые не требуют никакого специализированного «железа».

Session Initiation Protocol (SIP)
Inter-Asterisk eXchange (IAX) versions 1 and 2
Media Gateway Control Protocol (MGCP)
ITU H.323₂
Voice over Frame Relay (VOFR)

1.5 Вклад в создание и усовершенствование Asterisk

Не смотря на то, что, по большей части, Asterisk является результатом трудов и усилий Digium -главного корпоративного спонсора многих проектов Open Source (открытый источник)- большой вклад в развитие Asterisk внесли и многочисленные отдельные граждане. Внести свой вклад в Asterisk можно следующими способами:

-
1. Диалоговое устройство не поддерживается стандартом Asterisk, но доступно за деньги для клиентов с Intel/Dialogic устройствами.
 2. Во время написания этого документа поддержка H.323 доступна бесплатно в виде дополнения к Asterisk из 3 частей.

1.5.1 Вклад в развитие кодов

Если вы являетесь разработчиком, вы можете внести свой вклад в базу кодов Asterisk посредством баг фиксов, усовершенствование модификацией, новых приложений и драйверов каналов. Как правило, любители делают патчи на текущем CVS и ставят их в формате «unified diff», который можно создать, выполнив:

```
# cvs diff -u > mypatch.diff
```

Полученный файл (как было показано в предыдущем примере mypatch.diff) затем отправляют по e-mail автору (markster@digium.com). До того как патч появится в стандартном Asterisk, автор патча передает все свои права на использование патча Digium (правообладателю Asterisk). В конце этого документа предлагается два варианта передачи прав на использование патча. Любой вариант, по вашему усмотрению, может быть использован. Заполнив и подписав документ, отошлите его по факсу (а предпочтительнее, по почте) Digium. За дополнительной информацией обращайтесь на сайт <http://www.digium.com>.

1.5.2 Вклад в создание документации.

Если вы не являетесь разработчиком, вы можете внести свой вклад в Asterisk посредством создания документа, в котором вы опишите свой опыт привыкания к Asterisk. В свою очередь, такой документ поможет другому человеку в освоении этого программного обеспечения. Документ может содержать главы «Руководство по эксплуатации Asterisk» или просто «Руководство», примечания по использованию Asterisk в специфичных средах или при определенном способе использования (их называют «App Notes»), или задокументирование программирование API по Asterisk.

1.5.3 Канал IRC Asterisk и почтовый адрес.

Еще один эффективный способ внести свой вклад в развитие Asterisk – помощь в обсуждении по программированию и обеспечение технической поддержки других пользователей Asterisk по каналу IRC Asterisk или по почтовому списку Asterisk. Канал IRC Asterisk называется «#asterisk» (что неудивительно). Он находится на irc.freenode.net . За подробной информацией по почтовому списку Asterisk обращайтесь на сайт <http://lists.digium.com> .

1.5.4 Поддержка спонсоров Asterisk.

Когда вы покупаете у Digium- главного корпоративного спонсора Asterisk- устройство, поддержку или развитие, вы напрямую содействуете продвижению Asterisk.

1.5.5 Список предложений для разработчиков ядра.

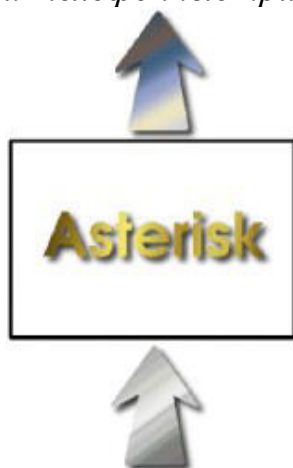
У нескольких независимых Asterisk разработчиков ядра есть «список предложений» в таких компаниях, как Amazon, ThinkGeek и др. Ваши предложения помогут им продолжить работу над развитием Asterisk.

Глава 2: Архитектура Asterisk

2.1 Обзор по архитектуре Asterisk

В своей основе архитектура Asterisk крайне проста, но, в то же время, она отличается от архитектуры большинства телефонных продуктов. В целом, Asterisk является промежуточным звеном между телефонными технологиями внизу и телефонными приложениями вверху, создавая тем самым устойчивую среду для использования смешанной телефонной среды. В телефонные технологии входят такие услуги VoIP, как SIP, H.323, IAX, MGCP (как входные так и телефонные), а также более традиционные TDM технологии- T1, ISDN PRI, аналоговые POTS и PSTN услуги, Basic Rate (базовая норма) ISDN (BRI) и другие.

Интернет и телефонные приложения



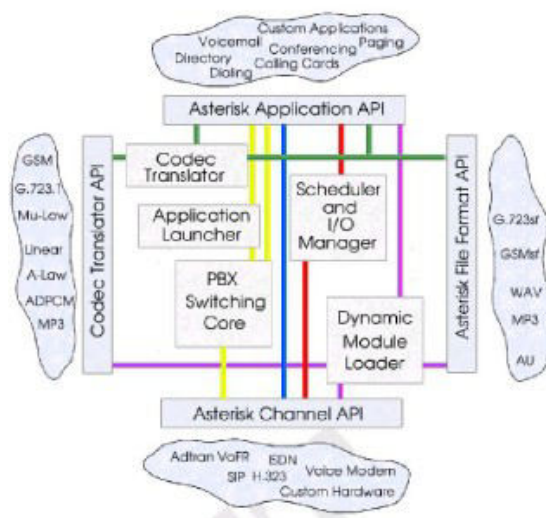
Интернет и телефонные технологии

Телефонные приложения включают в себя такие вещи, как call bridging (соединения звонков), конференции, голосовую почту, автоответчик, обычный IVR скриптинг, парковку звонков, интерком и др.

2.2 Подробное описание архитектуры Asterisk.

Ядро Asterisk содержит несколько двигателей, каждый из которых играет свою роль в работе программного обеспечения. Когда Asterisk впервые начинает работать Dynamic Module Loader загружает и инициализирует все драйверы, которые обеспечивают работу драйверов каналов, файловых форматов, запись звонков, кодов приложений и т.д., соединяя все это в соответствующие внутренние API. Затем Asterisk PBX Switching Core начинает прием звонков с интерфейсов и обрабатывает их в соответствии с планом набора, используя Application Launcher для звонящих телефонов, соединяя с голосовой почтой, набирая внешние линии и т.д. Также ядро обеспечивает стандартным Scheduler и I/O Manager, которыми пользуются приложения и драйверы.

Codec Translator Asterisk позволяет



загруженным кодеками каналам беспрепятственно говорить друг с другом. Универсальность и гибкость Asterisk идут от приложений, кодеков, драйверов каналов, файловых форматов и др., которые подсоединяются к разным параллельным интерфейсам Asterisk.

2.3 Образцы сетей

Asterisk невероятно гибкий для работы в сети. Вам будет полезно ознакомиться с несколькими примерами сетей, отраженных в данных схемах.

2.3.1 Mythical 1x1 PBX.



Часто задаваемый вопрос: «На сколько уменьшается PBX в сочетании с Asterisk?». Вы сможете уменьшить PBX до одного порта PSTN и одного порта аналоговой или IP телефонии. Да, это правда, вы можете сделать PBX с одним портом, но только в том случае, если вы любите оставлять себе голосовую почту или беседовать с автоответчиком.

В приведенной выше схеме аналоговый телефон подключен напрямую к ПК с Asterisk, и он обращается или к IP телефону через Интернет, или к аналоговому телефону через SLOOU UPS к конвертеру FXS, например. Такой PBX может обеспечить голосовую почту, работать как вход (gateway) или обеспечить базовый IVR скрипт (например, контролирует ваше овещение с X10 для всех домашних автоматических устройств).

2.3.2 8x16 Small Office PBX.



Типичная офисная схема сети чаще всего выглядит так: обеспечивается большая плотность телефонов внутри, а не снаружи телефонной сети. В данном примере 8 телефонных линий и 16 аналоговых телефонов объединены в канальный банк, который увеличивает каналы в один интерфейс T1, который, в свою очередь, проходит по кабелю всего несколько футов к ПК с Linux (с картой T1, например T100P), где Asterisk обеспечивает тоновый набор. К тому же телефоны VoIP можно подсоединить через Интернет независимо от количества телефонов, который можно использовать. Даже при такой, относительно небольшой, настройке вы можете воспользоваться конференциями, голосовой почтой (с индивидуальным почтовым ящиком для каждого), возможностью проверять голосовую почту через вэб, обычным скриптингом IVR, автоответчиком и другими функциями на зависть всем друзьям.

2.3.3 SME с удаленными офисами.



Одна из сильных черт Asterisk – это его способность соединять удаленные офисы SME (Small to Medium Enterprise-Малые- средние предприятия) вместе. Верхняя схема показывает, как вы можете создать индивидуальные маленькие PBX для многочисленных офисов, используя Asterisk, а затем соединить их вместе в единую прозрачную сеть.

2.3.4 Высокий уровень плотности IVR и конференции.



Asterisk можно использовать как IVR с высокой плотностью и платформу для конференции, применяя традиционные интерфейсы PRI/T1 и обеспечивая тем самым хороший уровень по отказоустойчивости и взаимодействию, используя TDM через Изернет, что позволяет Asterisk продвигать автобус TDM через сеть Изернет при минимальной задержке.

2.4 Устройство файловой системы.

Asterisk разработан в лучших традициях Linux и состоит из нескольких директорий.

/etc/asterisk

Эта директория содержит все конфигурационные файлы Asterisk. Более подробную информацию вы найдете в разделе о конференциях.

/usr/sbin

Системная двойная (бинарная) директория /usr/sbin содержит настоящие файлы asterisk, astman, astgenkey исполнения и скрипты Asterisk, включая safe_asteris.

/usr/lib/asterisk

Содержит бинарные объекты, относящиеся к Asterisk и его специальной архитектуре.

/usr/lib/asterisk/modules

Содержит модули запуска для приложений, драйверы каналов, кодеки, драйверы файлового формата и т.д.

/usr/include/asterisk

Содержит головные файлы, необходимые для создания приложений для Asterisk, драйверов каналов и другие загружаемые модули.

/var/lib/asterisk

Содержит разные данные, необходимые Asterisk в его обычной работе.

/var/lib/asterisk/agi-bin

Местоположение скриптов AGI, которые используются приложением AGI в плане набора.

/var/lib/asterisk/astdb

База данных Asterisk, грубо говоря, является эквивалентом Windows Registry. Этот файл никогда не используется напрямую, но его содержимое можно отобразить и изменить в командной линии Asterisk в наборе функций «database» .

/var/lib/asterisk/images

Хранилище изображений, относящихся к плану набора и приложениям.

/var/lib/asterisk/keys

Хранилище публичных и личных ключей, используемых для аутентификации RSA в Asterisk (в особенности, IAX).

/var/lib/asterisk/mohmp3

Хранилище MP3 музыки под рукой. Содержит любые mp3 по вашему желанию. Помните, что «музыка под рукой» должна быть сконфигурирована в /etc/asterisk/musiconhold.conf.

/var/lib/asterisk/sounds

Хранилище аудиофайлов, подсказок и т.п., используемое приложениями Asterisk. В последствии некоторые подсказки преобразуются в поддиректории в директории /var/lib/asterisk/sounds. |

/var/run

Asterisk сохраняет запуски, называемые pipe (трубки) и файлы PID в системной стандартной директории /var/run.

/var/run/asterisk.pid

Содержит первичные идентификаторы процесса (PID) запуска процессов Asterisk.

Описать самые важные из путей в заметке

/var/run/asterisk.ctl

Трубка с таким именем используется Asterisk для работы в удаленном режиме.

/var/spool/asterisk

Используется для запуска катушечных файлов голосовой почты, внешних звонков и т.д.

/var/spool/asterisk/outgoing

Необходим Asterisk для несвязанных звонков. Когда файл создается в /var/spool/asterisk/outgoing Asterisk пытается прочитать несвязанный звонок и, затем, он попадает в PBX, если на него отвечают. Более подробную информацию смотрите в разделе «Несвязанные звонки».

/var/spool/asterisk/qcoll

Создано для приложения qcoll, которые сейчас уже не используются. Не используйте его.

/var/spool/asterisk/vm

Хранилище ящиков голосовой почты, рекламы и папок.

2.5 Наименование каналов

Понимание правила наименования каналов в Asterisk является залогом эффективной работы. Имена внешним каналом (используется, например, с приложением Dial) даются в таком виде

```
<technology>/<dialstring>
```

Параметр <technology> означает вид интерфейса, который хотят создать или сделать на него ссылку (например, SIP, Zap, MGCP, IAX)- это специальная нить драйвера, указывающая на желаемое направление. В этом разделе рассказывается о правилах наименования всех типов каналов.

2.5.1 Zap: Zaptel TDM каналы

Исходящие:

Базовые форматы имени Zap канала:

```
Zap/[g]<identifier>[c][r<cadence>]
```

Где <identifier> является цифровым идентификатором номера физического канала желаемого канала. Если перед идентификатором стоит буква g, то номером мы будем понимать не номер канала, а номер группы. (см. Zapata.conf). После идентификатора могут быть одна или более опций. Если после идентификатора идут буква r и номер, то этот номер используется в качестве « очерченного круга» для этой команды набора (доступные номера 1-4) . Если после идентификатора следует буква c, то тогда требуется «Подтверждение ответа» (Answer Confirmation), при этом звонок не считается ответным до тех пор, пока принимающий звонок не нажмет кнопку #.

```
Zap/1 . TDM Channel 1  
Zap/g1 . First available channel in group 1  
Zap/3r2 . TDM Channel 3 with 2nd distinctive ring  
Zap/g2c . First available channel in group 2 with confirmation
```

Входящие:

Входящие Zap канала маркируются просто:

```
Zap/<channel>-<instance>
```

Где <channel> является номером канала, а <instance> – номером от 1 до 3, показывающим какой из 3-х логических каналов ассоциируется с одним физическим каналом.

```
Zap/1-1 . First call appearance on TDM channel 1  
Zap/3-2 . Second call appears on TDM channel 3
```

2.5.2 Начало работы протоколов каналов

Исходящие:

Обычно исходящие каналы принимают такую форму:

```
SIP/[<exten>@]<peer>[:<portno>]
```

Где <peer> является именем владельца IP удаленного сервера, <portno>- дополнительный номер порта (по умолчанию это SIP стандарт порт 5060), а <exten>- дополнительное расширение.

```
SIP/ipphone . SIP peer .ipphone.  
SIP/8500@sip.com:5060 . Extension 8500 at sip.com, port 5060
```

Входящие:

Входящие SIP каналы имеют такую форму:

```
SIP/<peer>--<id>
```

Где <peer> является идентифицированным владельцем, а <id>- случайный идентификатор для определения нескольких звонков от одного владельца.

```
SIP/192.168.0.1-01fb34d6 . A SIP call from 192.168.0.1  
SIP/sipphone-45ed721c . A SIP call from peer .sipphone.
```

2.5.3 IAX: Каналы внутри Asterisk обмена

Исходящие:

Исходящие каналы IAX имеют следующую форму:

```
IAX/[<user>[:<secret>]@]<peer>[:<portno>][/<exten>[@<context>][  
/<options>]]
```

Где <user> и <secret> являются дополнительными именами пользователя, а секрет служит для соединения с хостом, идентифицированным как <peer>, и номером дополнительного порта <portno>, дополнительно требующий специальное расширение <exten> в дополнительном контексте <context> и дополнительно со связью опций <options>, из которых только «а» сейчас определена для «запрошенного автоответа».

```
IAX/mark:asdf@myserver/6275@default . Call to .myserver.  
using .mark. as username and .asdf. as  
password, and requesting extension 6275 in  
default context  
IAX/iaxphone/s/a . Call to .iaxphone. requesting immediate answer.  
IAX/guest@misery.digium.com . Call Digium
```

Входящие:

Входящие каналы IAX имеют следующую форму:

```
IAX[ [<username>@]<host> ]/<callno>
```

Где <username> является именем пользователя, если оно известно, <host> - связующий хост и <callno>- местный телефонный номер.

```
IAX[mark@192.168.0.1]/14 . Call number 14 from user .mark. at  
192.168.0.1  
IAX[192.168.10.1]/13 . Call 13 from 192.168.10.1
```

Глава 3: Запуск Asterisk

Asterisk запускается сразу. Если он начинает работу без проблем, то он запускает daemon процесс. Часто бывает полезным запустить его в Verbose, консольном режиме, что обеспечит вас полезной информацией о запуске и о его состоянии, а также предоставит доступ к мощному интерфейсу командной строки Asterisk.

3.1 Аргументы командной строки Asterisk

Как и большинство приложений Linux Asterisk имеет несколько опций командной строки. Обычно они предворяются знаком «-», а несколько опций могут стоять в ряду после знака «-». Например

```
# asterisk -vvvgc
```

Это пример самой распространенной командной строки Asterisk

-c

Запускает режим консоли. Если этот режим запущен, Asterisk выдает командную строку, которая используется для формулирования команд и наблюдением за состоянием системы. Введите -f как обычно.

-C <configfile>

Исполняет Asterisk с другими конфигурационными файлами

-d

Запускает дополнительную загрузку по всем модулям

-f

Предотвращает daemonizing погружение Asterisk в фон.

-g

Заставляет Asterisk сбросить ядро в случае насильственной сегментации

-h

Отображает помощь в базовой командной строке

-i

Заставляет Asterisk подсказать криптографические инициализирующие пропускные коды на старте.

-n

Блокирует цветовую поддержку ANSI

-p

Запускает с приоритетом по времени

-q

Запускает в спокойном режиме

-r

Соединяет с уже запущенной частью Asterisk

-v

Заставляет Asterisk производить больше многословных выходов. Больше -v означает больше многословия

-x <command>

Выполняет команду в Asterisk (когда комбинируется с -r)

Отразить команды в
заметках

3.2 Интерфейс командной строки Asterisk

Командная строка Asterisk является одним из мощнейших интерфейсов для наблюдения за состоянием работающего Asterisk. Полное описание всех опций не может уместиться в рамках данного руководства, поэтому мы расскажем об этом коротко. Когда вы запускаете Asterisk с флажком *-r* или *-c*, тогда у пользователя будет подсказка Asterisk CLI, которая выглядит не впечатляюще, а именно так:

```
*CLI>
```

или

```
localhost*CLI>
```

В любом случае, находясь в командной строке, вы вводите инструкции путем ввпечатывания их и нажатия клавиш ввода. В CLI есть команда завершения, которая осуществляется нажатием клавиши *tab*. Самая полезная команда-это *help*, которая покажет вам список всех команд Asterisk CLI, которые можно ввести:

```
*CLI> help  
add extension Add new extension into context  
...  
zap show channel Show information on a channel  
*CLI>
```

Чтобы получить информацию о специфичной команде, напечатайте `help <command>`. Например:

```
*CLI> help soft hangup  
Usage: soft hangup <channel>  
Request that a channel be hung up. The  
hangup takes effect the next time the  
driver reads or writes from the channel  
*CLI>
```

Это значит, что закрывающая программу команда берет аргумент (имя канала), что, в свою очередь, требует закрытия канала. Эту команду, например, можно использовать для закрытия активного звонка в системе.

Еще несколько крайне полезных команд:

```
iax debug: Enable IAX debugging  
mgcp debug: Enable MGCP debugging  
reload: Reload configuration files  
restart when convenient: Restarts Asterisk when all calls are gone  
show agi: Displays AGI commands  
show applications: Shows all Asterisk apps  
show application <app>: Shows usage of a specific Asterisk app  
show channels: Shows all active channels  
show channel <channel>: Shows information on a specific channel  
sip debug: Enable SIP debugging  
stop now: Stops Asterisk immediately
```


Глава 4: План набора номера в Asterisk

Самое главное в Asterisk- это понять план набора номера. Именно план набора ведет каждый звонок в системе, от источника через разные приложения к пункту назначения. Все, что делается в голосовой почте, конференции в голосовом меню автоответчика, подчинено определенной концепции и логике.

4.1. Введение в расширенные контексты

4.1.1 Использование расширенных контекстов

План набора номера состоит из одного или более расширенных контекстов. Каждый расширенный контекст сам по себе является набором расширений. У каждого расширенного контекста в плане набора номера есть собственное имя, закрепленное за ним. Эти контексты могут быть использованы для введения номера по важным параметрам, включая:

Security . Permit long distance calls from certain phones only
Routing . Route calls based on extension
Autoattendant . Greet callers and ask them to enter extensions
Multilevel menus . Menus for sales, support, etc.
Authentication . Ask for passwords for certain extensions
Callback . Reduce long distance charges
Privacy . Blacklist annoying callers from contacting you
PBX Multihosting . Yes, you can have .virtual hosts. on your PBX
Daytime/Nighttime . You can vary behavior after hours
Macros . Create scripts for commonly used functions

Цель этой главы- познакомить вас с конституциями, стоящими за планом набора номера, показать несколько примеров и вооружить вас знаниями, необходимыми для выполнения тонких трюков, а также произвести впечатление на друзей, коллег и соперников знанием Asterisk.

4.1.2 Базовый расширенный контекстное

Вот пример расширенного контекста:

default	
<i>Extension</i>	<i>Description</i>
101	Mark Spencer
102	Wil Meadows
103	Greg Vance
104	Check voicemail
105	Conference Room
0	Operator

В этом контексте (под названием «по умолчанию») первые три расширения (101-103) связаны со звонящими телефонами, принадлежащими разным служащим. Четвертое расширение (104)- разрешение кому-то проверять их голосовую почту. Пятое расширение (105)- конференц-зал. Последнее «0» расширение-оператор.

4.1.3 Образец голосового меню

Другой пример расширенного контекста выглядит так:

mainmenu	
<i>Extension</i>	<i>Description</i>
s	Welcome message and instructions
1	Sales
2	Support
3	Accounting
9	Directory
#	Hangup

У этого примера с названием «Главное меню» есть только одно цифровое расширение. Расширение «s» -это расширение Start, где звонящий начинает. Это расширение играет роль послания вдоль строчек «Спасибо за звонок в нашу компанию. Нажмите 1 для отдела продаж, 2 для поддержки, 3 для бухгалтерии, 9 для администрации компании или # для сброса номера.» Каждое меню опции является, по сути, расширением и может или набрать чье-то настоящее расширение, или отправить кого-то, например. В другое меню.

4.1.4 Образец постановки расширения

Расширения могут также подходить под образцы, а не оставаться только цифрами. Чтобы образцы стали подходящими, они должны запускаться со знаком «-» и могут использовать любую из данных специальных характеристик:

X	– Any digit from 0-9
N	– Any digit from 2-9
[14-6]	– Any a 1,4, 5, or 6
.	– Matches anything

Рассмотрим этот контекст в качестве примера

routing	
<i>Extension</i>	<i>Description</i>
_61XX	Dallas Office
_62XX	Huntsville Office
_63XX	Dallas Office
_7[1-3]XX	San Jose Office
_7[04-9]XX	Los Angeles Office

Этот контекст (по названию «routing») объединяет звонки по их расширению, чтобы потом отправить их на разные серверы. В этом примере все расширения состоят из 4-х знаков (у Asterisk нет, конечно, ни таких требований, ни требований по поводу одинаковой длины у расширений. Однако, все, что начинается с 61 будет отправлено в офис в Далласе, с 62- в офис в Хантсвилле,

и т.д. Все, что начинается с 71, 72,73 будет направлено в Сан-Хосе.)

4.1.5 Контекстное присоединение

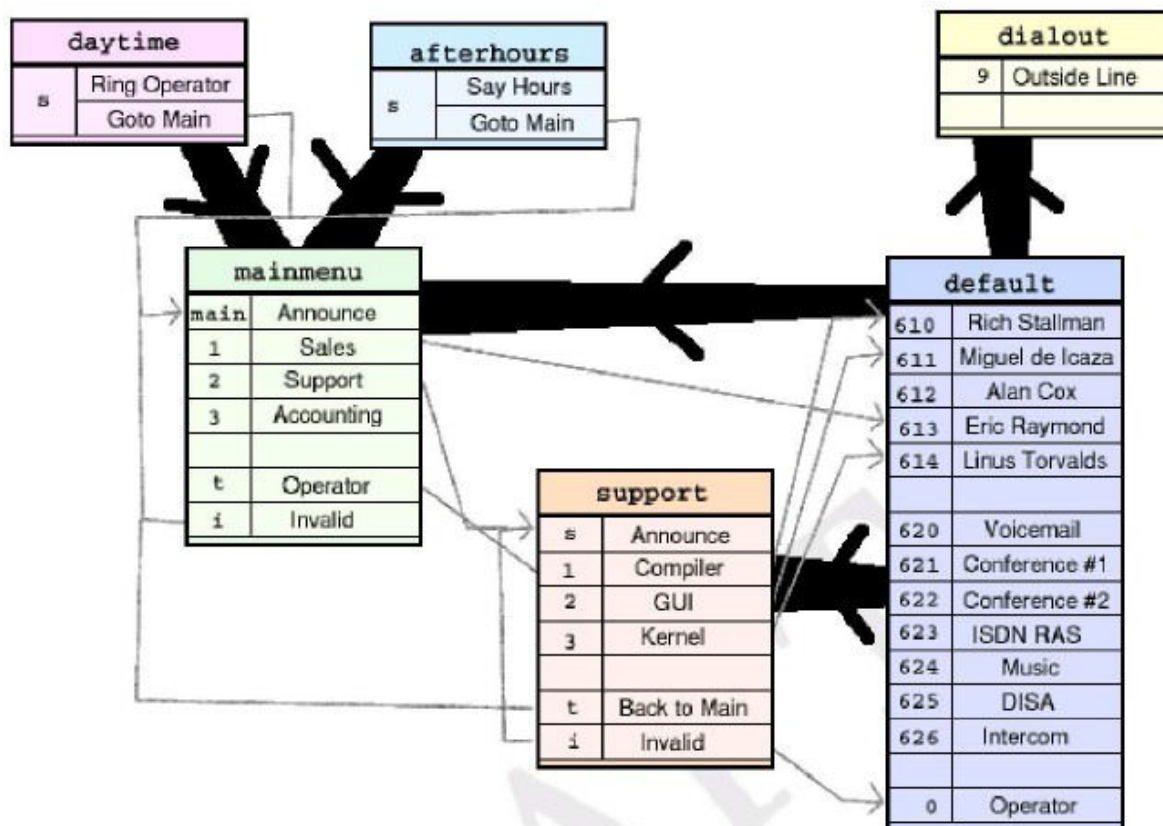
Один расширенный контекст может присоединять к себе содержание другого. Для примера рассмотрим данный контекст:

longdistance	
<i>Extension</i>	<i>Description</i>
_91NXXNXXXXXX	Long distance calls
include => "local"	

local	
<i>Extension</i>	<i>Description</i>
_9NXXXXXX	Local calls
include => "default"	

Здесь контекст под названием «local» обеспечивает одно расширение для набора местных номеров и содержит, также, расширение «по умолчанию». Далее, существует контекст «длинная дистанция», который содержит расширение для звонков на длинное расстояние, а также содержит «местный» контекст. Телефоны, находящиеся в контексте «длинная дистанция» смогут звонить на удаленные телефоны. Находящиеся в контексте «местный», могут набирать только местные 7-значные номера, а те, кто находятся в контексте «по умолчанию», не будут иметь доступа ко внешней линии вообще. Таким образом, используя расширенные контексты, вы можете тщательно контролировать доступ ко всем услугам.

4.2 Полный набор контекстов



Приведем подробный пример плана набора номера. Здесь дан план набора номера в вымышленной компании, использующей Open Source. Перечислены несколько контекстов со знакомыми названиями. Крупные черные линии указывают на то, какой из расширенных контекстов содержит в себе другой контекст. Тонкими серыми линиями показаны расширения, которых направляют вас в другое расширение или расширительный контекст. Расширения, являющиеся местными в рамках вашей компании, должны быть перечислены под контекстом default (по умолчанию) в связи с тем, что они должны быть доступны всем сотрудникам практически в любое время. Обычно ни один из телефонов не связан только с контекстом «по умолчанию», но он может быть связан с VoIP счетом, например. Местный (в нашем случае dialout) контекст может содержать в себе контекст по умолчанию или только местный или только набор удаленных номеров. Важно, что доступ к местному контексту не разрешается с «гостевых» счетов до тех пор, пока вы не разрешите использовать чужим людям ваши местные телефонные ресурсы. Также у нас есть пара контекстов меню-mainmenu (главное меню) и support (поддержка). Оба этих меню содержат контекст по умолчанию так, что прямые расширения могут быть набраны в любое время. Контекст mainmenu включает в себя daytime (дневное время) и afterhours (остальное время), что позволяет

входящему звонку попадать вначале к оператору в дневное время, и, напрямую, на объявление о режиме работы компании в вечернее время.

4.3 Определение расширений

В отличие от традиционного PBX, где расширения связаны с телефонами, интерфейсами, меню и т.д. в Asterisk расширение дается как список приложений (и аргументов) для запуска. Каждый шаг расширения рассматривается как приоритет. В целом, каждый приоритет выполняется в порядке очереди, хотя приложения (особенно, «Dial» и «Goto») могут перенаправить звонок к другому приоритету. Когда набирается расширение, то приоритет выполняется до тех пор, пока звонок не завершится или приложение не вернется к -1, или звонок не отправится к новому расширению. Обычно, каждый шаг в расширении помечается так:

```
exten => <exten>,<priority>,<application>, [(<args>)]
```

4.3.1 Пример базового расширения

Рассмотрим следующий пример:

```
exten => 100,1,Wait(1)
exten => 100,2,Answer
exten => 100,3,Playback(demo-congrats)
exten => 100,4,Hangup
```

Здесь создано расширение из 4 шагов. Когда звонок входит в это расширение, то вначале Asterisk ждет одну секунду. Затем Asterisk отвечает на звонок (если на него еще не ответили). В третьих, он запускает файл «demo-congrats» и проигрывает его звонящему. Если звонящий прекращает разговор в момент работы Asterisk в данном расширении, то этот процесс будет завершен в этот же момент.

4.3.2 Набор номера

Самый распространенный вид расширения используется для набора вызова другого интерфейса. Этот вызов осуществляется при помощи приложения «Dial». У этого приложения очень большой список опций (см. информацию по Dial (набор номера)), но в этом примере мы приводим лишь базовую форму:

```
exten => 100,1,Dial(Zap/1,20)
exten => 100,2,VoiceMail(u100)
exten => 100,102,VoiceMail(b100)
```

В этом примере показано одно из нескольких исключений по исполнению расширения, идущего не по порядку. Когда приходит такое расширение, первое действие Asterisk- это попытка дозвониться до интерфейса «Zap/1» в течение, максимум, 20 секунд. Если интерфейс занят, он перепрыгивает к интерфейсу n+101, если такой приоритет существует в данном расширении. В нашем случае есть такой приоритет (102), который направляет звонящего к голосовому ящику 100, предварив это действие объявлением «занято» (что-то вроде «Абонент в 100-ом расширении на связи»). Если ответа нет (или если занято, а у нас не было шага 102), тогда исполнение переходит к шагу 2, в котором звонящего переводят в голосовой почтовый ящик 100, но с объявлением о недоступности абонента (что-то вроде «Абонент в 100-м расширении временно недоступен»).

4.3.3 Маршрут ID-звонящего

Этот пример, известный как «Anti-Ex Girlfriend» («Меры против бывшей подружки»), расширения показывает, как Asterisk может обеспечивать связь не только вызываемым номерам, но и вызывающим номерам:

```
exten => 100/2565551212,1,Congestion
exten => 100,1,Dial(Zap/1,20)
exten => 100,2,VoiceMail(u100)
exten => 100,102,VoiceMail(b100)
```

Этот пример построен на основе предыдущего примера путем добывления специального правила, которое гласит: если звонящий- 2565551212 (его ведет ID- звонящий, индикатором которого является «/», и номер ID-звонящего немедленно подойдет следующему), им немедленно будет представлен тон Congestion. Другие звонящие будут дозваниваться в обычном режиме. Обычный пример маршрута ID-звонящего выглядит так:

```
exten => 100/1,Zapateller  
exten => 100,1,Wait(0)  
exten => 100,2,Dial(Zap/1)
```

В данном примере, если звонок принимается без ID-звонящего, то тогда запускается приложение Zapateller (оно запускает известный всем тон «специальная информация», который вы слышите при дозвоне по номеру, находящемуся вне зоны обслуживания, и часто требуется время, чтобы авто-набор рассоединился). Если ID звонящий обеспечен связью, тогда сигнал «Ждите» осуществляется за 0 секунд (другими словами, без временных затрат).

В ином случае, канал Zap/1 звонит неопределенно (например, без остановок).

4.3.4 Последовательный набор номеров

Часто хочется, чтобы данное расширение вначале звонило на один номер, а затем, если нет ответа, звонило на другой номер (или набор номеров).

Рассмотрим пример «Оператор»:

```
exten => 0,1,Dial(Zap/1,15)  
exten => 0,2,Dial(Zap/1&Zap/2&Zap/3,15)  
exten => 0,3,Playback(companymailbox)  
exten => 0,4,Voicemail(100)  
exten => 0,5,Hangup
```

В данном примере, когда звонящий набирает «0» для оператора, мы пытаемся задействовать интерфейс Zap/1 (который является телефоном, используемы в приемной, например). Если этот интерфейс занят или нет ответа в течение 15 секунд, то мы пытаемся позвонить группе телефонов (включая номер

приемной снова) в течение других 15 секунд. Если опять нет ответа (или если все заняты), то запускается объявление о недостаточности абонента и о возможности оставить сообщение в почтовом ящике компании. В завершении, звонящий попадает в голосовой почтовый ящик 100 без дополнительных объявлений.

4.3.5 Базовое голосовое меню

Типичное голосовое меню выглядит как собственный расширенный контекст:

```
[sales]
exten => s,1,Background(welcome-sales)
exten => 1,1,Goto(default,100,1)
exten => 2,1,Goto(default,101,1)
[mainmenu]
exten => s,1,Background(welcome-mainmenu)
exten => 1,1,Goto(sales,s,1)
exten => 2,1,Dial,Zap/2
exten => 9,1,Directory(default)
exten => 0,1,Dial,Zap/3
```

Обычно объявление проигрывается с расширением «S» над входом в меню. Затем приложение «Background» проигрывает подсказку в ожидании входа в расширение. В приведенном примере мы видим два меню, одно из которых называется «mainmenu» (главное меню), а другое - «sales» (продажи). Когда звонящий входит в контекст «mainmenu», то он услышит объявление (что-то вроде «Напишите 1 для отдела продаж, 2- для поддержки, 9-для администрации, 0-для оператора»). Перед входом в 1, звонящего переведут в меню «sales», которое по очереди будет представлять другие опции. Набор 2 вызовет Zap/2, 0- Zap/3, а 9 предоставит пользователю доступ к администрации компании.

4.3.6 Использование переменных величин

Asterisk извлекает пользу из глобальных и канальных переменных величин для аргументов приложений. Эти величины выражаются в плане набора номера как \${foo}, где «foo» является именем переменной величины.

Переменной может быть любое буквенно-числовое выражение, начинающееся с буквы. Однако, есть несколько переменных, чьи названия имеют специальные значения, особенно:

```
{CONTEXT} . The current context
{EXTEN} . The current extension
{EXTEN:x} . The current extension with x leading
digits dropped
{PRIORITY} . The current priority
{CALLERID} . The current Caller ID (name and number)
{CALLERIDNUM} . The current Caller ID number
{CALLERIDNAME} . The current Caller ID name
{RDNIS} . The current redirecting DNIS
```

Глобальные переменные могут быть определены в разделе [globals] в плане набора номера. Рассмотрим данный пример:

```
[globals]
MARK => Zap/1
GREG => Zap/2&SIP/pingtel
WIL => Zap/3
JUDY => Zap/4
[mainmenu]
exten => 1,1,Dial({GREG}&{MARK})
exten => 2,1,Dial({WIL}&{JUDY})
exten => 3,1,Dial({JUDY}&{MARK})
```

Организуя план набора номера таким образом, легко будет изменить физические интерфейсы под любого пользователя и, в то же время, иметь все ссылки на них в новом плане набора.

4.3.7 Включенные контексты

Один контекст может включать в себя ноль или более других контекстов, дополнительно к ограничениям по данным/времени. Контексты включаются в том порядке, в котором они перечислены. Включение осуществляется в таком формате:

```
include => <context>[|<hours>|<weekdays>|<monthdays>|<months>]
```

Где <context> является контекстом, который надо включить, <hours>- часы, в которые это включение будет полезным (в форме расписания, в военное время, например, с 9:00 до 17:00), <weekdays>- нужные дни недели (например, понедельник- пятница), <monthdays>- полезные дни месяца (например, 22-25) и <months>-полезные месяцы. Рассмотрим следующий пример:

```
[salespeople]
exten => 1000,1,Dial(Zap/1)
exten => 1000,2,VoiceMail(u1000)
exten => 1001,1,Dial(Zap/2)
exten => 1001,2,VoiceMail(u1001)
[techpeople]
exten => 2000,1,Dial(SIP/2000)
exten => 2000,2,VoiceMail(u2000)
exten => 2001,1,Dial(SIP/2001)
exten => 2001,2,VoiceMail(u2001)
[default]
include => salespeople
include => techpeople
```

В этом примере контекст по умолчанию содержит в себе два других контекста, тем самым делая контекст меньше и легче для понимания.

4.3.8 Режимы работы День/Ночь

Включенные контексты можно внедрить в режимы День/Ночь (и даже режим «праздники»), используя преимущества включений во времени и датах. Рассмотрим данный пример:

```
[newyears]
exten => s,1,Playback(happy-new-years)
[daytime]
exten => s,1,Dial(Zap/1,20)
[nighttime]
exten => s,1,Playback(after-hours-msg)
[default]
include => newyears|1|jan
include => daytime|9:00-17:00|mon-fri
include => nighttime
```

В этом примере нормальный режим работы происходит в ночное время.

4.3.9 Несвязанный набор номера

Несвязанный набор номера можно осуществить напрямую, связав короткое расширение (например, «9») с несвязанной линией или установив полные расширения для набираемых номеров. Рассмотрим пример:

```
[directdial]
ignorepat => 9
exten => 9,1,Dial(Zap/g2/)
exten => 9,2,Congestion
[international]
ignorepat => 9
exten => _9011.,1,Dial(Zap/g2/${EXTEN:1})
exten => _9011.,2,Congestion
include => longdistance
[longdistance]
ignorepat => 9
exten => _91NXXNXXXXXX,1,Dial(Zap/g2/${EXTEN:1})
exten => _91NXXNXXXXXX,2,Congestion
include => local
```

```
[local]
ignorepat => 9
exten => _9NXXXXXX,1,Dial(Zap/g2/${EXTEN:1})
exten => _9NXXXXXX,2,Congestion
include => default
```

Этот пример создает 4 отдельных контекста с разными уровнями доступа к телефонной сети. Представим, что вы хотите, чтобы цифра «9» стала соединяющей цифрой с внешней линией. Линии ignorepat инструктируют драйверы каналов Asterisk не убирать тон набора, когда происходит набор по данному образцу. Таким образом, даже после набора цифры 9 будет звучать тон набора. Контекст local (местный) может набирать только семизначные номера в дополнение к тому, что есть в контексте по умолчанию. Звонки отправляются любым каналом в «группе 2» драйвера Zaptel после удаления «9». Контексту longdistance разрешено набирать любые номера от 1, а так же все, что есть в контексте local. Контекст international дает звонящему возможность дозваниваться до любого номера, начинающегося с 011+, а также до всего, что есть в контексте longdistance. Контекст directdial (прямой набор) соединяет пользователя напрямую с главной линией, когда звонящий набирает цифру 9.

4.3.10 Сбой на главной линии и LCR

Одна из наиболее выгодных черт Asterisk- это его способность строить простые таблицы Least Cost Routing (LCR)- путь наименьшей стоимости- включая и ситуацию сбоя на главной линии. Рассмотрим этот оптимизированный план набора номера:

```
[tolllongdistance]
exten => _91NXXNXXXXXX,1,Dial(Zap/g2/${EXTEN:1})
exten => _91NXXNXXXXXX,2,Congestion
[hsvlongdistance]
exten => _91256NXXXXXX,1,Dial(IAX/hsv/${EXTEN})
exten => _91256NXXXXXX,2,Dial(Zap/g2/${EXTEN:1})
exten => _91256NXXXXXX,3,Congestion
[longdistance]
include => hsvlongdistance
include => tolllongdistance
```

```
include => local
```

В этом примере контекст `long distance` (на долгие расстояния) настроен на использование удаленного VoIP хоста под названием `hsv` (в основном, в Хантсвилле), чтобы набирать номера с местным кодом 256. Если это не удастся сделать, то тогда будет использован интерфейс TDM группа 2 (`toll call`) (в случае, если хост недоступен, например).

4.3.11 Использование Macros

В связи с тем, что логика расширений Asterisk весьма гибкая, он может создавать много подобных друг другу расширений. Чтобы упростить эту задачу, мы воспользуемся преимуществами `macros`, которые упрощают планы набора номеров и облегчают изменение потоков на большой шкале. `Macros` внедряются в систему путем создания расширенного контекста, чье имя начинается с «`macro-`», а затем идет имя `macro`. Исполнение начинается с «`s`» расширения и заканчивается как только расширение попадает на место, которое больше не находится внутри `macro`. `Macros` определяют несколько полезных местных вариаций, а именно:

```

${MACRO_EXTEN} . The extension calling the macro
${MACRO_CONTEXT} . The extension context calling the macro
${MACRO_PRIORITY} . The active priority when the macro was called
${MACRO_OFFSET} . If set, causes the macro to attempt to return to n + ${MACRO_OFFSET}
${ARGn} . The nth argument passed to the macro.
```

Рассмотрим данный пример:

```
[macro-oneline]
;
; Standard one-line phone.
;
; ${ARG1} . Device to use
;
exten => s,1,Dial(${ARG1},20)
exten => s,2,VoiceMail(u${MACRO_EXTEN})
exten => s,3,Hangup
exten => s,102,VoiceMail(b${MACRO_EXTEN})
exten => s,103,Hangup
[macro-twoline]
;
; Standard two-line phone.
;
; ${ARG1} . First phone
; ${ARG2} . Second phone
;
exten => s,1,Dial(${ARG1},20)
exten => s,2,VoiceMail(u${MACRO_EXTEN})
exten => s,102,Dial(${ARG2},20)
exten => s,103,VoiceMail(b${MACRO_EXTEN})
[default]
exten => 1000,1,Macro(oneline,Zap/1)
exten => 1001,1,Macro(oneline,SIP/1001)
exten => 1002,1,Macro(twoline,Zap/3,Zap/4)
```

После выполнения этой сложной работы по определению однолинейного тасго для одинарной линии телефона и двулинейного тасго для двулинейного телефона, внедрение контекста по умолчанию становится весьма легким делом, а каждое расширение требует только одинарной линии вместо нескольких подобных линий.

Глава 5: Конфигурационные файлы

5.1 Введение в конфигурационные файлы

Гибкость Asterisk чаще всего контролируется через конфигурационные файлы, расположенные в директории `/etc/asterisk`. Этот конфигурационный синтаксис был создан для упрощенного разбора как со стороны программного обеспечения (например, конфигурация GUI интерфейсов), так и со стороны человека (например, вас). Формат конфигурационных файлов Asterisk, по иронии судьбы, очень похож на формат `win.ini` времен Microsoft® Windows 3.1. Файлом является отформатированный файл ASCII, поделенный на секции, озаглавленные одним секционным именем в квадратных скобках; после него идут слова `keyword` и `value`, разделенные знаком `=` или `=>`. Точка с запятой имеет комментирующий характер (т.к. знак `#` часто используется в расширениях). Не обращайте внимание на пустые строки. Здесь приведен пример конфигурационного файла:

```
;
; The first non-comment line in a config file
; must be a section title
;
[section1]
keyword = value ; Variable assignment
[section2]

keyword = value
object => value ; Object declaration
```

Конфигурационный грамматический разборщик Asterisk интерпретирует знаки `<=>` и `<=>` одинаково, а синтаксис используется только для того, чтобы наглядно выделить пары, представляющие опции. И пары, представляющие создание какого-либо объекта.

5.2 Грамматика конфигурационных файлов

Несмотря на то, что у конфигурационных файлов Asterisk один и тот же синтаксис, существует, по крайней мере, три вида грамматики для специального пользования.

5.2.1 Простые группы (например voicemail.conf)

Формат «Simple Groups» (простые группы) является (и это неудивительно) простейшим форматом и используется такими конфигурационными файлами, в которых объекты задекларированы как all options on the same line (все опции на одной строке). Примеры содержат extensions.conf, meetme.conf, voicemail.conf и другие. Рассмотрим данный пример

```
[mysection]
object1 => option1a,option2a,option3a
object2 => option1b,option2b,option3b
```

В этом примере «object1» создается при помощи опций «.option1a», «option2a» и «option3a», а «object2» - при помощи «.option1b», «option2b» и «option3b».

Индивидуальные единства.

Конфигурационный синтаксис «Individual Entities» (индивидуальные единства) используются конфигурационными файлами. В которых объекты задекларированы со многими опциями, и эти опции редко используются с другими объектами. В данном формате каждая секция связана с каждым объектом (иногда существует общая или подобная секция для всех общих конфигурационных опций). Например6

```
[general]
globaloption1=globalvalue1
globaloption2=globalvalue2
[object1]
option1=value1a
option2=value2a
[object2]
option1=value1b
option2=value2b
```

В этом примере общая секция определяет два общих варианта «globaloption1» и «globaloption2» со значениями «globalvalue1» и «globalvalue2»

соответственно. Затем создаются два объекта («object» и «object2») при помощи каждой из этих двух опций.

5.2.2 Унаследованные опции объекта (например, zapata, conf)

Формат «Inherited Option Object» (унаследованные опции объекта) используется zapata.conf, phone.conf, mgcp.conf и другими интерфейсами, которые имеют много опций, при этом большинство интерфейсов или объектов делят одно значение для опций. В этом классе конфигурационных файлов обычно существует еще одна секция, которая содержит декларации одного или более каналов (или объектов). Опции объектов определены над декларацией объекта и могут меняться для другой декларации. Возможно, эта конституция сложна для понимания, но если вы разберетесь в этом вопросе, то в дальнейшем вы убедитесь, как проста она в использовании. Рассмотрим этот базовый пример:

```
[mysection]
option1 = foo
option2 = bar
object => 1
option1 = baz
object => 2
```

Первые две строки устанавливают значение опций «option1» и «option2» к «foo» и «bar» соответственно. Когда объект «1» сформирован, то он затем создается при помощи его опции 1=«foo» и опции 2=«bar». После объявления об объекте «1», мы меняем значение опции «1» на «baz» и создаем новый объект «2». Теперь объект «2» создается при помощи опции 1=«baz» и опции 2, остающейся «bar» как и в объекте «1». Изменение значения «option1» не влияет на его значение в объекте 1, а только в объекте 2.

5.2.3 Полное единство объекта (например, iax.conf)

Формат «Inherited Option Object» (полное единство объекта) используется iax.conf, sip.conf и другими интерфейсами, в которых есть много единств со многими опциями, которые обычно имеют мало общих настроек. Каждое единство получает собственный контекст (иногда существует резервный контекст, например, общий для всех настроек) . Затем опции определяются в контекстной декларации. Например:

```
[myentity1]
option1=value1
option2=value2
[myentity2]
option1=value3
option2=value4
```

У единства myentity1 есть значения value1 и value2 для опций option1 и option2 соответственно. У единства myentity2 есть значения value3 и value4 для опций option1 и option2 соответственно.

5.3 Интерфейсы каналов

В этом разделе мы подробно рассмотрим конфигурационные файлы для различных драйверов каналов в Asterisk.

5.3.1 zapata.conf

Краткий обзор.

Файл zapata.conf содержит параметры, относящиеся к каналам TDM, находящимся в слое интерфейса zapatel. Каналы должны быть определены в этом файле до того, как они будут использоваться Asterisk. В дополнение скажем, что количество модификаций по работе Asterisk с каналами можно сконфигурировать здесь же.

Устройство.

Файл `zapata.conf` состоит из ключевого слова и пар значений. Ключевые слова служат настройке параметров для работы с каналами. Они могут быть булевские или могут содержать определенные значения под ключевое слово. Большая часть ключевых слов настраивает параметры для работы каналов. Значения будут действительными для всех последующих определений каналов до тех пор, пока их не «заездят».

Ключевые слова.

Ключевые слова находятся в `zapata.conf`

контекст: определяет исходный контекст для канала. Этот контекст будет необходим каналу в начале звонка. Помните, что контексты являются важной частью местной безопасности. Исходный контекст будет управлять доступностью расширений в данном канале. Если расширение размещается в контексте. Отличном от исходного. Тогда оно будет недоступно звонящему.

```
context = default
```



Важное примечание: Неаккуратное

использование контекстов ведет
к доступу в счетные услуги
и внутренние модификации

канал: Определяет канал или ряд каналов. Определение каждого канала унаследует все опции, заявленные перед ним в файле. Каналы могут определяться индивидуально, отделяясь друг от друга запятыми, или как ряд, отделенный дефисом.

```
channel => 16  
channel => 2,3  
channel => 1-8
```

группа: позволяет использовать некоторое количество каналов как один с целью набора номера. Для набора номера каналы будут вызываться на первой доступной основе. Для дозвона на станцию все каналы группы будут звонить вместе. Членство в многочисленной группе будет обозначено запятыми, а отсутствие группового членства- отсутствием перечисления после знака равенства.

```
group = 1  
group = 2,3  
group =
```

тип включения: настраивает тип сигнализации для линии PRI. Принимаемые значения:

```
national: National ISDN  
dms100: Nortel DMS100  
4ess: AT&T 4ESS  
5ess: Lucent 5ESS  
euroisdn: EuroISDN
```

```
switchtype = national
```

pri-dialplan: настраивает опцию, необходимую для нескольких (редко) включений, что требует проверки параметра набора номера. Многие включения PRI игнорируют эту опцию. Она бывает необходимой только в некоторых частях «железа». Доступные опции unknown, local, private, national, and international .

```
pri_dialplan = national
```

Почти всегда эту опцию можно оставлять ненастроенной.

сигнализация: настраивает тип сигнализации для последующих определенных каналов. Эти параметры должны подходить каналам как это определено в `/etc/zaptel.conf`. Правильный выбор основывается на имеющемся у вас «железе». Asterisk не запустится, если определение сигнализации канала неправильное или неработоспособное, если предложения не подходят `zaptel.conf`, или если устройство отсутствует или несконфигурировано должным образом.

Законные значения для сигнализации:

`fxo_ks`: FXO Kewlstart signalling. Used to signal an FXS device within the system, which would normally drive a handset or other station device. Kewlstart is Loopstart with Disconnect Supervision.
`fxs_ks`: The opposite side of `fxo_ks`. To signal an internal (or T1 connected) FXO device.
`fxo_gs`: Use FXO groundstart signalling.
`fxs_gs`: Use FXS groundstart signalling.
`fxo_ls`: Use FXO loopstart signalling
`fxs_ls`: Use FXS loopstart signalling
`pri_cpe`: Use PRI signalling, customer equipment side. Used when terminating a PRI line into Asterisk channels.
`pri_net`: Use PRI signalling, network side.
`em`: Use E&M signalling
`em_w`: Use E&M wink signalling
`featd`: Feature Group D, Adtran compatible. For use with the Atlas and similar equipment made by Adtran (DTMF version).
`featdmf`: Standard Feature Group D (MF version).
`featb`: Feature Group B



Важное примечание: Сигнализация

аналогового телефона может стать источником путаницы. Каналы FXS сигнализируют FXO и наоборот. «Разговоры» Asterisk с внутренними устройствами являются обратной стороной. Карта интерфейса FXO сигнализируется

FXS Asterisk и должна быть
сконфигурирована таким образом.

```
signalling => fxs_ks  
signalling => feaTd
```

Продвижение аналогового звонка

Мы хотим вас убедить в том, что использование аналоговых каналов не дает нам выхода на цифровую информацию, в то время как линии, подобные PRI, представляют нам информацию по продвижению звонка.

busydetect: это попытка отследить стандартный сигнал «занято» на аналоговом (FXS и FXO) или на конкретных сигнальных типах T1 (E&M, Wink, Feature Group D). Эта опция используется при определении момента завершения звонка или момента внутренней работы Asterisk при условии «занято».

Принимает «yes» или «no».

callprogress: Используется в сочетании с разными телефонными линиями. Он побуждает Asterisk наблюдать за состоянием звонка и отслеживать звонок, «занято» и линию разговора. Помните, что все это полностью не поддерживается технологией телефонной линии и возможны ошибки, особенно в отслеживании разговоров. Это работает только с телефонными танами US (на момент написания этого руководства). Принимает «yes» или «no».

```
busydetect = yes  
callprogress = yes
```

Опции множественной связи PPP (для PRI, требует сетевой поддержки): Эти опции используются при настройке опций множественной связи PPP на линиях PRI, которые поддерживает их. Множественная связь PPP- это технология, позволяющая каналам на PRI динамично растягиваться между голосом и данными.

Asterisk может взять голосовые каналы, расположенные в ней, набрать сервер удаленного доступа и перевести каналы в специальное расширение, которое доставляет канал к слою данных zaptel (см. zapRas).

minunused: Доступное минимальное количество неиспользуемых каналов.

Если доступно меньше каналов, то Asterisk не будет пытаться связаться с такими каналами и передаст их в данной связи. Принимает целое число.

minidle: Минимальное количество резервных каналов для связи с данными.

Asterisk составляет эти каналы открытыми чаще для данных, чем для голосовых каналов. Принимает целое число.

idledial: Резервный номер. Обычно этот номер используется при дозвоне к серверу удаленного сервера (RAS). Резервные каналы для данных направляются к этому расширению. Принимает целое число, что не приводит к конфликту с другим расширением в плане набора номера, и определяется как idleext.

idleext: Используется в качестве резервного расширения. Принимает значение в форме [exten@context](#). Обычно это расширение запускает приложение zapRas.

```
minunused => 2
minidle => 1
idleext => 6999@idle
idledial => 6999
```

Временные параметры:

Эти ключевые слова используются только с (не PRI) линиями T1. Все значения даются в миллисекундах. В большинстве конфигураций они не требуют настройки, как и работа «по умолчанию» в большинстве деталях «железа». Замечено, что обычный Adran Atlas использует длинные мигания около 300 миллисекунд и эти каналы следует конфигурировать

соответствующим образом.

prewink: Устанавливает время перед миганием
preflash: Устанавливает время перед вспышкой
wink: Устанавливает время мигания
rxwink: Устанавливает время приема мигания
rxflash: Устанавливает время приема вспышки
flash: Устанавливает время вспышки
start: Устанавливает время начала
debounce: Устанавливает время разграничения

```
rxwink => 300  
prewink => 20
```

ID опции звонящего

Эти ключевые слова устанавливают разные ID опции звонящего, такие как выключение отдельных модификаций и настройка ID линии звонящего на каналы. Многие из этих опций существуют по умолчанию.

Эти три опции являются булевскими (yes/no).

usecallerid: Блокирует или осуществляет передачу ID звонка по следующим каналам.

hidecallerid: Прячет внешний ID звонок, по усмотрению. По умолчанию настроен на «нет».

calleridcallwaiting: Решает, принять или нет ID звонок во время индикации «ожидание звонка».

```
usecallerid => yes  
hidecallerid => no
```

callerid: Настраивает линию ID звонящего на данный канал. Это ключевое слово поддерживает хорошо отформатированную строку, содержащую имя и телефонный номер, как звонок ID. Звонок может быть установлен в asreceived на стволе интерфейса, чтобы продвигать полученный ID звонок вперед.



● Важное примечание: Звонок ID

может быть передан на общую телефонную сеть при наличии поддерживающего «железа», как PRI. Невозможно настроить внешний ID звонок на аналоговой линии. В поддерживающих системах телефонная компания только получает номер, а имя берет из своих записей.

```
callerid = "Mark Spencer" <256 428-6000>  
callerid =  
callerid = asreceived
```

Опции по характеристикам звонка

Эти опции поддерживают или блокируют доступность передовых модификаций, предлагаемых Asterisk, таких как 3-направленный звонок и продвижении звонка по интерфейсам FXS (с сигналом FSO). Все эти опции являются булковскими (yes/no).

threewaycalling: Устанавливает, разрешить или не разрешить 3-направленный звонок из канала.

cancallforward: Блокирует или поддерживает продвижение звонка. Такое продвижение активируется при помощи *72, а деактивируется- *73.

transfer: Блокирует или поддерживает передачу вспышки звонка. Чтобы эта опция работала, настройте 3-направленный звонок на «да».

Immediate: Когда Asterisk находится в быстром режиме, вместо того, чтобы обеспечивать тон набора и читать цифры, он немедленно входит в расширение «s». Часто это называют режимом batphone.

adsi: Полностью поддерживает или блокирует поддержку ADSI. Спецификация ADSI- это система, подобная Caller ID, которая пропускает закодированную информацию к аналоговому набору. Она разрешает создавать интерактивные визуальные меню на многолинейном дисплее, предлагающие доступ к таким услугам, как голосовая почта через текстовый интерфейс.

```
threewaycalling = yes
transfer = yes
immediate = no
adsi = yes
cancallforward = yes
```

Опции по настройке качественного звука.

Эти опции настраивают определенные параметры Asterisk, влияющие на качество аудио в Zapata каналах.

echocancel: Блокирует или поддерживает удаление эхо. Практически в каждой конфигурации рекомендуется не трогать эту опцию или оставлять неустановленной, т.к. она работает по умолчанию. Принимает «да», «нет» или несколько цифр. Возможные значения- 16, 32, 64, 128 или 256.

echocancelwhenbridged:? Поддерживает или блокирует удаление эхо во время TDM звонков по мосту. В принципе, такие звонки не нуждаются в удалении эха, но часто звуковое исполнение улучшается благодаря этой опции. Можете настроить эту опцию, а можете не настраивать. Принимает «да» или «нет».

rxgain:Отлаживает получаемую громкость. Используется для повышения или понижения входящей громкости звука, чтобы компенсировать разницу в «железе». Диапазон возможности этой опции от -100% до +100%.

txgain:Используется для повышения или понижения исходящей громкости звука, чтобы компенсировать разницу в «железе». Принимает те же самые аргументы, что и rxgain.

```
echocancel = yes  
echocancelwhenbridged = no  
rxgain = 20%
```

Опции по логинам звонка

Эти опции меняют способ записи звонков в детальных записях звонков, производимых Asterisk.

amaflags:Настраивает флаги АМА, влияющие на разграничение записей по категориям в детальных записях звонков. Принимает следующие значения:

```
billing: Mark the entry for billing  
documentation: Mark the entry for documentation.  
omit: Do not record calls.  
default: Sets the system default.
```

accountcode:Настраивает отчетный код для звонков, размещенных на канале. Отчетный код может представлять собою любой буквенно-числовой ряд.

```
accountcode = spencer145  
amaflags = billing
```

Смешанные опции

Существует еще несколько ключевых слов, у которых нет четкого соответствия предыдущим категориям.

mailbox: Установка этого ключевого слова может означать, что Asterisk предлагал звуковую (и визуальную, если позволят настройки) индикацию ожидания сообщения в случае, если стационарные настройки готовы. Когда вы определите ключевое слово **mailbox** и немое сообщение уже будет существовать в соответствующем **Inbox**, тогда Asterisk произведет запинаящийся тон набора в течение одной секунды после того, как телефон будет готов. На соответствующем оборудовании будет активизирована лампочка ожидания сообщения. Принимает как аргумент номер почтового ящика (который должен быть **voicemail.conf**).

language: Предназначение для международного общения и устанавливает язык общения. Эта опция настроит всю систему сообщений на нужный язык общения. Несмотря на то, что эта опция готова, все же английский язык остается единственным языком, на котором полностью записана инсталляция Asterisk по умолчанию.

stripmsd: Полоса «Самая значимая цифра» является первой цифрой или цифрами всех звонков, связанных с данным стволем каналов. Принимает в качестве аргумента количество цифр с полосы. Подробности в приложении «StripMSD» или `${EXTEN:x}`.

Полный пример файла

Это полный пример файла **zapata.conf**. Он основан на 8 FXO канальным банком 16 FXS T1 .

```
[channels]
;set the FXO's in a group so we can dial out of
;them
;on a first-available basis
group = 1
;set the correct context for our dialout lines
context = pstn
;set the signalling (remember that we signal fxs
;channels
;with fxo, and vice versa)
```

```
signalling = fxs_ls
;set the AMA flags for clarity in the logs
amaflags = documentation
;define the channels that will be covered by the
;previous declarations (in this case all of our
;FXO's)
channel => 1-8
;reset the group, so we don't send outgoing
calls to
;the internal lines
group = 2
;change the context, so we can allow greater
;access to
;services to internal users
context = internal
;set the signalling on the station lines (fxs)
signalling = fxo_ks
;set a mailbox number on the following channels
mailbox = 1234
;set the callerid string (though since we don't
;have a PRI
;it's only seen inside, not on the PSTN.)
callerid = "Dave Schools" <256 555 1234>
;and state the channel this will apply to
channel => 9
;continue and state more channels with mailbox
;indication
;and caller id strings
mailbox = 1235
callerid = "Michael Houser" <256 555 1235>
channel => 10
```

```
mailbox = 1236
callerid = "John Bell" <256 555 1236>
channel => 11
mailbox = 1237
callerid = "Grace Slick" <256 555 1237>
channel => 12
;remember the downward inheritance of options.
;if the next channel doesn't have a voicemail
;box, we need
;to set an empty string, or he'll know whenever
;Grace has a message. Also the callerid should
;be nulled as well
mailbox =
callerid =
;define a bunch of channels with no other
options
channel => 13-22
;Put this phone in a different context, so we
;can give it
; a different initial dialplan...perhaps a lobby
;phone
;with public access
context = lobby
callerid = "Lobby" <5000>
channel => 23
;and turn the callerid off
callerid =
;we can create a 'hotline' phone by placing a
;phone in a special context
;and setting it to answer immediately. In
;extensions.conf we can route
;the phone to an IVR, direct to security, or
;make it call Steak-Out
context => hotline
immediate => yes
channel => 24
```

5.3.2 sip.conf

Предисловие

Файл sip.conf содержит параметры, относящиеся к конфигурации доступа Session Initiation Protocol (SIP) к серверу Asterisk. Клиентов необходимо сконфигурировать в этом файле до того как они смогут размещать или получать звонки при помощи сервера Asterisk.

Устройство

Файл sip.conf читается сверху вниз. В первом разделе находятся общие опции сервера, такие как IP адреса и номер порта, с которым надо связаться. Последующие разделы определяют параметры клиентов, такие как имя пользователя, пароль, IP адрес по умолчанию для нерегистрированных клиентов. Разделы определяются именем в скобках. Первый раздел называется общим (и его нельзя использовать как имя клиента). Последующие разделы начинаются с имени клиента в скобках, а затем идут опции клиента.

Ключевые слова

Ключевые слова даны в sip.conf.

Ключевые слова из общего раздела:

Эти настройки предназначены для [general] раздела sip.conf и отлаживают общие настройки для семейства SIP.

port: Этот порт Asterisk должен прослушивать входящие соединения SIP. По умолчанию это 5060 в соответствии со стандартами. Принимают как аргумент номер порта (который не должен использоваться другой услугой).

bindaddr: Asterisk должен прослушивать IP адрес для входящих SIP соединений. Если у машины есть много реальных или вымышленных IP адресов, то эта опция используется для отбора прослушиваемых IP адресов для Asterisk. По умолчанию все доступные и вымышленные интерфейсы

прослушиваются. В качестве аргумента принимает IP адрес (который д.б. Доступным для интерфейса в системе).

context: Настраивать контекст по умолчанию для всех его последующих клиентов до тех пор, пока он не будет заполнен их определениями единств.

allow: Полностью разрешает кодекс SIP. Помните, что кодеки предпочтительнее размещать в том порядке, в котором происходит их разрешение.

disallow: Категорически запрещает использовать кодек SIP.

tos: Конфигурирует тип услуги (TOS), используемой в передачах SIP и SIP+RTP. Принимаемые значения: lowdelay, throughput, reliability и mincost. Также могут быть использованы цифры в промежутке от 0 до 255.

maxexpirey: Максимально допустимая длина регистрационного запроса в секундах по умолчанию.

defaultexpirey: Длина регистрационного запроса в секундах по умолчанию.

register: Регистрирует Asterisk с другим хостом. Принимает SIP адрес (без sip:), после которого идут слэш («/») и расширение, используемое для контактов.

```
[general]
port = 5060
bindaddr = 192.168.0.1
context = default
disallow = g729
allow = ulaw
allow = gsm
maxexpirey = 180
defaultexpirey = 160
register => 1234@mysipprovider.com/1234
register => 2345@myothersipprovider.com
```

Опции единства:

После общего раздела идет список единств в конфигурации SIP. Единства делятся по трем категориям:

peer: A SIP entity to which Asterisk sends calls (a SIP provider for example)
user: A SIP entity which places calls through this Asterisk (A phone which can place calls only)
friend: An entity which is both a user and a peer. This make sense for most desk handsets and other devices.

type:Типовая опция настраивает класс связи для клиента. Опции-владелец, пользователь и друг.

host:Настраивает IP адрес или придуманное имя хоста данного устройства. Обычно настройка идет на «dynamic» и в этом случае хост идет от любого IP адреса. Это наиболее используемая опция и, обычно, она нужна в сети DHCP.

defaultip:Эту опцию можно использовать, когда ключевое слово хост настроено на dynamic. В этом случае сервер Asterisk будет пытаться направитьзвонки на этот IP адрес, когда звонок принимается для SIP клиента, который еще не зарегистрирован на этом сервере.

username:Эта опция настраивает имя пользователя, к которому сервер Asterisk пытается подсоединиться, когда звонок уже получен. Используется в случае, когда по каким-то причинам значение не совпадает с именем пользователя, по которому зарегистрирован клиент.

canreinvite:Эта опция используется, когда серверу нужно сказать, чтобы он ни под каким предлогом не приглашал повторно клиента. Это нужно для взаимодействием с некоторым «железом», которое ломается при повторном приглашении, например, Cisco ATA186.

context:Когда наступает ясность в определении клиента, тогда это ключевое слово настраивает контекст по умолчанию на «только этот клиент».

dtmfmode:Отбирает какие DTMF цифры следует отправить в группу, а какие-из группы. Существующие значения:

inband: DTMF is send as audio in-band, and is detected in-band.
rfc2833: DTMF is sent out-of-band using RFC2833 (default)
info: DTMF is sent and received out of band using INFO messages
(very rarely used)

mailbox:Один или более почтовых ящиков могут быть перечислены (отделяются запятыми) для отправки сообщений Message Waiting Indicator (индикатор ожидания сообщения) (MWT) к данному SIP владельцу.

qualify: Максимальное количество времени в миллисекундах для владельца на ответ. Это заставляет Asterisk периодически подсчитывать время и решать, превышен ли лимит на ответ или нет.

secret:Секрет, известный владельцам и пользователям для аутентификации регистрации на право звонить.

nat:Заставляет Asterisk рассматривать владельца или пользователя как потенциальный сетевой адрес, переведенный в хост. Это полезно, когда владельцы находятся позади firewalls.



● Помните, что активация функции nat побуждает Asterisk заставлять определенные RFC пути иметь дело с контактом: и порциями звонков SDP, чтобы сработаться с хостами NAT. Во время написания этого документа nat=yes было невыполнимо с телефонами Pingtel.

Полный пример файла SIP:

В данной рамке содержится полный пример работоспособного файла sip.conf.

```
[general]
port=5060
bindaddr=192.168.0.10
context=default
register => 1234@mysipprovider.com
[snom]
type=friend
secret=snom100
host=dynamic
defaultip=192.168.0.15
mailbox=2345,1234
[cisco]
type=friend
secret=mysecret
host=192.168.0.20
canreinvite=no
mailbox=1234
context=trusted
```

5.3.3 iax.conf

Предисловие

Этот файл предназначен для конфигурации клиентов, получающих вызовы через протокол Inter-Asterisk eXchange. В основном, IAX используется для пропуска звонков между серверами Asterisk. Часто серверы Multiple Asterisk конфигурируются для того, чтобы взаимодействовать друг с другом, используя этот файл. Файл iax.conf есть как в версии 1 IAX, так и в версии 2.

Устройство

Файл iax.conf начинается с общего раздела, который устанавливает общие серверные опции. Внутри общего раздела мы также можем сконфигурировать

сервер Asterisk, чтобы зарегистрировать его как клиента к удаленному серверу, для доступа к плану набора номера другой системы Asterisk.

После общего раздела идут определения клиентов- один на раздел. Разделы отделяются друг от друга их именами в скобках.

Ключевые слова

Данные ключевые слова, используются в `iax.conf`.

В общем разделе:

port: Это порт для прослушивания входящих соединений. По умолчанию это порт 5036. Принимает в качестве аргумента номер порта (который не должен использоваться другой услугой).

bindaddr: Если множественные IP адреса в одной системе, то эту опцию можно настроить для привязки Asterisk к единому интерфейсу.

```
port = 5036
bindaddr = 0.0.0.0
```

amaflags: Настраивает флаги АМА, влияющие на разбивку единств по категориям в подробных записях о звонках. Это ключевое слово можно настроить на каждого клиента внутри его определения. Принимает следующие значения:

```
billing: Mark the entry for billing
documentation: Mark the entry for documentation.
omit: Do not record calls.
default: Use the system default.
```

accountcode: Настраивает отчетный код на iog IAX звонки по умолчанию. Это ключевое слово можно также использовать в определении клиента, чтобы настроить отчетный код для этого клиента.

```
accountcode = wmeadows  
amaflags = documentation
```

bandwidth: Эта опция нужна для контроля за общим использованием кодеков. Она не только разрешает или запрещает определенные кодеки, но ее также можно настроить на «низкий», чтобы автоматически избегать кодеков, которые плохо работают в небольшой по ширине группе. Принимает «низкий» или «высокий».

allow: Специально разрешает использовать определенный кодек. Принимает «codec» или «all». Использование «all»- это то же самое, что **bandwidth=high**.

disallow: Специально запрещает определенный кодек (см. **allow**)

```
bandwidth=low  
disallow=all  
allow=gsm
```

jitterbuffer: Включает или выключает буфер jitter (дрожжания). Он используется для доведения качества звука до максимума путем балансировки латентности и количества выпавших пакетов. Существует несколько ключевых слов для улучшения звучания буфера дрожжания.

dropcount: Настраивает максимальное количество пакетов, чтобы «уронить» их для уменьшения латентности, на размер памяти.

maxjitterbuffer: Настраивает максимальный размер буфера.

maxexcessjitterbuffer: Настраивает максимальный излишек буфера jitter, что приводит к медленному понижению, а это, в свою очередь, улучшает латентность.

register: Регистратор нужен для того, чтобы приказать серверу Asterisk зарегистрироваться с другим сервером Asterisk. Обычно это используется, если

у нашего локального сервера есть динамичный IP адрес и требуется сказать другому серверу, где его найти. Формат утверждения регистратора таков

```
register => username:secret@server
```

Поле «секрет» является дополнительным, если до этого никакого секрета не было установлено на сервере, с которым происходит соединение. Если используется RSA encryption, определитесь с ключом, котрый вы отправите на сервер в таком формате:

```
register => username:[key]@server
```

tos:Определяет тип сервисных битов, которые надо установить на пакетах IAX, которые в свою очередь могут улучшить путь пакетов. Доступные значения:

```
lowdelay: minimize delay  
throughput: maximize throughput  
reliability: maximize reliability  
mincost: use the lowest cost path  
none: use no routing flags
```

```
tos = lowdelay
```

Опции для единств

Определения единств начинаются с имени единств в скобках. После имени идут несколько пар ключевых сло/значений, применимых в единстве, в котором они настроены.

Для пользователей есть несколько ключевых слов:

type:Эта опция устанавливает тип единства для клиента. Возможные типы:

user: A user can place calls to or through the Asterisk server.
peer: A peer receives calls from the Asterisk server, but does not place them
friend: A friend both sends and receives calls through the Asterisk server. This makes the most sense for handsets or other station devices. When in doubt use this type.

context: При использовании внутри определения клиента, это ключевое слово превращает настройку входящего контекста по умолчанию в общий раздел только для пользователя.

callerid: Настраивает линию ID звонящего на данное единство. Эту линию будут использовать внутри и отправлять в PSTN, если линия PRI будет занята отправкой звонка во внешний мир. Если эту опцию оставить пустой, то отправленный единством Caller ID будет использоваться вместо него.

```
callerid => .Judy. <256 555-1234>
```

auth: Настраивает тип аутентификации. IAX поддерживает три метода аутентификации. Первый (наименее безопасный) – plaintext. Пароли (или секреты) направляются в простом тексте через сеть. Второй- md5, он использует алгоритм вызова ответа md5, он использует алгоритм вызова ответа md5. Обе машины имеют доступ к простому тексту с паролем, но их будут подтверждать, используя hash md5, при прохождении через сеть. Наиболее безопасным является третий метод – использование общего/личного encryption ключа RSA для передачи и хранения секрета. Пары общей/личный ключ можно создать, используя включенный в программу astgenkey. Общий ключ необходимо вручную передать на сервер и сохранить в /var/lib/asterisk/keys/name.pub. Личные серверные ключи находятся там же, где и name.key.



● Важное примечание: Чтобы использовать ключи RSA с Asterisk, вам придется «init keys» в консоль во время запуска. Asterisk будет подсказывать вам это действие каждый раз, когда он стартует.

inkeys: Общие ключи используются для decrypt аутентификации для входящего клиентского запроса или регистрации.

outkey: Личный ключ используется для encrypt исходящей аутентичной коммуникации для этого клиента.

```
auth=md5  
secret=password
```

```
auth=rsa  
inkeys=theirkey  
outkey=mykey
```

permit: Хосты для разрешения подсоединения к этому пользователю. Это может быть как единичный хост, так и пара хост/сетевая маска (netmask).

deny: Хосты для запрещения любой попытки входящего соединения к этому пользователю. Deny принимает такой же формат аргумента, как и permit .

```
deny = 0.0.0.0/0.0.0.0  
permit=192.168.0.1/255.255.255.0  
permit=216.207.245.45
```

host: Настраивает предполагаемый исходящий хост для этого клиента. Можно настроить на ip адрес или dynamic, что позволит принять входящие соединения от любого хоста (если он не запрещен).

defaultip: Это IP адрес по умолчанию для клиента IAX. С этим полем происходит консультация, если Asterisk получает звонок для клиента IAX, который является динамичным, и он не был зарегистрирован, что не позволяет Asterisk узнать его текущий IP адрес. Принимает в качестве аргумента IP адрес.

```
host=dynamic  
defaultip=192.168.0.1
```

accountcode: При использовании внутри определения клиента, настраивает код отчета только для этого клиента. Это используется услугой логин звонка.

qualify: Говорит Asterisk, следует ли проводить тестирование на предмет жизнеспособности владельца до попытки соединения со звонком. Если опция говорит «да», то Asterisk периодически будет контактировать с владельцем до того, как представить какую-либо информацию о звонке. Специальным аргументом является максимальное количество миллисекунд, которые владелец может использовать на ответ до того момента, когда он будет считаться «недоступным».

```
qualify=1000
```

mailbox: Обеспечивает принадлежность почтового ящика к данному владельцу таким образом, что когда он регистрирует, то можно определить любую принадлежность ожидающего сообщения.

```
mailbox=1234  
mailbox=1002,1003
```

trunk: Разрешает или запрещает главную линию для данного пользователя или владельца. Режим trunk является более эффективным методом работы IAX, если по линии обычно проходит много звонков. Режим trunk требует наличия интерфейса Zaptel на сервере Asterisk.

```
trunk=yes
```

Полный пример файла

```
[general]
;set up some general items
port=5036
accountcode=iaxcalls
amaflags=default
bandwidth=low
allow=gsm
disallow=lpc10
jitterbuffer=yes
dropcount=3
maxjitterbuffer=500
maxexcessjitterbuffer=100
register =>
asterisk1:opensecret@telco.digium.com
context=iax
;from here on it.s client definitions
[trustedhost]
host=192.168.0.50
trunk=yes
context=trusted
[authhost]
secret=foobar
host=dynamic
defaultip=68.62.178.239
[rsahost]
auth=rsa
inkeys=rsapublickey
host=dynamic
defaultip=216.207.245.55
accountcode=log1234
callerid=.Mark Spencer. <256 428 6000>
```

5.4 Конфигурации приложения

В этом разделе подробно описывается синтаксис файлов конфигурации для разных приложений Asterisk.

5.4.1 voicemail.conf

Краткий обзор

Файл voicemail.conf конфигурирует многие системные параметры для системы голосовой почты и хранит информацию с почтовых ящиков, включая номер почтового ящика для маппинга пропускного кода, имена владельцев ящиков, e-mail адреса полученных сообщений.

Устройство

Файл voicemail.conf состоит из двух разделов. Первый раздел -общий- содержит различные системные параметры, такие как форматы, в которых сообщения будут храниться, e-mail адрес из системы голосовой почты, по которому можно определить принадлежность. Второй раздел- по умолчанию- содержит конфигурации для индивидуальных ящиков голосовой почты.

Ключевые слова

В общем разделе есть следующие ключевые слова и опции:

format: Формат настраивает форматы файлов для хранения голосовой почты . Если множественные форматы определены, то все форматы будут записаны, а самый доступный из них будет использован для отмотки назад. Последний из перечисленных форматов используется для голосовой почты по e-mail, если такие опции активизированы. Доступные форматы:

gsm : use raw gsm encoding. Best for VoIP.
wav: MS wav format, 16 bit linear
WAV: MS wav format, gsm encoded

g723sf: [G.723.1](#) simple frame (note that Asterisk cannot directly encode , due to licensing issues. It can, however, store and transmit file received from an external source, i.e. from a SIP phone with a built in codec).

```
format=gsm|wav|WAV
```

В данном примере каждая полученная голосовая почта будет записана в gsm, MS-GSM и в линейных форматах wav.

```
format=gsm
```

Данный пример будет хранить голосовую почту только в необработанном формате gsm.

serveremail: Эта опция настраивает e-mail адрес, по которому ожидающаяся голосовая почта по e-mail будет определяться. Это значение будет использоваться в поле e-mail «From». Доступными опциями является любой буквенно-цифровой ряд или буква/цифра.

Например:

```
serveremail=asterisk
```

В этом примере поле «From» будет настроено на «Asterisk». В большинстве случаев внешний почтовый сервер будет дополнять локальное имя хоста.

```
serveremail=asterisk@myhost.com
```

Этот пример будет настраивать e-mail. Обычно он не переписывается заново внешним почтовым сервером. Это целесообразно делать, если вы хотите,

чтобы e-mail появлялся, выходя из другого имени хоста, а не из хоста на локальной машине.

append: Эта опция определяет нужно ли добавить голосовой файл голосовой почты к определению e-mail. Принимает, как аргумент, «да» или «нет».

append=yes

append=no

Раздел по умолчанию принимает в качестве ключевого слова номер почтового ящика. Ключевое слово принимает в качестве параметров пропускной код, имя владельца и e-mail адрес владельца, на который отправят уведомление об ожидающем сообщении.

1234 => 4321,John Doe,jdoe@misc.com



● **Важное примечание:** Имя владельца используется приложением «Directory», чтобы найти расширения, основанные на именах, которые представлены звонящим.

maxmesssage: Устанавливает максимальную длину для сообщения голосовой почты в секундах. Эта опция удерживает людей от слишком длинных сообщений.

maxgreet: Устанавливает максимальную длину в секундах на приветствия, которые пользователь записывает на случай, если он занят, недоступен или для названий сообщений.

Полный пример файла

Это полный пример рабочего файла voicemail.conf file.

```
[general]
format=gsm|wav
serveremail=asterisk@mymachine.com
append=yes
maxgreet=30
maxmessage=90
[default]
1234 => 4321,John Doe,jdoe@mycompany.com
```