

IT база знаний
wiki.merionet.ru

Бесплатный курс по Asterisk

УСТАНОВКА, НАСТРОЙКА, БЕЗОПАСНОСТЬ, ТРАБЛШУТИНГ



МЕРИОН НЕТВОРКС

Блок 2.

Установка и настройка Asterisk через консоль (CLI)

Обзор

1. [Видео - установка Asterisk из консоли](#)
2. [Asterisk - настройка с нуля через CLI](#)
3. [Домашнее задание.](#)

Видео - установка Asterisk из консоли

Посмотрите видео о том, как установить Asterisk 14.6.0 на операционную систему CentOS 7. Из консоли. С нуля.



Материал по [установке Asterisk 14.6.0 на CentOS 7](#) также можно прочитать в статье (включая нужные команды)

Настройка с нуля через консоль

Сделаем базовую настройку Asterisk через конфигурационные файлы. Предполагается, что у нас уже установлена и первоначально настроена ОС, скачены и установлены модули dahdi, libpri, iax2, необходимые голосовые файлы и кодеки и проинсталлирован Asterisk.

А еще вам понадобится [установить sngrep](#) для трассировки и отладки SIP-сообщений.

Теория

Итак, приступаем к внедрению Asterisk. Структура используемых Астериском директорий следующая:

- **/usr/lib64/asterisk/modules** – тут находятся загружаемые модули;
- **/var/log/asterisk** – тут находятся лог-файлы, в том числе и лог звонков (если не настроено другое);
- **/var/spool/asterisk** – тут находятся подпапки, в которых находятся бэкапы, записи разговоров, голосовая почта, факсы и так далее;
- **/var/lib/asterisk** – тут находятся подпапки, в которых находятся звуковые файлы для музыки на удержании, звуковые файлы для выбранных языков (например для проигрывания голосовых сообщений в IVR), записанные голосовые сообщения для приветствия и так далее.

Конфигурационные файлы находятся в папке /etc/asterisk. Для работы каждого модуля Asterisk необходим конфигурационный файл. Эти файлы (с расширением .conf), содержат определения каналов, описывают различные внутренние сервисы, определяют местоположения других модулей, устанавливают связь с диалпланом. Необязательно настраивать все файлы. Требуют настройки только те, которые необходимы для вашей конфигурации.

Основные конфигурационные файлы:

- **asterisk.conf** – определяет глобальные параметры, директории и опции для запуска Asterisk;
- **cdr.conf** – определяет настройки для записи параметров вызовов в файл или базу данных;
- **sip.conf** – определяет настройки для использования SIP-протокола (как общие, так и параметры для регистрации провайдеров, внутренних пользователей и так далее);
- **rtp.conf** – определяет порты для голоса (RTP);
- **iax.conf** – определяет настройки для использования IAX-протокола (как общие, так и параметры для регистрации провайдеров, внутренних пользователей и так далее);
- **extensions.conf** – основной файл, в котором описывается весь диалплан, то есть правила обработки всех вызовов;
- **features.conf** – описывает дополнительные функции (переадресации, парковка вызова, включение записи по запросу и так далее);
- **logger.conf** – определяет тип и детальность сообщений, записываемых в файлы журналов;
- **modules.conf** – определяет какие модули будут или наоборот не будут загружаться при запуске Asterisk;
- **musiconhold.conf** – используется для конфигурации разных классов музыки, используемых в приложениях музыки во время ожидания, и их местоположений;

Напомним, что это только часть конфигурационных файлов. Необходимые файлы можно добавлять в любой момент по мере необходимости. Примеры и содержание таких файлов можно найти в архиве по кнопке ниже:

[Скачать архив](#)

Сразу после установки asterisk, если не была выбрана установка базовой конфигурации, в ней нет ни одного файла.

Имя	Размер	Время правки
..	-ВВЕРХ-	Фев 1 11:29

Для подключения к asterisk в режиме командной строки необходимо ввести

```
asterisk -rvvvvv
```

- **r** – подключение к уже запущенному процессу;
- **vvvvv** – уровень логирования, то есть вывода информации (от слова verbose - v).
Чем больше v выставляем, тем более детальная информация будет выдаваться в командную строку;

```
[root@mcc-kama-02 ~]#
[root@mcc-kama-02 ~]# asterisk -rvvvvvvv
Unable to open specified master config file '/etc/asterisk/asterisk.conf', using built-in defaults
Asterisk 15.7.1, Copyright (C) 1999 - 2016, Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 15.7.1 currently running on mcc-kama-02 (pid = 5654)
mcc-kama-02*CLI>
mcc-kama-02*CLI>
```

Создаем и редактируем необходимые файлы

Начнем с файла **asterisk.conf**:

```
asterisk.conf [----] 18 L: [ 1+18 1
[directories](!)
astetcdir => /etc/asterisk
astmoddir => /usr/lib64/asterisk/modules
astvarlibdir => /var/lib/asterisk
astdbdir => /var/lib/asterisk
astkeydir => /var/lib/asterisk
astdatadir => /var/lib/asterisk
astagidir => /var/lib/asterisk/agi-bin
astspooldir => /var/spool/asterisk
astrundir => /var/run/asterisk
astlogdir => /var/log/asterisk
astbindir => /usr/sbin

[options]
transmit_silence_during_record = yes
languageprefix = yes
execincludes = yes
hideconnect = yes
dontwarn = yes
; verbose=3
debug = no
maxcalls = 10
```

- **[directories](!)** – указываем расположение необходимых директорий. Знаком (!) указывается признак шаблона. В шаблоне указываются общие настройки, на которые можно ссылаться дальше.
- **[options]** – указываем необходимые опции, одна из необходимых **maxcalls** указывает на количество одновременных вызовов, разрешенных на Asterisk;
 - **transmit_silence_during_record = yes** - передавать тишину SLINEAR во время записи канала;

- **languageprefix = yes | no** - Должен ли код языка быть последним или первым компонентом имени звукового файла? Если выключен, поиск звуковых файлов ведется в формате <путь>/<язык>/<файл> Если включен, поиск ведется в формате <язык>/<путь>/<файл>;
- **execincludes = yes | no** - Разрешить записи #exes в конфигурационных файлах;
- **hideconnect = yes | no** - Показывать сообщение о подключении удаленных консолей;
- **dontwarn = yes | no** - Отключить предупреждения (warning messages);
- **debug = no** - Отладка: No или значение (1-4);
- **maxcalls = 10** - Максимальное число одновременных вызовов;

Приступаем к файлу **cdr.conf**. Комментарии к опциям в конфиге:

```
[general]

enable=yes           ; включаем саму возможность
логирования звонков
unanswered=no        ; неотвеченные звонки не
логируем
safeshutdown=yes     ; при выключении сервера будем
ждать, пока не допишутся ВСЕ логи

[csv]

usegmttime=yes       ; лог date/time в формате GMT. По умолчанию
NO
loguniqueid=yes
loguserfield=yes
```

Закончили. Теперь файл **features.conf**:

```
[general]

[featuremap]         ; тут описываем
используемые функции и их параметры

blindxfer => ##       ; безусловный перевод
atxfer => *2           ; условный перевод
automon => *1
disconnect => **
parkext => 700         ; парковка
```

```

parkpos => 710-780 ; диапазон портов для
парковки
context => parkedcalls ; контекст для обработки
запаркованных звонков
parkingtime => 180 ; время парковки
comebacktoorigin => no ; возвращать звонок на
инициатора, когда закончилось время
парковки вызова
parkedplay => both ; кому играть courtesytone когда
вызов снимается с парковки. Опции: callee, caller,
both или no(по умолчанию)

parkedcalltransfers => caller ; Кто может сделать
трансфер припаркованного вызова с помощью
DTMF. Опции: callee, caller, both или no(по умолчанию)
parkedcallrepark => caller ; Кто может перепарковать,
припаркованный вызов с помощью DTMF. Опции:
callee, caller, both или no(по умолчанию)
parkedcallhangup => no ; Кто может закончить,
припаркованный вызов с помощью DTMF Опции:
callee, caller, both или no(по умолчанию)
parkedcallrecording => no ; Кто может инициировать
запись, с помощью DTMF. Опции: callee, caller, both или
no(по умолчанию)..
parkedmusicclass => default ; Класс музыки ожидания для
припаркованного,
adsipark => no ; Передавать или нет ADSI инфо о
припаркованном вызове тому кто припарковал
findslot => first
pickupexten => *8 ; перехват звонка

[applicationmap] ; тут описываем используемые
приложения

sendsms => *99,peer/both,Macro,sendsms
pitch => *00,self/both,Macro,pitch

```

Теперь конфигурируем RTP в файле **rtp.conf**

```

[general]
rtpstart=36600
rtpend=39999

```

Музыка на ожидании в здании. Открываем файл **musiconhold.conf**

```
[default]
mode=files
directory=/var/lib/asterisk/moh/
```

Следом открываем файл **logger.conf**:

```
[general]
[logfiles]
console => notice,warning,error,dtmf,verbose(5) ; уровень
детализации сообщений, выводимых в консоль
full => debug,notice,warning,error,verbose(9),dtmf,fax,security ;
уровень детализации сообщений, выводимых в
лог-файл
```

Для удобства работы рекомендуется ограничивать уровень детализации сообщений, выводимых в консоль, но для вывода в файл выставить максимальный уровень детализации.

И напоследок - файл **modules.conf**. Есть 2 варианта: либо читаем все модули и указываем те, которые не надо читать:

```
[modules]
autoload=yes
noload => codec_g723-ast110-gcc4-glibc-x86_64-core2-sse4.so
```

Либо указываем конкретные модули, которые необходимо прочитать и запрещаем чтение всех. В этом случае для удобства лучше поделить модули на секции. Ниже приведена часть такого варианта:

```
[modules]
autoload = no

; Applications

load = app_bridgewart.so
load = app_dial.so
load = app_playback.so
```



```
; Bridging

load = bridge_builtin_features.so
load = bridge_builtin_interval_features.so
load = bridge_holding.so

; Call Detail Records

load = cdr_custom.so

; Channel Drivers

load = chan_bridge_media.so
load = chan_sip.so

; Codecs

load = codec_gsm.so
load = codec_ulaw.so
load = codec_alaw.so
load = codec_g722.so

; Formats

load = format_gsm.so
load = format_pcm.so
load = format_wav_gsm.so
load = format_wav.so

; Functions

load = func_callerid.so
load = func_cdr.so
load = func_pjsip_endpoint.so

; Core/PBX

load = pbx_config.so

; Resources
```

```
load = res_musiconhold.so
load = res_pjproject.so
load = res_pjsip_acl.so
```

В данном разделе мы используем первый вариант. На этом с настройкой основных файлов закончим. В дальнейшем по мере необходимости в них можно вносить изменения. Все последующие настройки мы будем вносить в файлы sip.conf и extensions.conf. Погнали к созданию и регистрации внутренних абонентов.

Создание и регистрация внутренних абонентов

В sip.conf указываем сначала общие параметры SIP для Asterisk:

```
[general]
bindaddr=0.0.0.0      ; указываем IP-адрес и порт, на
                        ; котором будет приниматься SIP-трафик
bindport=5060
language=ru           ; используемый язык для голосовых
                        ; сообщений
alwaysauthreject=yes
allowguest=no          ; запрещаем принимать
                        ; «гостевые» звонки, то есть вызовы от
                        ; незарегистрированных пользователей
```

Так же в этой секции можно указать поддерживается ли видео, время регистрации, перечислить локальные сети, указываем внешний IP-адрес в случае использования NAT и так далее.

В случае, когда у нас есть разные группы абонентов (например, есть несколько отделов, подразделений либо другие какие-то признаки группировки абонентов или абонентов большое количество), рекомендуется использовать шаблоны, в которые можно выносить обобщенные настройки.

Имя шаблона берется в скобки [] и следом указывается (!). В шаблоне можно указать контекст для этих абонентов, используемые кодеки, разрешенные/запрещенные сети для регистрации этих абонентов, использование NAT и так далее. Пример шаблона приведен ниже:

```
[office](!)

type=friend
deny=0.0.0.0/0.0.0.0
permit=192.168.10.0/255.255.255.0
host=dynamic
context=from-internal
nat=no
qualify=yes
directmedia=no
disallow=all
allow=alaw
allow=ulaw
dtmfmode=info
```

И таких шаблонов можно может быть несколько. Теперь для создания записи для регистрации абонентов нам достаточно указать только отличительные параметры, такие как внутренний номер, имя абонента, пароль для регистрации и так далее.

```
[HOMEP](ШАБЛОН)
callerid=ИМЯ <HOMEP>
secret=ПАРОЛЬ
callgroup=5 ; номер группы вызова
pickupgroup=1,2,3,4,5 ; номера групп перехвата
ВЫЗОВОВ
```

Пример настройки:

```
[office](!)
type=friend
secret=AsterPASS
deny=0.0.0.0/0.0.0.0
permit=10.26.24.192/255.255.255.192
host=dynamic
context=from-internal
nat=no
qualify=yes
directmedia=no
disallow=all
allow=alaw
allow=ulaw
dtmfmode=info
```

```
[1001](office)
nat=comedia
callerid=Alexey Fedorov <1001>
secret=qazQAZ!123
callgroup=5
pickupgroup=1,2,3

[1002](office)
callerid=Mikhail Ivanov <1002>
secret=AsterPASS
callgroup=1
pickupgroup=1,2,

[1003](office)
callerid=Yuriy Yakovlev <1003>
secret=Qwerty!123
callgroup=1
pickupgroup=1,2,3
```

В результате, по команде **sip show peers** мы видим зарегистрированных пользователей

```
mcc-kama-02*CLI> sip show peers
Name/username      Host              Dyn Forcerport Comedia  ACL Port  Status      Description
1001/1001           10.26.24.212     D No      Yes      A 5357    OK (5 ms)
1002/1002           10.26.24.212     D No      No       A 5302    OK (5 ms)
1003/1003           10.26.24.209     D No      No       A 5060    OK (10 ms)
```

Абоненты зарегистрировались, но позвонить даже между собой они пока не могут. Для того, чтобы они могли совершать и принимать звонки необходимо настроить маршрутизацию (или диалплан). Делать это мы будем в файле `extensions.conf`, там тоже есть своя структура. И тут мы снова немного погружаемся в теорию:

Диалплан состоит из следующих основных элементов:

- **контексты;**
- **добавочные номера;**
- **приоритеты;**
- **приложения;**

Контекст – часть (раздел) диалплана, описывающая алгоритм обработки вызова и изолированная от остального диалплана. Содержит дополнительные номера (extension). Дополнительные номера, определенные в одном контексте, полностью изолированы от добавочных номеров в другом контексте, если это не разрешено специально. Так же с помощью контекстов можно ограничивать доступ к различным функциям (например к междугородним или международным звонкам). Имя контекста заключается в квадратные скобки []. Рекомендуется создавать разные контексты для внутренних абонентов и для транков.

В начале диалплана находятся два специальных контекста, `[general]` и `[globals]`

- **[general]** – содержит список общих настроек диалплана;

- **[globals]** – содержит глобальные переменные;

Эти два контекста являются специальными. Контекст является одним из обязательных параметров как для абонента, так и для транка.

Asterisk определяет контекст для обработки по тому принципу откуда пришел вызов, а не куда он пришел, то есть если пришел вызов на мобильный номер от абонента, то применяться будет тот контекст, который прописан у конкретного абонента, а не указанный в транке.

Добавочные номера – это широкое понятие, которое определяет уникальные последовательности шагов (каждый шаг включает приложение), которые Asterisk будет применять к вызову по этой линии. В каждом контексте может быть задано столько добавочных номеров, сколько требуется. При вызове конкретного добавочного номера (входящим или внутренним звонком) Asterisk будет выполнять шаги, определенные для этого добавочного номера. Поэтому именно добавочные номера определяют, что происходит со звонками при их обработке соответственно диалплану.

Полный добавочный номер состоит из трех компонентов:

- **Имени (или номера).** В качестве имени может быть использованы любые комбинации цифр и букв;
- **Приоритета** (каждый добавочный номер может включать множество шагов; порядковый номер шага называется его приоритетом);
- **Приложения** (или команды), которое выполняет некоторое действие над вызовом;

Эти три компонента разделяются запятыми:

```
exten => имя , п р и о р и т е т , п р и л о ж е н и е ( )
```

Есть ещё зарезервированные добавочные номера:

- **s** - когда в контекст поступают вызовы, для которых не указан конкретный добавочный номер, они передаются на добавочный номер s. (s - сокращение от start (начало), поскольку именно здесь начнется обработка вызова, если не передана информация о добавочном номере.;
- **i** - когда абонент нажимает не ту кнопку (не существующий добавочный номер), вызов направляется на добавочный номер i;

- **t** - если абонент слишком долго не нажимает кнопку после запуска приложения WaitExten(), вызовы направляются на добавочный номер t (время ожидания по умолчанию - 10 с);
- **h** - экстеншен обрабатываемый при завершении вызова. После того как медиаканал закрылся;

Иногда можно встретить использование same вместо exten. Это применяют в основном с автоматическим выставлением приоритета, то есть same => n и означает «тоже самое, продолжение предыдущего»

Приоритеты – последовательность выполнения приложений. Каждый приоритет пронумерован последовательно, начиная с 1, и выполняет одно определенное приложение. В Asterisk есть еще приоритет n, что означает «следующий». Каждый раз, когда Asterisk встречает приоритет n, она берет номер предыдущего приоритета и добавляет 1. Это упрощает внесение изменений в диалплан, поскольку теперь не надо изменять номера всех шагов.

Приложения – выполняет определенное действие в конкретном дополнительном номере (например воспроизведение звука, прием тонального ввода, вызов канала, разрыв соединения и так далее).

Для выполнения некоторых приложений, таких как Answer() и Hangup(), не требуется никаких дополнительных инструкций. Некоторым приложениям необходима дополнительная информация. Эти данные, называемые аргументами, могут передаваться в приложения, чтобы оказывать влияние на то, как они выполняют свои действия. Чтобы передать аргументы в приложение их указывают через запятую в круглых скобках, следующих за именем приложения.

Для внесения комментариев в файл extensions.conf используют ; - вы уже могли об этом догадаться, судя по нашим комментариям прямо в конфигах :)

Таким образом можно как делать пометки для себя, так и делать невыполнимыми строки конфигурации (например, во время отладки)

Теперь давайте вернемся к нашим созданным абонентам. Создадим контекст, который указан у абонентов (**context=from-internal**)

```
[from-internal]
exten => 1001,1,Dial(SIP/1001),tTm
exten => 1002,1,Dial(SIP/1002),tTm
exten => 1003,1,Dial(SIP/1003),tTm
```

В нем мы прописали что при наборе номера (ИМЯ), с приоритетом 1 выполнить приложение Dial с параметрами ПРОТОКОЛ/НОМЕР. Когда номеров немного, то можно конечно и так описывать. Но более правильно и красиво сделать тоже самое, но с использованием «маски»:

```
exten => _10xx,1,Dial(SIP/${EXTEN},60,tTm)
```

То есть при наборе любого номера из диапазона 10xx (шаблон показан нижним подчеркиванием) выполнить вызов с приоритетом 1 через приложение Dial с параметрами ПРОТОКОЛ/НАБРАННЫЙ_НОМЕР, время вызова 60 секунд и можно использовать перевод звонка (transfer).

Шаблон номера - это уникальный набор цифр, который определяет использование этого номера. Если набранный номер соответствует этому шаблону, то последующие номера не рассматриваются. Формат заполнения шаблона:

- **X** - совпадение любой цифры от 0 - 9;
- **Z** - любая цифра от 1 до 9;
- **N** - совпадение любой цифры от 2 - 9;
- **[1237-9]** - соответствует любым цифрам или буквам и скобках (в этом примере,1,2,3,7,8,9);

Перечитываем диалплан в консоли Asterisk командой dialplan reload и видим выполнение вызова.

```
mcc-kama-02*CLI>
== Using SIP RTP CoS mark 5
> 0x7f4acc010120 -- Strict RTP learning after remote address set to: 10.26.24.212:10138
-- Executing [1003@from-internal:1] Dial("SIP/1001-00000006", "SIP/1003,60,tTm") in new stack
== Using SIP RTP CoS mark 5
-- Called SIP/1003
-- Started music on hold, class 'default', on channel 'SIP/1001-00000006'
-- SIP/1003-00000007 is ringing
> 0x7f4acc010120 -- Strict RTP switching to RTP target address 10.26.24.212:10138 as source
> 0x7f4acc010120 -- Strict RTP learning complete - Locking on source address 10.26.24.212:10138
-- Stopped music on hold on SIP/1001-00000006
== Spawn extension (from-internal, 1003, 1) exited non-zero on 'SIP/1001-00000006'
mcc-kama-02*CLI>
```

Таким образом мы можем придумать и реализовать практически любой диалплан.

Например для запрета вызовов на международную связь достаточно прописать 3 строчки:

```
; запрет набора международной связи с проигрываем сообщения
exten => _810.,1,Answer()
exten => _810.,n,Playback(destination-closed)
exten => _810.,n,Hangup()
```

То есть при наборе 810 будет проиграно сообщение destination-closed (если оно было загружено в Asterisk) и будет отправлен сигнал отбоя.

Создание и регистрация транков

Ну, начнем с того, что IP-транки, используемые в Asterisk, бывают 2-х видов – SIP и IAX.

SIP-транки в основном используются для подключения провайдеров, а IAX-транки для подключения других Asterisk. Транки могут быть с регистрацией (то есть когда провайдер выдает логин, пароль и адрес или домен для регистрации у него) и без регистрации (то есть когда подключение идет по IP-адресу без логина и пароля). В случае с регистрацией в файле sip.conf необходимо сразу после секции [general] указать строку регистрации в формате:

```
register => ЛОГИН:ПАРОЛЬ@SIP-ПРОВАЙДЕР/НОМЕР
```

Тут:

- **SIP-ПРОВАЙДЕР** - указывается или IP-адрес провайдера или его домен;
- **ЛОГИН:ПАРОЛЬ** - выдаются провайдером для подключения;
- **НОМЕР** - указывается городской номер, выданный провайдером для совершения звонков;

Рассмотрим создание SIP-транка с регистрацией. Опять же если у нас несколько (до 3-5) таких транков, то можно их описать каждый отдельно. А если их больше или в дальнейшем планируется увеличить их количество, то можно использовать шаблон для подключения к оператору.

```
[voip-provider](!)           ; имя шаблона

type=peer                    ; тип подключения
context=from-trunk           ; используемый контекст для
о б р а б о т к и   в ы з о в о в
disallow=all                  ; выключаем все кодеки
allow=alaw                    ; указываем используемые
к о д е к и
allow=ulaw
insecure=invite,port          ; не запрашивать
а в т о р и з а ц и ю   н а   в х о д я щ и е   з в о н к и
qualify=yes                   ; проверка доступности
directmedia=no                ; запрещаем установление
п р я м ы х   с о е д и н е н и й   д л я   п е р е д а ч
```


Г о л о с а

```
dtmfmode=rfc2833 ; указываем используемый тип DTMF
```

Дальше достаточно описать конкретные настройки для конкретного оператора и указать какой шаблон использовать

```
[О П Е Р А Т О Р -1](voip-provider)
defaultuser=ЛО Г И Н -1
fromuser=ЛО Г И Н -1
secret=ПА РО Л Ь -1
host=ДОМЕН1- ИЛИ IP-АДРЕС ДЛЯ ПОДКЛЮЧЕНИЯ
fromdomain= ДОМЕН-1 ИЛИ IP-АДРЕС ДЛЯ ПОДКЛЮЧЕНИЯ
```

```
[О П Е Р А Т О Р -2](voip-provider)
defaultuser=ЛО Г И Н -2
fromuser=ЛО Г И Н -2
secret=ПА РО Л Ь -2
host=ДОМЕН-2 ИЛИ IP-АДРЕС ДЛЯ ПОДКЛЮЧЕНИЯ
fromdomain= ДОМЕН ИЛИ IP-АДРЕС ДЛЯ ПОДКЛЮЧЕНИЯ
```

Дальше указываем строки для регистрации у данных операторов:

```
register => ЛО Г И Н -1:ПА РО Л Ь -1@ДОМЕН-1/НОМЕР-1
register => ЛО Г И Н -2:ПА РО Л Ь -2@ДОМЕН-2/НОМЕР-2
```

Перечитываем файл **sip.conf** и проверяем регистрации:

```
mcc-kama-02@CLI> sip show peers
Name/username      Host
1001/1001           10.26.24.212
1002/1002           10.26.24.199
1003/1003           10.26.24.209
1004                (Unspecified)
1005                (Unspecified)
1006                (Unspecified)
1010                (Unspecified)
cc/749             /749
m/10               /10
Dyn Forcerport Comedia ACL Port Status Description
D No Yes A 5357 OK (5 ms)
D No No A 5060 OK (3 ms)
D No No A 5060 OK (8 ms)
D No No A 0 UNKNOWN
D No No A 0 UNKNOWN
D No No A 0 UNKNOWN
D No No A 0 UNKNOWN
Auto (No) No 5060 OK (6 ms)
Auto (No) No 5060 OK (4 ms)
```

В случае подключения транка без регистрации можно использовать тот же шаблон, а в настройках транка указать изменяемые параметры

```
[AST10SIP](voip-provider)
type=friend ; для транка без регистрации
указываем friend (то есть мы доверяем этому
подключению)
```

```
port=5060 ; указываем порт для подключения
insecure=port,invite
host=IP-АДРЕС_ПРОВАЙДЕРА
context=from-trunk-sip-AST10SIP ; если обработки
вызовов через этот транк используется
другой контекст, то указываем его тут.
```

Перечитываем файл **sip.conf** и проверяем регистрации:

```
AST10SIP 10.25 No No 5060 OK (1 ms)
```

Теперь рассмотрим создание IAX-транка. Для настройки IAX-транков используется файл **iax.conf**, который содержит всю информацию, необходимую Asterisk для создания и управления каналами, работающими по протоколу IAX. Структура его примерно такая же, как и у **sip.conf**:

```
[general] ; указываем глобальные параметры
для протокола IAX

bindaddr=0.0.0.0
bindport=4569 ; по-умолчанию IAX-протокол
использует порт 4569 можно оставить его, а
можно и переопределить
language=ru

; указываем строки для регистрации транков
register => msk-spb:SuperPASS@10.10.10.10

[msk-spb]

username = msk-spb ; логин для регистрации на
удаленной стороне
type = friend
trunk = yes
secret = SuperPASS ; пароль для регистрации
qualify = yes
host = 10.10.10.10 ; IP-адрес удаленной стороны
disallow= all
context = from-iax ; контекст для обработки
вызовов, поступающих через этот транк
allow = alaw
allow = ulaw
```

Сохраняем файл **iax.conf**, перечитываем и проверяем регистрацию командой **iax2 show peers**:

```
ippbxstand01*CLI> iax2 show peers
Name/Username      Host      Mask      Port      Status      Description
AST101IAX/AST101  10.25.    (s) 255.255.255.255  4569 (T)    OK (3 ms)
```

Если есть абоненты, работающие по протоколу IAX, то их регистрацию описываем тоже в этом же файле аналогично SIP-регистрации.

Итак, сейчас мы имеем зарегистрированных абонентов, которые могут звонить друг другу, и зарегистрированные транки. Внутренних абонентов мы можем группировать по отделам:

```
exten => 500,1,Playback(it-otdel)      ; проигрывается
сообщение it-otdel
exten => 500,1,Dial(SIP/1001,5),Tt      ; 5 секунд вызов идет
на номер 1001
exten => 500,n,Dial(SIP/1002&SIP/1003) ; потом вызов идет
одновременно на 1002 и 1003
```

Можем настраивать различные функции, запускать различные команды (в том числе и для выполнения через ОС), настраивать запись и прослушивание разговоров и так далее:

```
; ответить, подождать 2 секунды и положить
трубку
exten => 060,1,Answer()
same => n,Wait(2)
same => n,Hangup()
; ответить, проиграть сообщение hello-world и
положить трубку
exten => 061,1,Answer()
same => n,Playback(hello-world)
same => n,Hangup()
; записать сообщение в файл somefile.gsm и потом
его проиграть
exten => 067,1,Record(/tmp/somefile.gsm,3,30)
same => n,Playback(/tmp/somefile)
```

Для совершения звонков через созданные и зарегистрированные транки SIP и IAX: Допустим через транк IAX у нас подключен другой Asterisk с внутренней нумерацией, начинающейся с 1, 2, 3. И для вызова этих абонентов мы будем использовать префикс (код выхода на маршрут) 2. Тогда строки настройки будут следующие:

```
exten => _2[1-3].,1,Dial(IAX2/msk-spb/${EXTEN:1},30,r)
exten => _2.,2,Hangup()
```

То есть при наборе, начинающемся с 21-23, будет осуществлен вызов через транк msk-spb по протоколу IAX набранного номера, предварительно «отрезав» 1 (первую) набранную цифру. Если в течение 30 секунд не будет получен ответ, то вызов будет прекращен.

Для выхода в город мы используем транк с оператором-1 и префикс выхода будем использовать 9

```
exten => _9849[589]XXXXXXX,1,Dial(SIP/О П Е Р А Т О Р -1/${EXTEN:1})
; то есть при наборе, начинающемся с 9, будет
осуществлен вызов через транк ОПЕРАТОР-1 по
протоколу SIP набранного номера,
предварительно «отрезав» 1 (первую) набранную
цифру
```

Тут важно понимать, что все, что мы реализовываем для внутренних абонентов, должно быть описано в соответствующем контексте.

Теперь перейдем к транкам и входящим звонкам. Соответственно для того, чтобы принимать входящие вызовы, необходимо прописать маршрутизацию уже в контексте транка (**context=from-trunk** или **context = from-iax**)

Для возможности через транк осуществлять вызов нашего внутреннего абонента (например через транк со встречной АТС) необходимо в контекст транка вставить **exten => _10XX,1,Dial(SIP/\${EXTEN},60,tTm)**

Давайте рассмотрим реализацию обработки входящего вызова от оператора (вызов на городской номер) через создание меню IVR и реализуем ещё определение рабочего и нерабочего времени. Схема обработки входящего вызова следующая:

Рабочее время у нас определено с 9:00 до 19:00 и с понедельника по пятницу. При поступлении звонка в нерабочее время после сообщения с приветствием (**01-hello**) проигрывается сообщение с указанием рабочего времени (**07-working-hours**). При поступлении звонка в рабочее время (проверка осуществляется в строке `GoToIfTime(09:00-19:00,mon-fri)`) после приветствия осуществляется переход в другой контекст (**[working-time]**), где предлагается выбрать необходимый пункт меню (0 – вызов секретаря, 1 – вызов на группу тех. поддержки, 2 – переход в другое меню выбора (`GoTo(ivr-2,s,1)`)), в котором по такому же принципу осуществляется выбор. В каждом меню реализован донabor внутренних номеров (`exten => _1xxx,1,NoOp`), обработка

неправильного набора номера (exten => i,1,NoOp), обработка в случае, что если ничего не выбрали (exten => t,1,NoOp), вызов переводится на секретаря.

Естественно необходимо загрузить все используемые голосовые файлы в **/var/lib/asterisk/sound/ru** в случае использования русского языка. Тут давайте немного поподробнее.

Как мы уже указывали выше в системе мы определили какой основной язык у нас будет использоваться для голосовых файлов (в файле sip.conf параметр language = ru). Это значит, что Asterisk будет искать имена файлов, которые мы указываем, например, в меню ivr в папке /var/lib/asterisk/sound/ru (смотрим обозначения директорий при запуске asterisk в начале раздела). Если бы мы использовали в качестве основного языка английский, то папка была бы /var/lib/asterisk/sound/en. В каждой из этих папок находятся голосовые файлы выбранных языков и в выбранных форматах, указанных при компилировании asterisk. Если мы хотим записать свои сообщения (персональные приветствия, необходимые объявления, произносимые в создаваемых меню ivr и так далее), нам необходимо положить эти файлы в папку с соответствующим языком. Сами файлы при этом можно записать любой звукозаписывающей программой (хоть программой Звукозапись, входящей в стандартный дистрибутив любой версии Windows) и сохранить в формате wav (**несжатый голос, 8кГц, 16 Бит, Моно**)

Тут главное не перепутать имена файлов, находящихся в папке с голосовыми сообщениями, с именами, указанными в ivr меню. при этом в ivr меню имена указываются без расширения. Сам листинг приведен ниже.

```
[from-trunk]
```

```
exten => _X.,1,NoOp(Проверка времени: Если попали в
диапазон - переходим в контекст working-time, если
нет - продолжаем выполнение)
same => n,Answer()
same => n,Playback(01-hello)
same => n,GoToIfTime(09:00-19:00,mon-fri,*,*?working-time,s,1)
same => n,Playback(07-working-hours)
same => n,Hangup()
```

```
[working-time]
```

```
exten => s,1,Answer()
same => n,Background(01-ivr1)
same => n,StartMusicOnHold()
```

```

same => n,WaitExten(5)
;
exten => 0,1,NoOp(Если нажали "0" - звоним секретарю)
same => n,Playbacr(ostavaite-na-linii)
same => n,Dial(SIP/1005,30,mtT)
same => n,Hangup()
;
exten => 1,1,NoOp(Если нажали "1" - звоним на группу
вызова: 1001+1002)
same => n,Playback(it-otdel)
same => n,Dial(SIP/1001&SIP/1002,30,mtT)
same => n,Hangup()
;
exten => 2,1,NoOp(Если нажали "2" - перенаправляем на
ivr-2)
same => n,GoTo(ivr-2,s,1)
;
exten => _1xxx,1,NoOp(Прямой набор внутренних
номеров)
same => n,Playback(ostavaite-na-linii)
same => n,Dial(SIP/${EXTEN}15,mtT)
same => n,Hangup()
;
exten => i,1,NoOp(Обработка ошибочного
набора:i=illegal)
same => n,Playback(oshibka)
same => n,Dial(SIP/1005,30,r)
;
exten => t,1,NoOp(В случае, если не дождались
нажатия)
same => n,Playback(ostavaite-na-linii)
same => n,Dial(SIP/1005,30,m)
;

[ivr-2]

exten => s,1,Background(02-ivr2)
same => n,StartMusicOnHold()
same => n,WaitExten(5)
;
exten => 1,1,NoOp(Если нажали "1" - звоним на 1001)

```

```

same => n,Dial(SIP/1001,30,mtT)
same => n,Hangup()
;
exten => 2,1,NoOp(Если нажали "2" - звоним на 1002)
same => n,Dial(SIP/1002,30,mtT)
same => n,Hangup()
;
exten => _1xxx,1,NoOp(Прямой набор внутренних
номеров)
same => n,Dial(SIP/${EXTEN}15,mtT)
same => n,Hangup()
;
exten => i,1,NoOp(Обработка ошибочного
набора:i=illegal)
same => n,Playback(oshibka)
same => n,Dial(SIP/1005,30,r)
;

```

После сохранения файла extensions.conf перечитываем диалплан в консоли (dialplan reload) и проверяем. На этом закончим с примерами.

Конфигурируя **Asterisk** через конфигурационные файлы, мы получаем возможность реализовать практически любую логику работы, проводить интеграции со сторонними сервисами, запускать и выполнять скрипты на уровне ОС и так далее.

Домашнее задание N°2

В окончание второго учебного блока, ваша задача заключается в следующем:

- a. Разверните на виртуальной машине на своем компьютере/сервере IP - ATC Asterisk.
Посчитать технические требования к виртуальной машине вы можете с помощью нашего калькулятора инсталляции (Урок N°1).
- b. Создайте и зарегистрируйте ([MicroSIP](#)) 2 внутренних номера
- c. Подключите SIP - транк у провайдера

Совет. Вы можете купить не дорогой номер у любого провайдера, либо можете придумать тестовые авторизационные параметры.

- d. Совершите внутренний вызов между двумя софтфонами;
- e. Совершите вызов в город (если у вас есть настоящие параметры авторизации у провайдера)