

4

Исходная конфигурация Asterisk

Я не всегда понимаю, о чем говорю, но всегда уверен, что я прав.

– Мухаммед Али

После выполнения всех этапов, описанных в главе 3, должна быть получена рабочая система Asterisk. Если что-то не так, не пожалейте времени, вернитесь назад и еще раз просмотрите все шаги, проконсультируйтесь в Википедии, привлечите сообщество и заставьте свою систему работать.

К сожалению, пока мы не можем никуда позвонить, потому что еще не создан ни один канал. Чтобы оторвать этот самолет от земли, необходима взлетная полоса. Существуют десятки разных типов каналов и различных способов сконфигурировать каждый из них, но мы просто хотим получить возможность позвонить, поэтому давайте сделаем это и не будем пока вдаваться в тонкости. Мы решили привести здесь рекомендации по конфигурации четырех каналов: Foreign eXchange Office (FXO), Foreign eXchange Station (FXS), Session Initiation Protocol (SIP) и Inter-Asterisk eXchange (IAX)¹. Были выбраны именно эти типы, потому что, бесспорно, они самые популярные из каналов, используемых в малых системах Asterisk, а одной из принципов данной книги является разумная простота. Рассматривая эти каналы, мы не стремимся к тому, чтобы дать исчерпывающий и всесторонний обзор всех типов

¹ Официально текущей версией является IAX2, но, поскольку от поддержки IAX1 отказались много лет назад, под IAX и IAX2 подразумевается одна и та же версия.

линий или топологий, а просто представляем основную информацию, необходимую для разработки телекоммуникационной системы. Подробно конфигурация сценариев и каналов рассматривается в приложении D.

Начнем с изучения базовой конфигурации аналоговых интерфейсов, таких как порты FXS и FXO, на примере платы Digium TDM11B (которая является аналоговой платой с одним портом FXS и одним портом FXO)¹.

Далее мы коснемся нескольких интерфейсов для передачи голоса по IP-протоколу (Voice over Internet Protocol, VoIP): рассмотрим локальный канал SIP и IAX2, подключенный к программному телефону или телефонному аппарату, а также соединение двух серверов Asterisk по этим двум популярным протоколам.

Для SIP будут рассмотрены телефонные аппараты Linksys, Polycom, Aastra, Grandstream и Cisco. Приносим извинения, если в этот список не вошла используемая вами модель телефона. Однако важно то, что при всем многообразии предлагаемых этими аппаратами настраиваемых параметров обычно, чтобы устройство заработало, необходимо задать только несколько из них. Это и будет нашей целью, потому что мы считаем, что для первого раза достаточно просто привести устройство в рабочее состояние, а не выполнять полную идеальную настройку. Здесь не будут обсуждаться все функции, которые, возможно, вам хотелось бы использовать (такие, как идентификация вызывающего абонента (Caller ID) или расширенные настройки кодеков и политики безопасности), но с вашего телефона можно будет звонить и принимать звонки, а это должно заставить вас улыбнуться – хорошее состояние духа не помешает, когда мы углубимся в дебри.

Проработав эту главу, вы получите базовую систему, состоящую из нескольких полезных интерфейсов. Они обеспечат основу, необходимую для изучения файла `extensions.conf` (подробно обсуждаемого в главе 5), в котором хранится диалплан (фактически этот файл содержит инструкции, необходимые Asterisk для построения диалплана). Не имея аналогового оборудования, некоторые примеры будет невозможно реализовать, но все равно у вас будет сконфигурированная система, подходящая для среды, поддерживающей исключительно VoIP.

Что мне на самом деле нужно

Символ звездочки (*) используется как символ подстановки во многих приложениях. Это хорошее имя для данной офисной АТС по многим

¹ Эта конфигурация была известна как комплект Digium Dev-lite. Далее будет подробнее рассмотрено, чем различаются FXS и FXO. Попросту говоря, эта плата предоставит один порт для подключения к традиционной аналоговой линии телефонной компании (FXO) и один порт для подключения к аналоговому телефону (FXS). Это может быть телефон любого типа, который будет работать с традиционной домашней телефонной сетью.

причинам, одна из которых – огромное число типов интерфейсов, с которыми может соединяться Asterisk. К ним относятся:

- Аналоговые интерфейсы, такие как телефонная линия и аналоговые телефоны.
- Цифровые линии, такие как линии T1 и E1.
- Протоколы VoIP, такие как SIP и IAX¹.

Asterisk не требует никакого специализированного оборудования – даже звуковой карты, – хотя естественно ожидать, что система телефонной связи должна физически соединяться с телефонной сетью. Существует множество типов канальных плат, которые обеспечивают возможность соединения Asterisk с такими устройствами, как аналоговые телефоны или PSTN-сети, но они не являются обязательным условием функционирования Asterisk. На стороне пользователя (или станции) системы может использоваться любой программный телефон Windows, Linux и других операционных систем или практически любой физический IP-телефон. То есть вопрос с телефонами системы решен. Что касается линии связи, если нет прямого подключения к центральной АТС, можно передавать звонки по Интернету с помощью провайдера сервиса VoIP.

Работа с конфигурационными файлами интерфейсов

В данной главе будет построена конфигурация Asterisk на платформе, установленной в предыдущей главе. Для первых нескольких разделов, в которых рассматриваются каналы FXO и FXS, предполагается использование комплекта TDM11B производства Digium (который поставляется с одним FXO- и одним FXS-интерфейсом). Он обеспечит возможность подключения к аналоговой сети (FXO) и аналоговому телефону (FXS). Обратите внимание, что этот аппаратный интерфейс не является обязательным; если вы хотите настроить конфигурацию для общения только по IP-протоколу, можете пропустить раздел, посвященный конфигурации SIP.

Сама по себе выполняемая здесь конфигурация не будет иметь особенного практического значения, но она станет основой для дальнейшей работы. Будут рассмотрены следующие файлы:

zaptel.conf

Здесь будет выполнена низкоуровневая конфигурация аппаратного интерфейса. Будут настроены один канал FXO и один канал FXS. Это сконфигурирует драйвер для ядра Linux.

¹ ...и H.323, и SCCP, и MGCP, и UNISTIM.

zapata.conf

В данном файле будет выполнена настройка взаимодействия Asterisk с оборудованием. Этот файл обеспечивает конфигурацию оборудования, используемого в пользовательском процессе Asterisk, на более высоком уровне.

extensions.conf

Будут созданы достаточно примитивные диалпланы, но они подтверждают работоспособность системы.

sip.conf

Здесь будет сконфигурирован SIP-протокол.

iax.conf

Здесь будет выполнена конфигурация входных и выходных IAX-каналов.

Следующие разделы посвящены редактированию нескольких конфигурационных файлов. Чтобы внесенные изменения возымели действие, придется перезагрузить эти файлы. После редактирования файла *zaptel.conf* понадобится выполнить перезагрузку конфигурации оборудования, используя строку `/sbin/ztcfg -vv` (`-vv` можно опустить, если не нужен развернутый вывод). Изменение файла *zapata.conf* потребует выполнения команды `module reload` из консоли Asterisk; однако изменение методов обмена сигналами требует перезагрузки системы (команда `restart`). После редактирования файлов *iax.conf* и *sip.conf* необходимо выполнить команды `iax2 reload` и `sip reload` соответственно.

Для тестирования вновь определенных устройств необходим диалплан, посредством которого могут устанавливаться соединения. Хотя диалплан Asterisk еще не рассматривался (этим мы займемся в следующей главе), для тестирования работы, выполняемой в данной главе, необходимо создать базовый файл *extensions.conf*.

Создайте резервную копию файла-шаблона *extensions.conf* (попробуйте команду `bash mv extensions.conf extensions.conf.sample`), затем создайте пустой файл *extensions.conf* (используя команду `bash touch extensions.conf`) и вставьте в него следующие строки:

```
[globals]

[general]
autofallthrough=yes

[default]

[incoming_calls]

[internal]

[phones]
include => internal
```



В разделе [general] задано `autofallthrough=yes`, что указывает Asterisk продолжать выполнение, если все действия для добавочного номера исчерпаны. Если задать этому параметру значение `no`, Asterisk после выполнения всех предусмотренных приоритетов будет ожидать ввода. Это значение является предпочтительным, если последним приложением, выполняемым для добавочного номера, является приложение `Background()`. Если задано значение `yes` (которое стало значением по умолчанию в версии 1.4), Asterisk прервет вызов после завершения выполнения `Background()` (после воспроизведения имеющихся звуковых файлов). Чтобы заставить Asterisk ожидать ввода номера после того, как приложение завершит воспроизведение предоставляемых ему голосовых сообщений, использует приложение `WaitExten()`.

Не пугайтесь, если вышесказанное не имеет сейчас для вас особого смысла. Просто мы еще не рассматривали диалплан, приложения, приоритеты и добавочные номера (все это ожидает нас в следующей главе). Поэтому пока что просто задайте `autofallthrough=yes`. Безопаснее использовать команду `autofallthrough=yes`, поскольку мы не хотим, чтобы Asterisk простаивала без дела в ожидании ввода номера, если это не указано ей явно.

Пока что это все, но данный файл будет использоваться по ходу этой главы для построения тестового диалплана, который поможет убедиться в работоспособности всех наших устройств. Также не забудьте выполнить команду `dialplan reload` из командной строки Asterisk, чтобы внесенные изменения вступили в силу. Проверьте свои изменения, введя в командной строке команду `dialplan show`:

```
*CLI> dialplan show
[ Context 'phones', created by 'pbx_config' ]
  Include =>      'internal'                      [pbx_config]

[ Context 'internal', created by 'pbx_config' ]

[ Context 'incoming_calls', created by 'pbx_config' ]

[ Context 'default', created by 'pbx_config' ]

[ Context 'parkedcalls', created by 'res_features' ]
  '700' => 1.    Park((null))                      [res_features]

-- 1 extension (1 priority) in 5 contexts. --
```



Контекст `parkedcalls` – это внутренний контекст Asterisk, заданный в файле `features.conf`.

Настройка диалплана для выполнения тестовых вызовов

Теперь давайте подробнее остановимся на тестовом диалплане, о котором мы начали разговор в предыдущем разделе, – он позволит выполнять обратный вызов программного телефона, после того как тот будет настроен, и использовать приложение диалплана `Echo()` для тестирования двусторонней аудиосвязи. Более подробно диалпланы рассматриваются в главе 5, а пока что просто добавим выделенные курсивом строки в существующий файл `extensions.conf`. Мы будем использовать этот диалплан по ходу данной главы и дополнять его в определенных разделах. После ввода текста в `extensions.conf` диалплан необходимо перезагрузить, выполнив команду `dialplan reload` из консоли Asterisk:

```
[globals]

[general]

[default]
exten => s,1,Verbose(1|Unrouted call handler)
exten => s,n,Answer()
exten => s,n,Wait(1)
exten => s,n,Playback(tt-weasels)
exten => s,n,Hangup()

[incoming_calls]

[internal]
exten => 500,1,Verbose(1|Echo test application)
exten => 500,n,Echo()
exten => 500,n,Hangup()

[phones]
include => internal
```

Каналы FXO и FXS

Каналы FXO и FXS отличаются друг от друга лишь тем, что один из них обеспечивает тональный сигнал готовности линии. FXO-порт не генерирует тонального сигнала, он его принимает. Самый простой пример – тональный сигнал, поставляемый телефонной компанией. FXS-порт обеспечивает и тональный сигнал, и напряжение сигнала вызова (звонка), предупреждающего пользователя о входящем вызове. Оба интерфейса обеспечивают двустороннюю связь (то есть передачу и прием в обоих направлениях одновременно)¹.

¹ Двустороннюю связь иногда называют также полнодуплексной. Полудуплексная связь – это связь, которая осуществляется одновременно только в одном направлении.

Если у вашего сервера Asterisk есть совместимый FXO-порт, в него можно подключить телефонную линию телефонной компании (или telco), что позволит Asterisk использовать эту телефонную сеть для отправки и приема телефонных звонков. Кроме того, если ваш сервер Asterisk имеет совместимый FXS-порт, в него можно подключить аналоговый телефон. Таким образом, Asterisk сможет направлять поступающие вызовы в телефон и вы будете способны использовать этот телефон для звонков куда-либо.

В конфигурации порты определяются протоколом обмена сигналами, который они используют, а не их физической сущностью. Например, физический FXO-порт будет определен в конфигурации протоколом обмена сигналами FXS, а FXS-порт – протоколом FXO. Это может сбивать с толку до тех пор, пока не станут ясны причины такого явления. Платы FX_ названы так не исходя из их физической сути, а соответственно тому, какие устройства подключаются к ним. Следовательно, плата FXS – это плата, подключаемая к станции (station). Таким образом, можно заметить, что, для того чтобы справиться со своими задачами, плата FXS должна вести себя как центральная АТС и использовать протокол обмена сигналами FXO. Аналогично, плата FXO подключается к центральной АТС. Это означает, что она должна будет вести себя как станция и использовать протокол обмена сигналами FXS. Модем в компьютере – классический пример устройства FXO.



Более старая плата X100P производства компании Digium использовала набор микросхем Motorola, а плата X101P (которую Digium продавала до того, как полностью переключилась на TDM400P) базируется на наборе микросхем Ambient/Intel MD3200. Эти платы являются модемами с драйверами, которые можно использовать в качестве отдельного устройства FXO (интерфейс для телефонии не может использоваться как FXS-порт). От поддержки плат X101P отказались в пользу плат TDM-серий.

Эти платы (или их клоны) НЕЛЬЗЯ использовать в средах производственной эксплуатации. Ведь не просто так они стоят на eBay всего \$10.

Платы X100P/X101P не годятся для производственной эксплуатации, потому что часто являются причиной возникновения эха и не дают возможности удаленного управления разведением. Будьте благоразумны и не тратьте время на это оборудование. Если обратиться с вопросом об этих платах к сообществу, многие ответы будут недружелюбными. Мы вас предупредили.

Как найти порты FXO и FXS на плате TDM400P

На рис. 4.1 представлена плата TDM400P с модулями FXS и FXO. Фото черно-белое, поэтому невозможно различить цвета, но под номером 1 –

FXS-модуль зеленого цвета, а под номером 2 – FXO-модуль, оранжево-красный. В нижнем правом углу рисунка можно увидеть разъем Molex, посредством которого плата подключается к блоку питания компьютера.



Подключение FXS-порта (зеленый модуль) к PSTN может привести к выходу из строя модуля и платы из-за подачи напряжения в систему, которая предназначена для его производства, а не потребления!

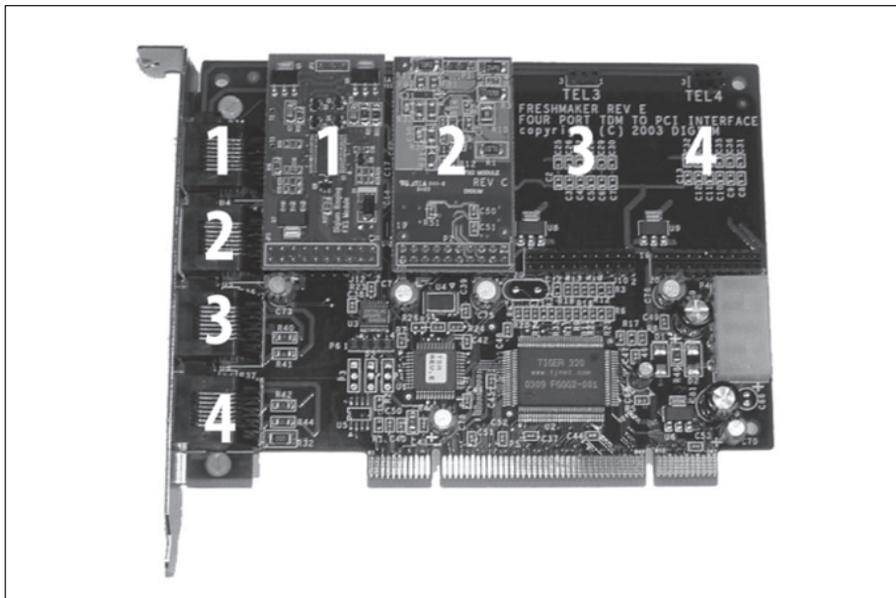


Рис. 4.1. Плата TDM400P с модулем FXS (1 по горизонтали) и модулем FXO (2 по горизонтали)



Если на плате TDM400P есть модули FXS, убедитесь, что вы не забыли подключить питание к разъему Molex, который используется для обеспечения напряжения, необходимого для возбуждения генератора звонка в FXS-портах. Разъем Molex не нужен, если имеются только FXO-модули.

Конфигурация канала FXO для соединения с PSTN

Начнем с конфигурации FXO-канала. Сначала сконфигурируем оборудование Zaptel, а затем – устройства Zapata. Зададим очень простой диалплан и покажем, как тестировать канал.

Конфигурация оборудования Zaptel

Для конфигурации оборудования используется файл `zaptel.conf`, который находится в папке `/etc/`. Приведенная минимальная конфигурация определяет FXO-порт, использующий протокол обмена сигналами FXS:

```
fxsks=2
loadzone=us
defaultzone=us
```

В первой строке, кроме указания используемого протокола обмена сигналами (FXO или FXS), для канала 2 задается один из следующих протоколов:

- Loop start (ls)
- Ground start (gs)
- Kewlstart (ks)

Разница между loop start (кольцевая сигнализация) и ground start (сигнализация с заземлением) в том, как оборудование запрашивает тональный сигнал: линия, использующая сигнализацию с заземлением, подает сигнал дальнему концу о том, что хочет получить тональный сигнал, путем мгновенного заземления одного из проводов; линия с сигнализацией по шлейфу для запроса тонального сигнала использует замыкание. Хотя в новых системах их уже практически нет, аналоговые линии с сигнализацией с заземлением по-прежнему применяются во многих областях¹. Сигнализация с заземлением на самом деле довольно странная штука, потому что в Asterisk нет ее аналоговой формы. Фактически передается не напряжение заземления, а сигнальный бит, предусмотренный для аналоговых сетей, присутствующих на стороне T1. Если вы чего-то здесь не понимаете, не отчаивайтесь; скорее всего, вам никогда не придется использовать сигнализацию с заземлением. Сейчас почти все линии связи (и аналоговые телефоны/модемы/факсы) используют сигнализацию по шлейфу. Kewlstart, в сущности, — это то же самое, что и кольцевая сигнализация, за исключением того что он обладает более развернутой логикой и, таким образом, может определять отсоединение удаленного конца². Kewlstart является пред-

¹ Да, в многоканальных линиях T1 существует такая вещь, как сигнализация с заземлением, но она не имеет ничего общего с фактическим заземлением линии (полностью цифровой).

² Отсоединение удаленного конца имеет место, когда на дальнем конце линии кладут трубку. В неконтролируемой сети нет средства оповещения ближнего конца о том, что вызов завершен. Если используется телефон, это не представляет проблемы, поскольку абонент поймет, что звонок завершен, и сам повесит трубку. Однако система голосовой почты, записывающая сообщение, не имеет никакой возможности определить, что дальний конец прервал соединение, и, таким образом, будет продолжать записывать тишину, или даже тональный сигнал, или сигнал занятости линии. Kewlstart может определять такие условия и отсоединять линию.

почтительным протоколом обмена сигналами для аналоговых линий в Asterisk.

Чтобы определить другой метод обмена сигналами, не `kewlstart`, замените `ks` в выражении для `fxsks` на `ls` или `gs` (для кольцевой сигнализации или сигнализации с заземлением соответственно).

Параметр загрузки зоны `loadzone` задает набор сигналов (которые указаны в файле `zonedata.c`), используемых для канала. Файл `zonedata.c` содержит информацию о всевозможных звуках, воспроизводимых телефонной системой в определенной стране: тональный сигнал, сигналы дозвона, сигнал «занято» и т. д. Если применить загруженную тоновую зону к `Zap`-каналу, этот канал будет воспроизводить сигналы заданной страны. Для разных каналов можно задать различные наборы сигналов. Если для канала зона не задана, используется параметр зоны по умолчанию `defaultzone`.

После конфигурации `zaptel.conf` можно загрузить драйверы для платы. Команда `modprobe` применяется для загрузки модулей, используемых ядром Linux. Например, чтобы загрузить драйвер `wctdm`, нужно выполнить команду

```
# modprobe wctdm
```

Если при загрузке драйверов не выводится никаких сообщений, значит, драйверы загружены успешно¹. Убедиться в том, что оборудование и порты были загружены и сконфигурированы правильно, можно с помощью программы `ztcfg`:

```
# /sbin/ztcfg -vv
```

На экран будут выведены сконфигурированные каналы и используемый метод обмена сигналами. Например, для `TDM400P` с одним модулем `FXO` будет получен такой вывод:

```
Zaptel Configuration
=====
Channel map:
Channel 02: FXS Kewlstart (Default) (Slaves: 02)
1 channels configured.
```

Если получено сообщение о следующей ошибке, для канала неправильно сконфигурирован используемый метод обмена сигналами (или отсутствует оборудование по этому адресу):

```
ZT_CHANCONFIG failed on channel 2: Invalid argument (22)
Did you forget that FXS interfaces are configured with FXO signaling and that
FXO interfaces use FXS signaling?
```

(ZT_CHANCONFIG дала сбой для канала 2: Недействительный аргумент (22).

¹ Обычно в случае успешной загрузки модулей можно не просматривать отладочную информацию, но если вы хотите увидеть ее, проверьте вывод консоли (по умолчанию он находится на ТТУ-терминале номер 9, но это можно изменить в сценарии `safe_asterisk` – подробнее см. в предыдущей главе).

Вы забыли, что FXS-интерфейсы конфигурируются на использование протокола обмена сигналами FXO и FXO-интерфейсы используют протокол FXS?)

Чтобы выгрузить драйверы из памяти, используйте команду удаления модуля `rmmod`:

```
# rmmod wctdm
```

Программа `zttool` является инструментом диагностики, используемым для определения статуса оборудования. Выполнив ее, мы получаем список всех установленных аппаратных средств. В этом списке можно выбирать то или иное устройство и просматривать информацию о его текущем статусе. Статус `OK` означает, что устройство загружено успешно:

```
Alarms   Span
OK       Wildcard TDM400P REV E/F Board 1
```

Конфигурация оборудования Zapata

Asterisk использует файл `zapata.conf` для определения настроек и конфигурации оборудования для телефонии, установленного в системе. Файл `zapata.conf` также управляет различными возможностями и функциями аппаратных каналов, такими как Caller ID, отложенный вызов, эхоподавление и многие другие.

Во время конфигурации `zaptel.conf` и загрузки модулей Asterisk не имеет ни малейшего представления о том, что будет конфигурироваться. Необязательно, чтобы оборудование использовалось Asterisk; его с таким же успехом могло бы применять другое ПО, взаимодействующее с модулями Zapata. С помощью `zapata.conf` мы сообщаем Asterisk об аппаратных средствах и управляем соответствующими функциями:

```
[trunkgroups]
; описание всех магистральных групп

[channels]
; аппаратные каналы
; значения по умолчанию
usecallerid=yes
hidecallerid=no
callwaiting=no
threewaycalling=yes
transfer=yes
echocancel=yes
echotraining=yes

; описание каналов
context=incoming ; Используется контекст [incoming], описанный в extensions.conf
signaling=fxs_ks ; Для канала FXO используется протокол обмена сигналами FXS
channel => 2 ; PSTN подключается к порту 2
```

Раздел `[trunkgroups]` описывает соединения, в которых множество физических линий используются как одно логическое соединение с теле-

фонной сетью (они не обсуждаются в данной книге). Если вам необходим этот тип функциональности, в вашем распоряжении есть файл `zadata.conf.sample` и любимый поисковик для получения дополнительной информации.

Раздел `[channels]` определяет метод обмена сигналами для аппаратных каналов и их параметры. Если параметр задан, он наследуется далее по всему файлу. Для определения канала используется синтаксис `channel =>`. Каждое описание канала наследует все параметры, определенные в файле выше. Если для разных каналов требуется задать различные параметры, они должны задаваться перед описанием `channel =>`.

Строка `usecallerid=yes` служит для активации возможности Caller ID, а строка `hidecallerid=no` определяет, что идентификатор вызывающего абонента не будет скрыт для исходящих вызовов. Отложенный вызов деактивирован для FXO-линии посредством строки `callwaiting=no`. Активация вызова с подключением третьего абонента (строка `threewaycalling=yes`) позволяет переводить активный вызов в состояние ожидания путем быстрого нажатия рычажного переключателя (обсуждается в главе 7). При этом текущий вызов будет отложен. После этого можно позвонить третьей стороне и подключить ее к разговору, еще раз нажав переключатель. По умолчанию возможность вызова с подключением третьего абонента деактивирована.

Разрешение переадресации вызова с помощью рычажного переключателя настраивается строкой `transfer=yes`; для этого требуется, чтобы была активирована возможность вызова с подключением третьего абонента. Для устранения эха, которое может возникнуть в аналоговых линиях, используется эхокомпенсатор Asterisk, он активируется строкой `echocancel=yes`. Эхокомпенсатору в Asterisk требуется некоторое время на изучение эха, но этот процесс можно ускорить, включив обучение эхоподавления (`echotraining=yes`). В этом режиме Asterisk посылает тональный сигнал в линию в начале вызова, чтобы измерить эхо, и таким образом, на его изучение уходит меньше времени.

При поступлении вызова на интерфейс FXO вы захотите выполнять некоторые действия. Действия, которые будут осуществляться, задаются в блоке инструкций под названием `context`. Входящие вызовы FXO-интерфейса направляются в контекст `incoming` (входящие) посредством строки `context=incoming`. Команды, выполняемые в этом контексте, определены в файле `extensions.conf`.

Наконец, поскольку канал FXO использует протокол обмена сигналами FXS, это определено в строке `signaling=fxs_ks`.

Конфигурация диалплана

Приведенный ниже простейший диалплан использует приложение `Echo()` для проверки работы двунаправленной связи в канале:

```
[incoming]
; входящие вызовы, поступающие через порт FXO,
```

```
;направляются в этот контекст из zapata.conf
exten => s,1,Answer()
exten => s,n,Echo()
```

Приложение `Echo()` будет возвращать все, что вы скажете.

Звонок

Теперь, когда канал FXO сконфигурирован, протестируем его. Запустите приложение `ztool` и подключите линию PSTN к FXO-порту на плате TDM400P. Как только телефонная линия будет подключена к FXO-порту, красный световой индикатор погаснет.

Теперь наберите номер PSTN с другого внешнего телефона (например, мобильного). Asterisk ответит на вызов и выполнит приложение `Echo()`. Если вы услышали в трубке свой голос, значит, FXO-канал установлен и сконфигурирован успешно.

Конфигурация канала FXS для аналогового телефона

Конфигурация канала FXS аналогична настройке канала FXO. Рассмотрим ее подробнее.

Конфигурация оборудования Zaptel

Ниже представлена минимальная конфигурация для канала FXS платы TDM400P. Эта конфигурация аналогична приведенной выше конфигурации канала FXO, добавлена только строка `fxoks=1`.

Из наших предыдущих обсуждений вы должны помнить, что для каналов FXO и FXS используются противоположные типы протоколов обмена сигналами, поэтому для канала FXS мы зададим протокол FXO. В приведенном ниже примере канал 1 конфигурируется на использование протокола FXO с протоколом обмена сигналами `kewlstart`:

```
fxoks=1
fxsks=2
loadzone=us
defaultzone=us
```

После загрузки драйверов для своего оборудования можно проверить статус оборудования с помощью команды `/sbin/ztcfg -vv`:

```
Zaptel Configuration
=====
```

```
Channel map:
```

```
Channel 01: FXO Kewlstart (Default) (Slaves: 01)
Channel 02: FXS Kewlstart (Default) (Slaves: 02)
```

```
2 channels configured.
```

Конфигурация оборудования Zapata

Следующая конфигурация аналогична настройке канала FXO, добавлен только раздел для FXS-порта и строка `immediate=no`. Контекстом FXS-порта является `phones`, протокол обмена сигналами – `fxoks (kewlstart)`, номер канала – 1.

Каналы FXS можно сконфигурировать на выполнение одного из двух различных действий при снятии трубки телефона. Наиболее часто используемым (и обычно предпочтительным) вариантом для Asterisk является воспроизведение тонального сигнала готовности линии и ожидание ввода пользователя. Это действие конфигурируется строкой `immediate=no`. Альтернатива для Asterisk – вместо воспроизведения тонового сигнала автоматическое выполнение набора инструкций, заданных в диалплане. Такое поведение определяется строкой `immediate=yes`¹. Инструкции, которые должны выполняться для канала, находятся в заданном для него контексте и будут соответствовать добавочному номеру `s` (обе эти темы будут обсуждаться подробнее в следующей главе).

Вот наш новый файл `zapata.conf`:

```
[trunkgroups]
; описание всех магистральных групп

[channels]
; аппаратные каналы
; значения по умолчанию
usecallerid=yes
hidecallerid=no
callwaiting=no
threewaycalling=yes
transfer=yes
echocancel=yes
echotraining=yes
immediate=no

; описание каналов
context=phones ; Используется контекст [internal], описанный в extensions.conf
signalling=fxo_ks ; Для канала FXS используется протокол обмена сигналами FXO

channel => 1 ; Телефон подключается к порту 1
context=incoming ; Входящие вызовы направляются в контекст [incoming],
; описанный в extensions.conf
signalling=fxs_ks ; Для канала FXO используется протокол обмена сигналами FXS
channel => 2 ; PSTN подключается к порту 2
```

¹ Это также называют методом `Watphone` или, более формально, автоматическим прямым вызовом (`Automatic Ringdown`) либо автоматическим прямым вызовом частной линии (`Private Line Automatic Ringdown, PLAR`). Этот метод широко применяется на стойках оформления проката автомобилей и в аэропортах.

Конфигурация диалплана

Мы воспользуемся простейшим диалпланом, сконфигурированным ранее в данной главе для тестирования FXS-порта с помощью приложения `Echo()`. Соответствующий раздел, который уже должен присутствовать в диалплане, выглядит следующим образом:

```
[internal]
exten => 500,1,Verbose(1|Echo test application)
exten => 500,n,Echo()
exten => 500,n,Hangup()

[phones]
include => internal
```

Приложение `Echo()` будет возвращать все, что вы скажете.

Конфигурация SIP-телефонов

Протокол Session Initiation Protocol (SIP)¹, обычно применяемый в VoIP-телефонах (как аппаратных, так и программных), отвечает за установку и разъединение соединения, а также за любые изменения, происходящие во время соединения, такие как переадресации. Назначение SIP – помочь двум конечным точкам поговорить друг с другом (по возможности напрямую). Протокол SIP – это просто протокол обмена сигналами, то есть его задачей является лишь обеспечить возможность двум конечным точкам говорить друг с другом, но не работа с носителем вызова (голосом). Передача голоса осуществляется с помощью другого протокола – Real-Time Transport Protocol (транспортный протокол реального времени – RTP; RFC 3550) – для передачи медиа-данных непосредственно между двумя конечными точками.



Термин «медиа-данные» используется здесь для обозначения данных, передаваемых между конечными точками и используемых для воссоздания голоса на противоположном конце провода. Также он может использоваться в случае воспроизведения музыки или голосовых сообщений офисной АТС.

В мире SIP конечные точки называются агентами пользователя, которые могут быть двух типов: клиент и сервер. Клиент – это конечная точка, формирующая запрос, а сервер обрабатывает этот запрос и формирует ответ. Когда конечная точка желает выполнить вызов другой

¹ RFC 3261 доступен по адресу <http://www.ietf.org/rfc/rfc3261.txt>. Документ довольно объемный, но тем, кто желает стать специалистом в Asterisk, рекомендуем прочитать по крайней мере первые 100 страниц и разобраться, как устанавливать соединения, поскольку эти знания будут необходимы для работы с историей SIP (`sip debug` из консоли Asterisk) и поиска с ее помощью причины невозможности установления соединений.

конечной точки (например, наш программный телефон звонит на другой программный телефон), она формирует запрос и отправляет его на прокси-сервер SIP¹. Прокси-сервер принимает запрос, определяет его место назначения и направляет его туда. Если два агента пользователя успешно договорились и установили вызов, переносимый сигнал передается по RTP-протоколу и пересылается непосредственно от одного агента пользователя другому. SIP-прокси не обрабатывают медиа-данные; они просто работают с SIP-пакетами.

С другой стороны, Asterisk называют Back-To-Back User Agent (B2BUA). Это означает, что Asterisk действует как агент пользователя или в роли сервера (принимающий), или в роли клиента (посылающий). Итак, когда программный телефон звонит на добавочный номер, соединение устанавливается непосредственно между программным телефоном и Asterisk. Если логика, реализованная в Asterisk, определяет, что вызов адресован другому агенту пользователя, Asterisk действует как клиент и устанавливает другое соединение (известное как канал) с другим телефоном. При этом медиа-информация передается от телефона к телефону прямо через Asterisk². С точки зрения телефонов они взаимодействуют непосредственно с Asterisk.

Базовая конфигурация SIP-телефонов в Asterisk

Конфигурация SIP-телефона для работы с Asterisk не требует много усилий и времени. Однако здесь можно легко запутаться из-за обилия опций как в Asterisk, так и в конфигурации конкретного телефонного аппарата или программного телефона. Добавьте к этому тот факт, что одни и те же вещи могут называться по-разному, – вот и прекрасный повод для того, чтобы впасть в отчаяние. Поэтому мы собираемся рассмотреть лишь самые основные вопросы. У тех, кто следует нашим советам, должно получиться заставить работать рассматриваемые здесь аппараты (а также уверенно справиться с неупомянутыми здесь телефонами). Мы не говорим, что это лучший или даже верный путь, но это самый простой путь. Намного проще взять уже работоспособную базу и настраивать ее, добываясь необходимого решения.



Точно так же, как мы делали это для файла `extensions.conf`, выполните следующие команды в оболочке `bash`:

¹ OpenSER – превосходный прокси-сервер SIP с открытым исходным кодом, доступен по адресу <http://www.openser.org>.

² Да, существуют способы направить поток непосредственно между телефонами после того, как вызов установлен. Это можно настроить в файле `sip.conf`, используя или `directrtpsetup=yes` (экспериментальная опция, позволяющая перенаправлять переносимый сигнал в исходной настройке соединения), или `canreinvite=yes` (при этом переносимый сигнал проходит через Asterisk только в начале, до повторного сообщения `INVITE`, после чего сигнал может быть направлен непосредственно от телефона к телефону).

```
# mv sip.conf sip.conf.sample
# touch sip.conf
```

Определение SIP-устройства в Asterisk

Если внести следующие строки в файл `sip.conf`, можно будет зарегистрировать телефон в системе.

```
[general]

[1000]
type=friend
context=phones
host=dynamic
```

Несимпатично, небезопасно, не обладает гибкостью, неполнофункционально, но это будет работать.

Даже несмотря на то что это SIP-устройство названо 1000 и, вероятно, ему будет присвоен именно этот добавочный номер, следует отметить, что имя устройства может быть произвольным. `mysipset, john, 0004f201ab0c` — все эти имена действительны, широко используются и даже, возможно, больше отвечают требованиям пользователей¹. Главное, чтобы присваиваемое имя было уникальным идентификатором устройства, который станет частью мандата при выполнении вызова по каналу SIP.

Поскольку мы хотим как направлять вызовы в программный телефон, так и обеспечить клиенту возможность размещать вызовы, параметр `type` (тип) был определен как `friend` (друг). Существует еще два параметра: `user` (пользователь) и `peer` (равноправный участник сети). С точки зрения Asterisk `user` задается для входящих вызовов, а `peer` — для исходящих звонков (через приложение `Dial()`). `friend` — это просто краткая запись, определяющая и пользователя, и равноправного участника. Если есть сомнения, задавайте тип `friend`.

Опция `host` (хост) используется для определения местонахождения клиента в сети, когда Asterisk необходимо направить ему вызов. Это значение может быть задано статически, например `host=192.168.1.100`, или, если клиент имеет динамический IP-адрес, задается `host=dynamic`. Если для опции `host` задано значение `dynamic` и клиент сконфигурирован для автоматической регистрации, Asterisk получит от конечной точки (то есть от телефонного аппарата или программного телефона) пакет REGISTER, из которого Asterisk сможет узнать, какой IP-адрес использует равноправный SIP-участник.

Если вы не доверяете своей сети, вероятно, следует задать пароль. Для этого в описание устройства добавляется следующая строка. Это один из тех параметров, которые не являются обязательными, но желательными:

```
secret=guessthis
```

¹ Максимальная длина имени пользователя — 255 символов.