

ASTERISK И LINUX – МИССИЯ IP-ТЕЛЕФОНИЯ



У вас уже есть офисная мини-АТС или вы только планируете ее покупку? Думаете о предоставлении новых возможностей для повышения эффективности работы сотрудников? Сегодня мы приоткроем дверь в мир Open Source IP-телефонии, рассмотрев возможности одного из самых популярных продуктов для Linux – Asterisk PBX.

МИХАИЛ ПЛАТОВ

Итак, что же такое Asterisk? Читаем на официальном сайте [1]: «Asterisk – полностью программная учрежденческая АТС (УАТС), работающая под управлением операционной системы Linux». По возможностям (см. [2]) Asterisk зачастую опережает своих традиционных собратьев. Поддерживаются практически все популярные протоколы IP-телефонии (SIP, H323, SCCP, ADSI). Помимо стандартных и общеизвестных, Asterisk также имеет свой собственный протокол – IAX.

Его можно использовать как для подключения обычных клиентов, так и для передачи голосового трафика между несколькими серверами Asterisk. Для сопряжения с «традиционной телефонией» имеются аналоговые (FXO, FXS) и цифровые модули расширения (E1, T1). С помощью As-

terisk вы сможете реализовать проект любого масштаба – от простого домашнего сервера голосовой почты до телефонного сервера предприятия с функциями IVR (Interactive Voice Response – система голосовых меню).

Сегодня мы рассмотрим простой пример установки и настройки сервера Asterisk для работы с несколькими пользователями.

Можно ли на него посмотреть?

Думаю, что после знакомства со списком возможностей (а это более 70 позиций) у вас возникло желание как можно быстрее взглянуть на Asterisk своими глазами. Спешу вас обрадовать – посмотреть на Asterisk без его установки можно! Для этого воспользуйтесь специальным LiveCD, обра-

зами сетевой загрузки для PXE-совместимой сетевой платы или карты CompactFlash [3].

Однако, если вы считаете LiveCD или другие способы «знакомства» излишними, можете сразу перейти к установке, только предварительно давайте разберемся, что именно мы будем устанавливать.

Каким бывает Asterisk

Существует две версии Asterisk – HEAD и STABLE. Первая – версия для разработчиков и «продвинутых» пользователей. Все новые возможности первоначально добавляются именно в HEAD. Версия STABLE, напротив, содержит только проверенные и отлаженные компоненты и модули. В ней функциональность заведомо приносится в жертву стабильности и надежности. Сильно расстраиваться по этому поводу не стоит – немного терпения и все самое «модное» из HEAD плавно перейдет в STABLE.

Итак, для начинающих пользователей, желающих максимально быстро получить работающее и проверенное решение, рекомендуется использовать STABLE-версию. (Её же в силу простоты мы будем описывать в статье.)

Если же вы собираетесь принять участие в процессе разработки Asterisk или просто хотите всегда использовать все самое последнее и новое – устанавливайте версию HEAD.

Процесс установки. Что нам стоит «*» построить

Asterisk является свободно распространяемым продуктом для платформы. Рекомендуемый способ установки – компиляция из исходных кодов. (Существуют бинарные пакеты для Redhat/Fedora, Debian, а также ebuild для Gentoo, правда, как правило, более старой версии по сравнению с тем, что доступно на сайте.) Однако прежде чем что-либо компилировать, необходимо получить исходный код Asterisk. Сделать это можно двумя способами:

- Скачать tarball с официального сайта [1].
- Загрузить исходный код с CVS-сервера проекта.

Первый способ достаточно прост, и вряд ли он вызовет какие-то проблемы, поэтому при установке Asterisk воспользуемся вторым способом. К тому же, если вам потребуется установить HEAD-версию, вам просто придется использовать CVS. (Для Asterisk основной веткой CVS является HEAD, поэтому для загрузки STABLE-версии необходимо указать ревизию (-r v1-0).) Процесс компиляции для обеих веток абсолютно одинаков, поэтому все, что написано далее, применимо и для HEAD, и для STABLE.

Итак, нам понадобятся 3 модуля: zaptel – модуль аналоговой телефонии, libpri – модуль цифровой телефонии, и сам Asterisk. Для загрузки модулей с сервера CVS выполните следующие команды:

```
# cd /usr/src
# export CVSROOT=:pserver:anoncvs@cvcs.digium.com:/usr/cvsroot
# cvs login # введите пароль anoncvs
# cvs checkout -r v1-0 zaptel libpri asterisk
```

После выполнения последней команды в /usr/src будут созданы соответствующие подкаталоги. Процесс уста-

новки достаточно прост – необходимо поочередно зайти в каждый из каталогов и выполнить команды make clean и make install:

```
# cd zaptel
# make clean install
# cd ../libpri
# make clean install
# cd ../asterisk
# make clean install samples
```

Последняя команда не только скомпилирует и установит Asterisk, но и скопирует стандартные файлы конфигурации в /etc/asterisk. Вообще установочный скрипт копирует файлы в 3 основных каталога:

- /etc/asterisk – расположение конфигурации;
- /usr/lib/asterisk/modules – место хранения модулей Asterisk;
- /var/lib/asterisk/ – здесь находятся дополнительные файлы asterisk (звуки, agi-скрипты, картинки и т. д.).

Сам же исполняемый файл обычно помещается в /usr/sbin/asterisk.

Для проверки правильности установки выполним следующую команду:

```
# asterisk -vvvvvcd
```

Если все нормально, то мы увидим несколько экранов пока еще непонятной для нас информации, заканчивающейся приветствием командного интерфейса Asterisk (см. рис. 1).

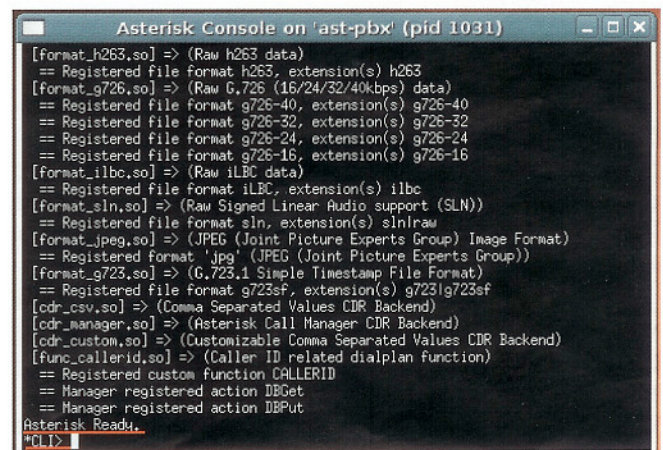


Рисунок 1. Консоль Asterisk

Не вдаваясь в подробности, напомним в консоли stop now и перейдем к более детальной настройке.

Чего же мы хотим?

Прежде чем приступить к редактированию конфигурационных файлов Asterisk, давайте определим, что же именно мы хотим от него получить. Итак, нам необходимо:

- Обеспечить возможность разговора абонентов по IP (SIP).
- Создать персонализированные голосовые ящики (с автоматической отсылкой голосовых сообщений на e-mail).
- Настроить «перехват», передачу и парковку звонков.

Таблица 1. Сравниваем возможности программных клиентов SIP

Возможность	X-ten xlite	Sjphone	FireFly	MS Messenger	kPhone
Поддержка протокола SIP	+	+	+	+	+
Поддержка протокола IAX	—	—	+	—	—
Работа с Asterisk	+	+	+	+	+
Поддерживаемые ОС	Windows	Windows, Linux	Windows	Windows	Linux
Поддерживаемые кодеки	G711, GSM, iLBC, Speex	G711, GSM, iLBC	G711, G729 ¹ , GSM, iLBC, Speex	G711	G711, iLBC, GSM
Поддержка функций Caller-ID, Call Transfer, Call Hold	+	+	+	—	+
Интерфейс пользователя ²	4	5	3	3	2
Возможность учета времени разговора	+	+	+	—	—
Доступность	бесплатно ²	бесплатно ³	бесплатно	бесплатно	бесплатно

С задачами вроде бы определились, поэтому, не откладывая дела в долгий ящик, приступим к ее выполнению.

А ты меня слышишь?

Первое, что мы сделаем с нашим сервером, – научим его обслуживать внутренние звонки. Начнем с редактирования файла /etc/asterisk/sip.conf. Убедимся, что в секции general присутствуют следующие строки:

```
[general]
context=default
allowguest=no
bindport=5060
bindaddr=0.0.0.0
```

В первой строке мы определяем контекст, в который будут попадать все входящие звонки. Во второй строке мы запрещаем неавторизованный доступ к нашему серверу, что особенно важно в случае, если Asterisk будет доступен из Интернета. В двух последующих строках мы задаем порт и интерфейс, на котором ожидаем запросы SIP-клиентов.

Теперь несколько слов о самих клиентах – они могут быть как программными (приложение на компьютере), так и аппаратными (автономное устройство, реализующее один из протоколов Asterisk). Для первого знакомства с Asterisk нам вполне хватит возможностей программных клиентов. Наиболее известные из них: X-Ten X-Lite, sjPhone, FireFly, MS Messenger, kPhone. Возможности этих программ лучше всего сравнивать, представив их в виде таблицы (см. **таблицу 1**).

В зависимости от решаемой задачи можно использовать различные программные клиенты. Если ваш сервер будет преимущественно использоваться для внутренних звонков, то имеет смысл обратить внимание на программы с более функциональным интерфейсом (X-lite, sjPhone). В случае большого количества междугородних звонков (либо работы в условиях ограниченной пропускной способности или ненадежного канала) лучше использовать программы,

поддерживающие кодек G729 (из бесплатных – firefly). Для тестового примера предположим, что у нас есть 3 пользователя, использующие X-lite, sjPhone и MS Messenger соответственно. Отразим эту информацию в конфигурационных файлах. Начнем с sip.conf:

```
[200] ; xlite phone
type=friend
host=dynamic
username=200
secret=user1_password
nat=no
canreinvite=no
context=office
callerid="User1" <200>
allow=gsm
allow=ulaw
allow=alaw

[201] ;sjphone
type=friend
host=dynamic
username=201
secret=user2_password
dtmfmode=rfc2833
context=office
callerid="User2" <202>
allow=gsm
allow=ulaw
allow=alaw

[203] ; ms messenger
type=friend
host=dynamic
username=202
secret=user3_password
dtmfmode=rfc2833
context=office
callerid="User3" <203>
allow=ulaw
allow=alaw
```

Небольшие пояснения по параметрам конфигурации. Поле type, определяющее тип клиента, может принимать три значения:

- user – возможны только входящие звонки;
- peer – возможны только исходящие звонки;

¹ Доступно с использованием бесплатной внешней библиотеки.
² Здесь приводится субъективная оценка автора.
³ Поддержка расширенной функциональности доступна в коммерческой версии.

■ friend – разрешены как исходящие, так и входящие вызовы.

Поле host позволяет определить IP-адрес, с которого разрешены подключения для данного клиента. В нашем случае задавать такое ограничение особого смысла нет, поэтому оставим динамические адреса (dynamic). Поля username и secret определяют имя пользователя и пароль, которые должны использоваться клиентом при регистрации. Dtmfmode задает способ передачи сигналов DTMF. Параметр nat говорит, может ли данный клиент работать из-за NAT. Canreinvite разрешает (yes) или запрещает (no) установку прямого соединения между участниками (минуя Asterisk). Некоторые клиенты поддерживают строго определенные способы работы, что необходимо отражать в файлах конфигурации. Context – имя контекста, в котором будет находиться данный абонент. CallerID – идентификационная строка, отображается при звонках от данного пользователя (АОН).

После задания всех вышеописанных настроек и сохранения файла sip.conf перейдем к определению номерного плана. Для этого нам понадобится отредактировать файл /etc/asterisk/extensions.conf. Но, прежде чем запустить редактор, давайте более детально посмотрим на то, что входит в стандартную поставку.

Номерной план в Asterisk (Dial plan) задается при помощи так называемых расширений (extension). В файле стандартной установки можно увидеть следующее:

```
[demo]
exten => s,1, Wait,1
exten => s,n, Answer
exten => s,n, SetVar(TIMEOUT(digit)=5)
exten => s,n, SetVar(TIMEOUT(response)=10)
exten => s,n(restart), Background(demo-congrats)
exten => s,n(instruct), Background(demo-instruct)
exten => s,n, WaitExten
```

Обработка контекста demo начинается с записей, в которых «S» является первым символом. Так, сначала Asterisk подождет 1 секунду, потом поднимет трубку и ответит на звонок, установит значения переменных (строки 2, 3), проигрывает содержимое файлов demo-congrats и demo-instruct и будет ожидать набора добавочного номера. Следующие строки определяют конкретные номера в контексте demo:

```
exten => 2,1, Background(demo-moreinfo)
exten => 2,n,Goto(s,instruct)
exten => 3,1, SetVar(Language)=fr
exten => 3,2, Goto(s, restart)
exten => 8500, 1, VoiceMailMain
exten => 8500, n, Goto,s
exten => 1000, 1, Goto(default,s,1)
```

Итак, определены 4 номера – 2, 3, 8500 и 100. При звонке на номер 2 мы прослушаем сообщение demo-moreinfo, после которого Asterisk перенаправит нас в начало, к сообщению demo-instruct. Набрав номер 3, мы изменим язык данной сессии на французский. (Чтобы Asterisk мог «разговаривать» с вами на французском языке, необходимо наличие соответствующих файлов озвучки в папке /var/lib/asterisk/sounds/fr.) При наборе номера 8500 нам ответит си-

стема голосовой почты Asterisk. Номер 1000 отправит нас в начало контекста demo.

Надеюсь, что теперь ситуация с контекстами немного прояснилась. Теперь добавим наши телефоны в этот номерной план. Для этого создадим следующий контекст:

```
[office]
exten => 200,1, Macro(stdexten,200,SIP/200)
exten => user1, 1, Goto(200|1)
exten => 201,1, Macro(stdexten,201,SIP/201)
exten => user2, 1, Goto(201|1)
exten => 202,1, Macro(stdexten,202,SIP/202)
exten => user3, 1, Goto(202|1)
include => demo
```

Мы определили три номера и три псевдонима, при наборе которых активируется макрос stdexten, отвечающий за обработку вызовов пользователей. Псевдонимы позволяют звонить абоненту не только по его номеру, но и по имени. В качестве параметров макросу передается номер данного пользователя и имя его SIP-устройства. Давайте посмотрим, что же делает этот макрос:

```
[macro-stdexten]
exten => s, 1, Dial (${ARG2},20,t4)
exten => s, 2, Goto(s-${DIALSTATUS},1)
exten => s-NOANSWER, 1, Voicemail(u${ARG1})
exten => s-NOANSWER, 2, (Goto(default,s,1))
exten => s-BUSY, 1, Voicemail(b,${ARG1})
exten => s-BUSY, 2, (Goto(default,s,1))
exten => _s-, 1, Goto(s-NOANSWER)
exten => a, 1, VoiceMailMain(${ARG1})
```

Данный макрос звонит указанному пользователю. Если в течение 20 секунд никто не ответил или вызываемый абонент был занят, звонящему проиграется соответствующее сообщение, после чего он получит возможность оставить сообщение голосовой почты. Правда, пока у наших абонентов голосовых ящиков еще нет. Исправим это недоразумение, добавив в файл /etc/asterisk/voicemail.conf следующие строки:

```
[general]
format=wav49
maxmessage=300
[office]
200 => 123, User1, user1@mail.company.com,,attach=yes
201 => 456, User2, user2@mail.company.com,,attach=yes
202 => 789, User3, user3@mail.company.com,,attach=yes
```

Этим мы определили 3 ящика голосовой почты для наших пользователей, с паролями 123, 456, 789 соответственно. При поступлении нового сообщения пользователю незамедлительно будет отсылаться e-mail на указанный адрес. Во вложении письма будет присутствовать само голосовое сообщение в формате wav (вместо wav можно использовать любой формат, поддерживаемый сервером). Максимальная длина сообщения – 5 минут. Если в качестве параметра ящика указать attach=no, то высылаться будет только уведомление об оставленных сообщениях. Для того чтобы сообщения отправлялись, нам понадобится определить еще один параметр – команду для отправки почты. По умолчанию она имеет следующий вид:

```
;mailcmd=/usr/sbin/sendmail -t
```

⁴ Символ t отсутствует в стандартной конфигурации. Если мы хотим использовать передачу звонков, его необходимо добавить.

Как видно из названия, для работы данной функции необходимо, чтобы на компьютере с Asterisk был установлен почтовый сервер sendmail или другой, имеющий подобной интерфейс вызова (postfix, exim и др.). Информацию по настройке почтового сервера можно найти в [4-6].

Для работы с голосовой почтой с телефона в нашем номерном плане используется номер 8500. После ввода номера телефона и пароля пользователь получает доступ к интерфейсу управления своим голосовым ящиком. Помимо стандартных возможностей прослушивания перемещения и удаления сообщений пользователь может самостоятельно записывать персональные приветствия, менять свой пароль или отправлять голосовые сообщения другим пользователям.

На этом базовую настройку Asterisk можно считать законченной, теперь самое время перейти к настройке наших программных клиентов.

Настраиваем X-Lite

Идем на сайт <http://www.xten.com>. Скачиваем, устанавливаем и запускаем программу X-Lite. При первом запуске программа предложит настроить микрофон и откроет окно настроек SIP прокси-сервера. В этом окне определим следующее:

```
Enabled - Yes
Display Name - Asterisk
Username - 200
Authorization User - 200
Password - user1_password
Domain/Realm - asterisk
SIP Proxy - 192.168.0.205
Outbound Proxy - 192.168.0.20
Register - Always
```

Если все настройки сделаны правильно, то в окне X-Lite мы увидим:



Рисунок 2. X-Lite, зарегистрированный на сервере Asterisk

Теперь можно попробовать позвонить по тестовым номерам стандартной инсталляции – 2, 3, 1000, 1234, 1236, 500, 600, 8500, 200, 201, 202. При звонках на последние два номера вам, скорее всего, ответит Allison Smith – обладательница приятного голоса, которым разговаривает Asterisk. Эти телефоны пока еще не настроены, и о том, как их можно найти, Asterisk еще ничего не знает. Что ж, давайте исправим это недоразумение!

Настраиваем sjphone

Скачиваем с сайта <http://www.sjlabs.com> программу sjphone.

Устанавливаем и запускаем соответствующий файл установки. Для настройки нажимаем на «кнопку с отверткой» (см. рис. 3).

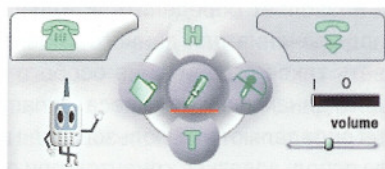


Рисунок 3. Элемент интерфейса sjPhone после установки

Переходим на закладку «Profile» и нажимаем на кнопку «New». Создадим новый профиль «Asterisk» с типом «Calls though SIP Proxy». Перейдем на закладку «SIP Proxy» и введем IP-адрес (или имя) сервера Asterisk и нажмем «OK» (см. рис. 4).

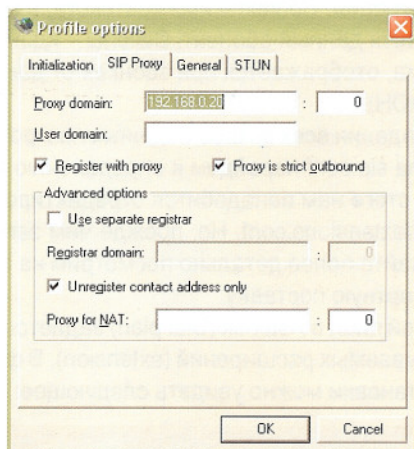


Рисунок 4. Настраиваем параметры SIP sjPhone

В появившемся окне введем имя и пароль для данного сервера – 201 и user2_password соответственно. После закрытия окна свойств sjphone изменит свой вид. Если данные введены правильно, то мы увидим следующее (см. рис. 5).



Рисунок 5. sjPhone, зарегистрированный на сервере Asterisk

Настраиваем Windows Messenger

Если устанавливать специального клиента не хочется, можно попробовать разговаривать с помощью Windows Messenger. Работа с Asterisk не является для него стандартной

⁵ IP-адрес или DNS-имя машины с Asterisk.

возможностью, так что вооружимся редактором реестра и отправимся в путь. Если мы используем Windows Messenger версии ниже 4.7, то нам понадобится следующий ключ реестра: HKEY_CURRENT_USER\Software\Microsoft\MessengerService.

Если же установлена версия 5.1, то ключ будет другим⁶: HKEY_CURRENT_USER\Software\Policies\Microsoft\Messenger.

По данному адресу нам необходимо исправить значение поля CorpPC2Phone с «0» на «1».

Перезапустим Messenger. Из меню откроем «Tools → Options → Accounts». В поле «Communications Service Account» введем имя пользователя Asterisk – 202@asterisk. Нажмите на кнопку «Advanced» и в открывшемся диалоговом окне введите адрес сервера Asterisk, попутно отметьте UDP в качестве транспорта. Теперь попробуем подключиться к серверу Asterisk (sign in) с именем 202@asterisk. В случае успеха мы увидим следующее (см. рис. 6).

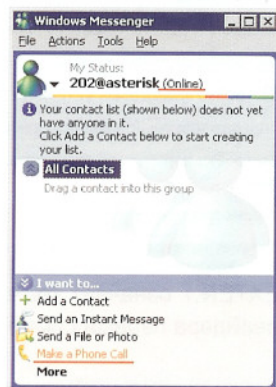


Рисунок 6. Windows Messenger, зарегистрированный на сервере Asterisk

Итак, теперь все три клиента настроены. Связь можно проверить, позвонив с каждого из них на приведенные выше номера нашего сервера.

Иду на перехват!

Как вы, наверное, уже догадались, сейчас речь пойдет о «перехвате звонков». Для начала несколько слов о том, что же это такое в понимании Asterisk. Допустим, вы слышите, как у человека рядом звонит телефон, а его самого на месте нет. По специфике вашей деятельности вы сами можете ответить на этот звонок, не заставляя звонившего лишней раз обращаться к голосовой почте. Для того чтобы не подходить к телефону физически, можно «перехватить» звонок со своего телефона. Нажав определенную последовательность клавиш, вы сможете ответить на «чужой» вызов. В Asterisk поддерживаются два вида таких «перехватов»:

- в рамках указанной группы;
- конкретного номера.

Для перехвата конкретного номера вы набираете на телефоне: *8#номер_звонящего_телефона, после чего звонок переключается на ваш номер. Первый способ более уни-

версален. Вы определяете так называемые группы вызова – телефоны, звонки на которые можно перехватывать, а также для каждого из телефонов определяете, какие группы они могут перехватывать (см. рис. 7).



Рисунок 7. Группы вызова и группы перехвата

В нашем случае мы реализуем самый простой вариант перехвата звонков – все телефоны смогут перехватывать звонки друг друга. Для этого внесем некоторые дополнения в файл sip.conf. В секциях описания наших телефонов (200, 201, 202) добавим две строчки:

```
callgroup=1
pickupgroup=1
```

Теперь добавим саму возможность «перехвата» звонков, внося соответствующие изменения в номерной план. В контекст [office] после правил вызова абонентов добавим следующую строчку⁷:

```
exten => *8,1,PickUP()
```

Теперь, после перезагрузки номерного плана Asterisk, у нас появится возможность перехвата вызовов. Для этого, когда вы услышите звонок телефона, просто нажмите *8, и разговор будет переключен на ваш телефон.

Можно здесь припарковаться?

Под «парковкой» в Asterisk понимается следующая ситуация: вам кто-то позвонил, в процессе разговора вы поняли, что вам нужно перейти к другому телефону. Вы «паркуете» ваш звонок в определенном месте, подходите к другому телефону и «подключаете» оставленный ранее звонок. Парковка в Asterisk сделана с использованием другой удобной возможности – передачи звонка. Определив некую последовательность символов (по умолчанию это 1#), можно в процессе разговора переводить звонки другим абонентам. Специфика же парковки заключается в том, что есть некие места – парковочные слоты, в которые собеседников можно помещать и забирать.

Давайте разберемся со всем этим на практике. Параметры парковки и передачи вызовов определяются в файле features.conf. В стандартной инсталляции для парковки звонков определяется 20 номеров, начиная с номера 701. Для активации функции убедимся, что файл features.conf содержит следующие данные:

```
[general]
parkext => 700
parkpos => 701-720
```

⁶ Если такого пути нет, его необходимо создать. CorpPC2Phone имеет тип DWORD.

⁷ Если вам нужно просто включить возможность перехвата для всех абонентов Asterisk, вместо модификации номерного плана можно просто раскомментировать строчку pickupexten в файле features.conf.

```
context => parkedcalls
xfersound =beep
xferfailsound = beeper
featuredigittimeout = 1000
```

```
[featuremap]
blindxfer => #1
```

Итак, для «парковки» звонка будем использовать номер 700, при его обратном подключении (или ошибке подключения) мы будем слышать звуковой сигнал, для активации режима передачи звонка будем использовать последовательность #1, при этом пауза между нажатиями кнопок не должна превышать одной секунды.

Теперь внесем изменения в номерной план, добавив в контекст [office] следующую строчку:

```
include => parkedcalls
```

Теперь, если во время разговора мы наберем 1#700, звонок «припаркуется» в свободный слот (Эллисон проговаривает нам его номер). Затем мы можем подойти к другому телефону (при данных настройках на это у нас есть 45 секунд) и, набрав известный нам номер слота, продолжить разговор с данного телефона.

Включаем межгород (Межгород? Соединяю...)

Для совершения междугородних звонков можно использовать как более «родной» для Asterisk протокол SIP, так и более распространенный среди российских провайдеров IP-телефонии протокол H323. Поддержка H323 появляется после установки определенных модулей (в нашей инсталляции этих модулей пока нет), а вот SIP присутствует по умолчанию. Кроме того, есть как минимум один российский провайдер, работающий с этим протоколом, — сеть Тарьо [7]. Работа в данной сети не бесплатна, поэтому предварительно нам придется зарегистрироваться⁸. После этого можно приступить к настройке. Откроем уже знакомый нам файл sip.conf и добавим в него следующие записи⁹:

```
[tario_out]
type=peer
host=адрес_сервера_tario
username=номер_лицевого_счета_tario
fromusername=номер_лицевого_счета_tario
fromdomain=домен_tario
secret=пароль_tario
usereqphone=yes
```

Для входящих вызовов:

```
register => номер_лицевого_счета_tario:пароль@tario_in
[tario_in]
type=user
context=tario_inc
host=ip_адрес_сервера_tario
```

Для исходящих звонков через этого провайдера будем использовать префикс 8. В контексте [office] добавим следующую строчку:

```
exten => _8, Dial(SIP/${EXTEN:1}@tario_out,20,rT)
```

Описание конфигурационных файлов Asterisk

musiconhold.conf	Настройка параметров «музыки при удержании» MusicOnHold
cdr_mysql.conf	Настройка хранения записей о звонках в MySQL
manager.conf	Настройка интерфейса Manager
meetme.conf	Настройка конференций (meetme)
mgcp.conf	Настройка параметров MGCP-устройств
parking.conf	Параметры «парковки» вызовов «Call parking»
voicemail.conf	Параметры голосовой почты (VoiceMail)
agents.conf	Агенты Asterisk
extensions.conf	Номерной план Asterisk
iax.conf	Настройка IAX-устройств
modem.conf	Настройка ISDN с использованием ISDN4Linux
phone.conf	Настройка интерфейса телефонии Linux
vpb.conf	Настройка аппаратуры Voicetronix
alsa.conf	Настройки звукового драйвера ALSA для консольного пользователя
festival.conf	Параметры работы с синтезатором речи festival
modules.conf	Настройка модулей Asterisk
zapata.conf	Настройка аналоговых устройств телефонии digium
asterisk.conf	Определение расположения каталогов Asterisk
indications.conf	Настройка параметров аналоговых АТС различных стран
rpt.conf	Параметры протокола rtp
logger.conf	Параметры журналирования Asterisk

Конструкция EXTEN:1 означает то, что при передаче звонка шлюзу провайдера первый символ будет отбрасываться.

Все входящие звонки от провайдера будем принимать в контекст [tario_inc]. Его можно сконфигурировать примерно так:

```
[tario_inc]
exten => s,1,Wait, 1
exten => s, 2, Answer
exten => s,3, Background(local-welcome)
exten => s,4, WaitExten
exten => 200,1, Macro(stdexten,200,SIP/200)
exten => 201,1, Macro(stdexten,201,SIP/201)
exten => 202,1, Macro(stdexten,202,SIP/202)
exten => 8500,1, VoiceMailMain
exten => 8500,n, Hangup
```

При звонке в этот контекст пользователь сначала услышит сообщение local-welcome (сообщение нужно будет предварительно записать), в котором его попросят ввести добавочный номер вызываемого абонента. После введения номера звонящего соединят с требуемым пользователем, если же этот пользователь недоступен, звонивший получит возможность оставить сообщение голосовой почты. Подобно доступу к шлюзу голосовой почты мы также можем предоставить доступ к другим функциям Asterisk.

Несколько слов о безопасности

При определении контекстов, доступ к которым будут получать внешние пользователи, необходимо дважды (а то и трижды) подумать. Так, если бы в последнем примере

⁸ О том, как это сделать, вы сможете прочитать на официальном сайте [7].

⁹ Предполагается, что подключение к провайдеру производится с компьютера, имеющего реальный интернет-адрес (без NAT).

вместо «ручного» создания правил для дозвона до каждого пользователя мы бы просто написали:

```
include => office
```

то все внешние пользователи, звонящие на наш сервер, помимо вполне легального доступа к внутренним номерам и серверу голосовой почты также получили бы не совсем легальный доступ к междугородним звонкам с использованием нашего провайдера. Однако дублировать записи вызова тоже не слишком красиво, поэтому более правильно было бы перенести все доступные извне правила маршрутизации в отдельный контекст [office-rules], а в контексте [office] вместо них написать так:

```
[office]
include => [office-rules]
exten => _8, Dial(SIP/${EXTEN:1}@tario_out,20,rT
```

После этого в контексте [tario-inc] вместо повторного перечисления правил дозвона до пользователей можно смело написать include => [office-rules]. При такой схеме office-rules не содержит ничего лишнего, и соответственно воспользоваться нашим «межгородом» внешние пользователи уже не смогут.

Заключение

Итак, сегодня мы рассмотрели основы настройки Asterisk и связанные с этим подробности – файлы конфигурации пользователей, настройки номерного плана, настрой-

ки «дополнительных» возможностей, а также способы конфигурации нескольких популярных программных клиентов. На этом наше знакомство с Asterisk не заканчивается. Как вы могли видеть, процесс конфигурации не является очень простым для непосвященного пользователя. Именно это мы постараемся исправить в следующий раз, рассмотрев несколько веб-интерфейсов, значительно упрощающих настройку Asterisk. Кроме того, постараемся более детально изучить «аппаратную» часть телефонных систем на основе Asterisk, а также расширить функциональность созданной сегодня системы. 

Литература и ссылки:

1. <http://www.asterisk.org>.
2. <http://www.asterisk.org/index.php?menu=features>.
3. <http://www.automated.it/asterisk>.
4. Шергин Д. Установка IMAP4-сервера на базе Cyrus-imapd + sendmail. – Журнал «Системный администратор», №10, октябрь, 2003 г. – 10-14 с.
5. Бешков А. Архитектура Postfix. – Журнал «Системный администратор», №6, июнь, 2004 г. – 4-8 с.
6. Стахов В. Почтовая система на базе MTA Exim. – Журнал «Системный администратор», №7, июль, 2004 г. – 6-13 с.
7. <http://www.tario.net>.
8. <http://www.voip-info.org>.
9. <http://www.asterisk-support.ru>.
10. <http://lists.digium.com>.
11. Платов М. Что важно знать об IP-телефонии. – Журнал «Системный администратор», №5, май, 2005 г. – 20-25 с.