آموزش Dialplan در استریسک

VoiPing.ir

مولف : محمد یزدی‌زاده

عناوین این مجموعه آموزشی

- معرفی ابزارها و برنامه‌های استفاده شده در این آموزش
- معرفی فایل‌های داخل پکیج آموزشی
- مفهوم توسعه در استریسک
- معرفی ابزارهای توسعه در استریسک
- مفهوم Channel و بررسی انواع آن در استریسک
- تشریح فایل extensions.conf , تعریف Dialplan , Context
- آشنایی با : Extension , Priority , n , Label , same , Comment
- فراخوانی Context ها و File ها در یکدیگر :
  - فراخوانی یک Context در Context ای دیگر
  - فراخوانی زمانبندی شده‌ی Conetxt ها
  - روش‌های فراخوانی یک فایل در فایلی دیگر
- تمرین Dialplan نویسی (مرور عملی مباحث گذشته)
- آشنایی کلی با ساختار FreePBX و ایزوهای مبتنی بر آن
- بررسی فرایند ساخت یک داخلی
  - در استریسک
  - در ایزوهای مبتنی بر FreePBX
- بررسی ارتباط Channel و Context
  - در استریسک
  - در ایزوهای مبتنی بر FreePBX
- تمرین Dialplan نویسی (مرور عملی مباحث گذشته)
- بررسی اپلیکیشن‌های Answer , Playback , Hangup
- دیباگ کدهای Dialplan و آشنایی مختصر با محیط CLI در استریسک
- بررسی اپلیکیشن‌های Background , WaitExten , Goto
- بررسی اپلیکیشن‌های Verbose , NoOp
- بررسی انواع متغییرها در Dialplan

VOIPING

عناوین این مجموعه آموزشی

- بررسی اپلیکیشن Dial
- سناریوی اول : برنامه‌ای هوشمند برای گرفتن موجودی و یا آخرین گردش‌ها از حساب بانکی
- اپلیکیشن Dial در CLI استریسک
- کار با Local Channel ها در استریسک
- آشنایی با Function ها در Dialplan
- فارسی سازی فایل‌ها و تغییر زبان پیش‌فرض استریسک
- بررسی اپلیکیشن‌های SayNumber , SayDigits , SayAlpha , SayPhonetic
- بررسی اپلیکیشن‌های Authenticate و DISA
- بررسی اپلیکیشن‌های System , Read
- بررسی Expression ها و کار با رشته های متنی در Dialplan
- بررسی اپلیکیشن‌های ExecIF, GotoIF , GotoIFTime
- سناریوی دوم : ساخت یک منشی دیجیتال یا AA
- سناریوی سوم : ساخت برنامه مدیر منشی
- بررسی Dial Patterns در استریسک
- سناریوی چهارم : توسعه برنامه مدیر منشی قبلی
- سناریوی پنجم : بستن 00 و موبایل برای برخی از داخلی‌ها
- بررسی اپلیکیشن ChanSpy
- سناریوی ششم : بهبود عمل شنود در Elastix , FreePBX , ISSABEL
- بررسی Extension های از پیش تعریف شده
- آشنایی با Macro و Subroutine ها
- آشنایی با مسیر های Dial در FreePBX و ایزوهای مبتنی بر آن

VOIPING

معرفی برنامه‌های استفاده شده در این آموزش

- برنامه WinSCP برای دسترسی آسان به فایل‌های سرور ویپ
- برنامه PuTTY برای اتصال به کنسول لینوکس و استریسک
- برنامه Notepad ++ برای ویرایش فایل‌ها و کدنویسی سناریوها
- برنامه Visual Studio Code برای ویرایش فایل‌ها و کدنویسی سناریوها
- سافت فون X-Lite برای ایجاد تماس
- سافت فون Zoiper برای ایجاد تماس

VOIPING

معرفی فایل‌های داخل پکیج آموزشی

- فایل‌های mp4 از محتویات دوره
- یک فایل Powr Point از محتویات دوره
- یک فایل PDF از محتویات دوره
- پوشه voiping_samples حاوی مثال‌های دوره
- پوشه voiping_scenarios حاوی تمامی فایل‌های مربوط به سناریوها
- پکیج فارسی‌سازی زبان و فایل‌های صوتی

مفهوم توسعه در استریسک

1. توسعه کدهای **C** برای ماژول‌های اصلی استریسک و اضافه کردن امکانات جدید که قبلا در استریسک نبوده است .
2. کد نویسی Dialplan و یا استفاده از API های استریسک به جهت ترکیب امکانات موجود در استریسک برای تولید سناریوهای جدید .

- در حالت اول نیاز به دانش زیاد و تخصصی ، در زمینه شناخت ماژول‌ها و هسته اصلی استریسک می‌باشد .
- برای 99.99 درصد سناریوهای لازم ، نیاز به توسعه در حالت اول نخواهد بود .
- از این به بعد منظور ما از توسعه استریسک به معنی کدنویسی Dialplan و یا استفاده از API های استریسک خواهد بود .
- به جرات می‌توان ادعا کرد ، با استفاده از ابزارهای توسعه در استریسک (APIs & Dialplan) ، هیچ سناریویی در مبحث ارتباطات وجود نخواهد داشت ، که با استریسک قابل پیاده سازی نباشد .

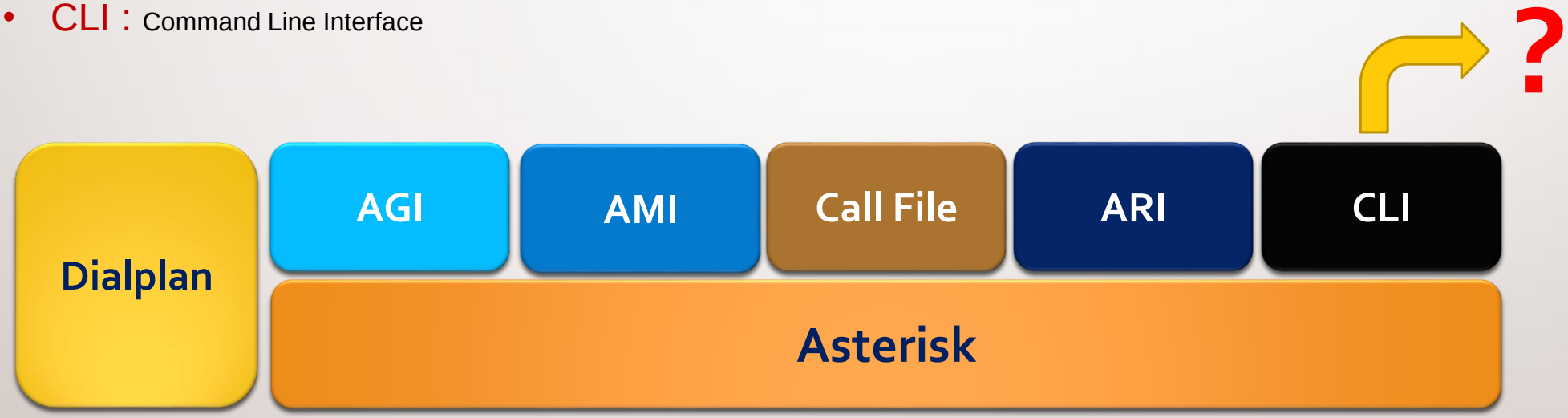
ما قرار نیست که چرخ و یا زنجیر را دوباره بسازیم ، استریسک هم چرخ دارد و هم زنجیر ، ما از چرخ و زنجیر استفاده خواهیم کرد تا دوچرخه بسازیم .

VOIPING

معرفی ابزارهای توسعه در استریسک

- Dialplan :
- AGI : Asterisk Gateway Interface
- AMI : Asterisk Manager Interface
- Call File
- ARI : Asterisk Rest-Full Interface
- CLI : Command Line Interface

Dialplan → extensions.conf → pbx_config.so
AEL → extensions.ael → pbx_ael.so
LUA → extensions.lua → pbx_lua.so



VOIPING

مفهوم Channel و بررسی انواع آن در استریسک

- یک Channel در واقع یک پل ارتباطی مابین استریسک و یک Device دیگر می‌باشد .
- Device دیگر می‌تواند هر کدام از موارد زیر باشد :
  - IP Phone ها و Soft Phone ها
  - گیتوی‌های آنالوگ (FXO , FXS)
  - گیتوی‌های دیجیتال (E1,T1)
  - کارت‌های آنالوگ (FXO , FXS)
  - کارت‌های دیجیتال (E1 , T1)
  - ATA ها
  - دیگر مراکز تلفن (PBX)
  - خود همان سرور استریسک
  - یک سیستم مبتنی بر استریسک دیگر
- در تعریف Channel نباید یک Channel را با یک تماس (Call) برابر دانست .
- یک تماس (Call) معمولا از اتصال دو (یا بیشتر) Channel تشکیل می‌شود .
- به اتصال دو یا چند Channel با هم Bridge گفته می‌شود .

VOIPING

مفهوم Channel و بررسی انواع آن در استریسک

مفهوم Channel و بررسی انواع آن در استریسک

- تقریبا در استریسک بدون Channel هیچ اتفاقی رخ نخواهد داد .
- در Channel ها صدا ، ویدئو ، و پیام‌های متنی قابلیت انتقال دارند .
- Channel ها در استریسک به چهار دلیل ایجاد می‌شوند .

1. زمانی که یک Device (تلفن) تقاضای صحبت با یک Device دیگری را به استریسک ارسال می‌کند . **Dialplan**
2. یک کاربر دستور ایجاد Channel در کنسول استریسک می‌دهد . (Originate , ...) **CLI**
3. یک Channel تقاضای ایجاد یک Channel دیگر را می‌دهد . (Dial , ...) **Dialplan**
4. استریسک دستوری از اینترفیس‌های توسعه خود (APIs) مبنی بر ایجاد یک یا چند Channel دریافت می‌کند . (Originate , CallFile , ...) **APIs**

- در استریسک مدیریت Channel ها با ابزارهای زیر می‌باشد .
  - **Dialplan**
  - **APIs (AGI,AMI,Call File,ARI)**
  - **CLI**

مفهوم Channel و بررسی انواع آن در استریسک

Channel های معروف

- SIP
- IAX2
- DAHDI
- LOCAL
- PJSIP

VOIPING

تشریح فایل extensions.conf و تعریف Dialplan

- اشتباه نکنید ، هیچ داخلی قرار نیست در این فایل تعریف شود .
- مسیردهی انواع Channel ها معمولا در این فایل مشخص می‌شود .
- دسترسی داخلی‌ها به مسیرهای خروجی و داخلی در سیستم تلفنی ، همگی در این فایل مشخص می‌شود .
- تمامی جریان‌های ارتباطی بین Device های مختلف در استریسک در این فایل مدیریت می‌شود .
- این فایل قلب سیستم تلفنی استریسک می‌باشد .

/etc/asterisk/extensions.conf

- به کدهای این فایل Dialplan گفته می‌شود .
- به کدنویسی در این فایل اصطلاحا Dialplan نویسی گفته می‌شود .
- اولین قدم در توسعه سناریوهای تلفنی در استریسک ، کد نویسی در همین فایل می‌باشد .

تعریف Context در استریسک

- کدهای Dialplan به یکسری سگمنت تقسیم‌بندی می‌شوند . به این سگمنت‌ها **Context** گفته می‌شود .
- Context ها واحدهای مستقل دسترسی و رفتاری برای Channel ها می‌باشند .
- Context ها با **[context-name]** شروع شده و کدهای زیر آن تا Context بعدی متعلق به آن Context می‌باشند .
- در نام‌گذاری Context ها می‌توان هم از اعداد و هم از حروف و یا ترکیب اعداد و حروف استفاده کرد و تا 79 کاراکتر این نام گذاری بلا مانع می‌باشد .

This is a Context

```
[voiping-context]
...
...
exten => 401,1,wait(30)
exten  =  401,2,dial(sip/401)
...
...
[another-context]
```

تعریف Context در استریسک

- در ابتدای فایل extensions.conf دو سگمنت ، به نام‌های general , globals وجود دارند ، که اولی حاوی تنظیمات کلی Dialplan و دومی محلی برای تعریف متغیرهای سراسری می‌باشد . این دو ، Context نیستند .
- در نام گذاری Context ها از نام‌های default , globals , general نباید استفاده کرد .

Extensions.conf :

```
[general]
...
[globals]
...
[context]
exten => extension_name , priority , asterisk_application
```

می‌توان از علامت **=** هم استفاده کرد

تعریف Extension در استریسک

- اشتباه نکنید ، در استریسک منظور از Extensoin ، داخلی نمی‌باشد .
- هر Context از لحاظ رفتاری به سگمنت‌هایی تقسیم‌بندی می‌شود که به آن‌ها Extension گفته می‌شود .
- هر Context می‌تواند یک یا چندین Extension در خود داشته باشد .
- Extension ها نام یک سری از دستورات به هم مرتبط و اولویت دار ، در Dialplan می‌باشند .
- در نام‌گذاری Extension ها می‌توان هم از اعداد و هم از حروف و یا ترکیب اعداد و حروف استفاده کرد .

```
[context]
exten => extension_name , priority , asterisk_application
```

[voiping-context]

This is an Extension

```
exten => 401,1,wait(5)
exten => 401,2,dial(sip/401)
...
```

This is an Extension

```
exten => voiping,1,wait(5)
...
```

مفهوم Priority , n در Dialplan

- ← **Priority**
  - ترتیب اجرای فرامین در یک Extension را مشخص می‌نماید .

```
[voiping-context]
...
exten => 401,1,wait(5)
exten => 401,2,verbose(this is for test)
exten => 401,3,noop(just noop)
exten => 401,4,dial(sip/401)
```

- ← **n**
  - برای جلوگیری از اشتباه اولویت گذاری در هنگام کد نویسی ، استفاده می‌شود .
  - استریسک اولویت‌ها را به صورت خودکار مشخص می‌نماید .
  - همیشه اولویت شماره ۱ ، باید صراحتا ذکر شود .

```
[voiping-context]
exten => 401,1,wait(30)
exten => 401,n,noop()
exten => 401,n,verbose(this is a test)
exten => 401,n,dial(sip/401)
```

VOIPING

مفهوم label , same در Dialplan

- **Label ←** به جهت نشان گذاری خطوط در یک Extension استفاده می‌شود .

```
[voiping-context]
...
exten => 401,n,wait(5)
exten => 401,n,verbose(this is for test)
exten => 401,n(voiping),noop()
exten => 401,n,dial(sip/401)
```

- **same ←** برای خلاصه‌نویسی در کدهای یک Extension استفاده می‌شود .

```
[voiping-context]
exten => 401,1,wait(5)
    same => n,noop(just noop)
    same => n,verbose(this is a test)
    same => n,dial(sip/401)
```

VOIPING

کامنت‌گذاری در Dialplan

- به جهت کامنت‌گذاری یک خط در کدهای Dialplan از کاراکتر **;** استفاده می‌شود .
- به جهت کامنت‌گذاری یک بلاک در کدهای Dialplan از ترکیب زیر استفاده می‌شود .

```
[voiping-context]
; this is a comment
exten => 401,1,noop(just noop)
;-- this lin is a comment
    also this one
    and this one --;
```

فراخوانی یک Context در Context ای دیگر

- با استفاده از عبارت **include =>** می‌توان یک Context را در Context ای دیگر فراخوانی کرد .
- اولویت اجرا اول از Context مادر شروع شده و در صورت نیافتن خط اجرایی مناسب Context های فرزند مورد بررسی قرار می‌گیرند .
- اولویت بررسی Context های فرزند ، به اولویت فراخوانی آن‌ها بستگی دارد .

```
[context1]
include => context2
exten => 401,1,wait(5)
exten => 401,2,dial(sip/401)

[context2]
exten => 301,1,verbose(this is a test)
```

=

```
[context1]
exten => 401,1,wait(5)
exten => 401,2,dial(sip/401)
include => context2

[context2]
exten => 301,1,verbose(this is a test)
```

VOIPING

فراخوانی زمانبندی شده‌ی Conetxt ها

- Context ها می‌توانند بر اساس زمان در یکدیگر فراخوانی شوند .

include => context, times, weekdays, mdays, months

[context1]
Include => context2,08:00-17:00,*,*,*
exten => 401,1,wait(5)
exten => 401,2,dial(sip/401)

[context2]
exten => 301,1,verbose(this is a test)

VOIPING

روش‌های فراخوانی یک فایل در فایلی دیگر

- با استفاده از چهار روش ، می‌توان یک File را در File ای دیگر به غیر از asterisk.conf و modules.conf فراخوانی کرد .

1. #include filename
2. #include /path/of/files/*.conf
3. #tryinclude filename
4. #exec program

- #exec به صورت پیش‌فرض فعال نمی‌باشد . برای فعال سازی این روش در سگمنت [option] در فایل asterisk.conf باید عبارت execincludes=yes را اضافه کرد .

extensions.conf :

```
[general]
[globals]
#include voiping.conf
[some-context]
...
```

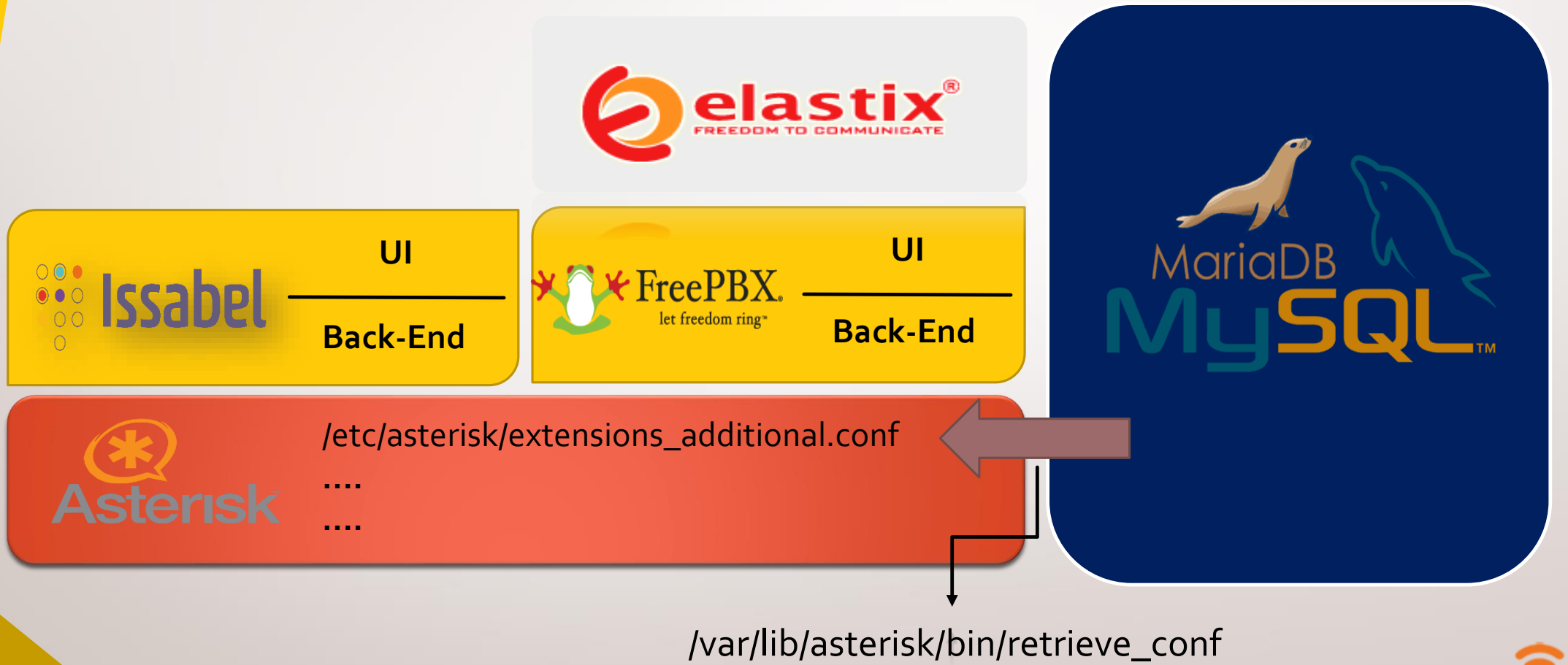
voiping.conf :

```
[voiping-context]
....
[another-context]
....
```

تمرین Dialplan نویسی (مرور عملی مباحث گذشته)

1. extensions.conf
2. context
3. extension
4. priority , n , label
5. same
6. comment in dialplan
7. include => context
8. #include filename
9. #tryinclude filename

آشنایی کلی با ساختار FreePBX و ایزوهای مبتنی بر آن



VOIPING

آشنایی کلی با ساختار FreePBX و ایزوهای مبتنی بر آن

- فایل‌های بدون عبارت پسوندی (filename.conf) فایل‌های اصلی استریسک بوده و استریسک فقط آن فایل‌ها را می‌شناسد .
- فایل‌های با عبارت _additional فایل‌های متعلق به FreePBX بوده و محتوای آن‌ها پس از هر apply در وب ، بازنویسی می‌شوند .
- فایل‌های با عبارت _custom فایل‌های متعلق به توسعه دهندگان می‌باشد . بیشتر زمانی استفاده می‌شود که توسعه دهنده قصد ایجاد یک context یا یک extension جدید و یا بازنویسی یک extension از extension های FreePBX را دارد .
- فایل‌های با عبارت _override فایل‌های متعلق به توسعه دهندگان می‌باشد . بیشتر زمانی استفاده می‌شود که توسعه دهنده قصد بازنویسی یک context از context های FreePBX را دارد .
- همیشه به ترتیب فراخوانی فایل‌ها در یکدیگر باید دقت کرد . ترتیب فراخوانی‌ها در فایل‌های مختلف متفاوت می‌باشند .

بررسی فرایند ساخت یک داخلی در استریسک

- تعریف داخلی‌هایی از نوع DAHDI , IAX2 , SIP به ترتیب در فایل‌های chan_dahdi.conf , iax.conf , sip.conf انجام می‌شود .
- لازم به ذکر است که فقط تعریف داخلی‌ها ، در این فایل‌ها صورت می‌گیرد .
- نوع رفتار داخلی‌ها و عملکرد هر داخلی ، همگی در فایل extensions.conf مشخص می‌شود .
- در استریسک صرفا با تعریف یک داخلی ، آن داخلی توانایی هیچ‌کاری از جمله برقراری تماس را ندارد .

sip.conf (asterisk)
[401]
…
context = context1

iax.conf (asterisk)
[402]
…
context = context2

chan_dahdi.conf (asterisk)
[403]
…
context = context3

VOIPING

بررسی فرایند ساخت یک داخلی در ایزوهای مبتنی بر FreePBX

- تعریف داخلی‌هایی از نوع SIP , IAX2 , DAHDI به ترتیب در فایل‌های sip_additional.conf , iax_additional.conf , chan_dahdi_additional.conf انجام می‌شود .
- لازم به ذکر است که فقط تعریف داخلی‌ها ، در این فایل‌ها صورت می‌گیرد .
- نوع رفتار داخلی‌ها و عملکرد هر داخلی ، همگی در فایل extensions_additional.conf مشخص می‌شود .

sip_additional.conf (FreePBX_ISO)
[401]
…
context = from-internal

iax_additional.conf (FreePBX_ISO)
[402]
…
context = from-internal

chan_dahdi_additional.conf (FreePBX_ISO)
[403]
…
context = from-internal

بررسی ارتباط channel و context در استریسک

- دسترسی و مسیر دهی Channel ها در فایل extensions.conf بوسیله مقداردهی context در فایل مربوطه انجام می‌شود .

sip.conf :
[401]
…
context = context1

iax.conf :
[402]
…
context = context2

chan_dahdi.conf :
[403]
…
context = context3

extensions.conf :
[context1]
….
….
[context2]
....
….
[context3]
....
….

VOIPING

بررسی ارتباط channel و context در ایزوهای مبتنی بر FreePBX

sip_additional.conf :
[401]
…
context = **from-internal**

iax_additional.conf :
[402]
…
context = **from-internal**

chan_dahdi_additional.conf :
[403]
…
context = **from-internal**

extensions_additional.conf :
[context1]
….
….
….
[from-internal]
....
….
….
[context3]
....
….
….

VOIPING

تمرین Dialplan نویسی (مرور عملی مباحث گذشته)

- می‌خواهیم دو کاربر به نام‌های Ali و Reza از نوع SIP در استریسک ایجاد نماییم ، به گونه‌ای که هر دو بتوانند با یگدیگر تماس برقرار نمایند .
- در ابتدا با شماره گیری شماره‌های 901 و 902 و سپس با شماره گیری نام آن‌ها .

sip_custom.conf

```
[Ali]
secret=220voipingali
context=voiping
type=friend
permit=0.0.0.0/0.0.0.0
host=dynamic
[Reza]
secret=220voipingreza
context=voiping
type=friend
permit=0.0.0.0/0.0.0.0
host=dynamic
```

extensions_custom.conf

```
[voiping]
exten => 901,1,dial(sip/ali)
exten => 902,1,dial(sip/reza)

exten => ali,1,dial(sip/ali)
exten => reza,1,dial(sip/reza)
```

اپلیکیشن‌های Answer , Playback , Hangup

Answer([delay])

- تغییر وضعیت کانال به وضعیت پاسخ داده شده در صورت وضعیت ringing . (پاسخ به کانال)
- استریسک به مدت delay میلی‌ثانیه بعد از پاسخ به کانال منتظر خواهد ماند .

Playback(filename&[filename2[&...]],[options])

- پخش یک یا چند فایل صوتی در کانال .
- درصورت مشخص نکردن مسیر فایل ، استریسک به صورت پیش‌فرض به مسیر /var/lib/asterisk/sounds مراجعه خواهد کرد و به دنبال فایل صوتی خواهد گشت .
- اکستنشن فایل نباید به دستور داده شود .
- استریسک توانایی پخش فایل‌های صوتی به فرمت 8 khz 16 bit mono را خواهد داشت .
- options
  - skip : عدم پخش در صورتی که به کانال هنوز پاسخ داده نشده است .
  - noanswer : عدم پاسخ به کانال و پخش فایل های صوتی
- در صورت عدم استفاده از options کانال را به صورت خودکار پاسخ می‌دهد .

Hangup()

- تغییر وضعیت کانال به وضعیت خاتمه یافته . (قطع کانال)

دیباگ کدهای Dialplan و آشنایی مختصر با محیط CLI در استریسک

1. دستور لازم برای ورود به محیط CLI استریسک ، asterisk به همراه یک یا چند سوئیچ می‌باشد .
2. سوئیچ **r** برای اتصال به کنسول استریسک ، در زمان اجرا بودن سرویس استریسک ، می‌باشد . (Remote Console)
3. سوئیچ **R** همانند سوئیچ **r** عمل کرده ، با این تفاوت که پس از قطع ارتباط ، به صورت خودکار سعی بر اتصال مجدد می‌کند .
4. از ترکیب سوئیچ‌های فوق با سوئیچ **v** می‌توان با امکان مشاهده رخدادهای استریسک ، وارد کنسول استریسک شد .
7. استفاده بیشتر از سوئیچ **v** موجب مشاهده بیشتر جزئیات در کنسول استریسک خواهد شد .
8. تعداد **v** مشخص کننده سطح Verbosity در کنسول استریسک می‌باشد .
9. از سوئیچ **x** می‌توان برای اجرای دستورات کنسول استریسک در کنسول لینوکس استفاده کرد .
10. از کاراکتر **!** می‌توان برای اجرای دستورات کنسول لینوکس در کنسول استریسک استفاده کرد .

```
debug ; notice ; warning ; error ; dtmf ; fax ; security ; verbose
```

VOIPING

اپلیکیشن‌های Background , WaitExten , Goto

Background(filename&[filename2[&...]], [options])

- پخش یک یا چند فایل صوتی در کانال .
- دریافت dtmf از کاربر .
- درصورت مشخص نکردن مسیر فایل ، استریسک به صورت پیش‌فرض به مسیر /var/lib/asterisk/sounds مراجعه خواهد کرد و به دنبال فایل صوتی خواهد گشت .
- اکستنشن فایل نباید به دستور داده شود .
- استریسک توانایی پخش فایل‌های صوتی به فرمت 8 khz 16 bit mono را خواهد داشت .
- options
  - s : عدم پخش در صورتی که به کانال هنوز پاسخ داده نشده است .
  - n : عدم پاسخ به کانال و پخش فایل های صوتی
- در صورت عدم استفاده از options کانال را به صورت خودکار پاسخ می‌دهد .
- در صورت عدم وجود extension مطابق با dtmf وارد شده ، جریان تماس به extension ای به نام **i** منتقل خواهد شد .
- در صورت عدم ورود dtmf ، جریان تماس به extension ای به نام **t** منتقل خواهد شد .
- از اکستنشن **e** برای تجمیع دو حالت فوق استفاده می‌شود .

WaitExten([delay])

- انتظار به مدت delay ثانیه برای وارد کردن dtmf .

Goto([[context,]exten,]priority)

- انتقال جریان تماس به آدرس داده شده .
- جریان تماس باز نخواهد گشت .
- اگر مقصد یک extension در context فعلی باشد ، می‌توان context را مشخص نکرد .
- اگر مقصد یک priority در extension فعلی باشد ، می‌توان context و extension را مشخص نکرد .

VOIPING

اپلیکیشن‌های Verbose , NoOp

Verbose([level,]message)

- چاپ message در verbosity سطح level .
- در صورت مشخص نکردن level سطح پیش‌فرض 0 در نظر گرفته می‌شود .

NoOp()

- در واقع این دستور هیچ کاری را انجام نمی‌دهد .
- ولی از آنجاییکه دستورات Dialplan در Verbosity سطح 3 ، نمایش داده می‌شوند ، می‌توان از این دستور برای انجام کارهایی از قبیل دیباگ Dialplan و چاپ مقدار متغییرها در کنسول استریسک استفاده نمود .

بررسی انواع متغییرها در Dialplan

متغییرهای عمومی Channel

- این نوع از متغییرها وابسته به Channel بوده و طول عمر آن‌ها تا پایان عمر Channel می‌باشد .
- در هر Channel مقدار متفاوتی خواهند داشت .
- این نوع از متغییرها به دو نوع تقسیم بندی می‌شوند :
  - متغییرهای از پیش تعریف شده سیستمی که با حروف بزرگ دیده می‌شوند .
  - متغییرهای قابل تعریف در Dialplan که به صورت set(var_name=value) توسط برنامه نویس تعریف می‌شوند .
- نحوه دسترسی به مقدار متغییرها به صورت ${var_name} می‌باشد .

متغییرهای Global

- این نوع از متغییرها در تمامی Channel ها در دسترس می‌باشند .
- نحوه تعریف آن‌ها :
  - در سگمنت Globals در فایل extensions.conf و یا extensions_additional.conf می‌باشد .
  - با استفاده از دستور Set(GLOBAL(var_name)=value) در Dialplan می‌باشد .

متغییرهای Environment

- متغییرهای محلی سیستم عامل Linux می‌باشند .
- نحوه دسترسی به این نوع از متغییرها به صورت ${ENV(var_name)} می‌باشد .

VOIPING

بررسی اپلیکیشن Dial در Dialplan

Dial(Technology/Resource&[Technology2/Resource2[&...]],[timeout,[options,[URL]]])

- با مقصدهای مورد نظر تماس گرفته می‌شود و در صورتی که یک مقصد پاسخ دهد :
  - وضعیت کانال مبدا به پاسخ داده شده تبدیل خواهد شد .
  - تماس با مقاصد دیگر قطع خواهد شد .
  - کانال مبدا با کانال مقصد bridge خواهند شد .
  - URL برای مقصد فرستاده خواهد شد .
- به مدت timeout ثانیه منتظر پاسخ از مقاصد مورد نظر می‌ماند . در صورت مشخص نکردن مقدار پیش فرض 136 سال می‌باشد .
- این اپلیکیشن متغییر DIALSTATUS در کانال را مقدار دهی می‌کند .
  - ANSWER : یکی از مقاصد پاسخ داده است .
  - NOANSWER : از مقاصد مورد نظر پس از timout ثانیه پاسخی دریافت نشد .
  - BUSY : مقصد مورد نظر مشغول می‌باشد .
  - CONJESTION : شماره مقصد و یا کانال مقصد نامعتبر می‌باشد .
  - CHANUNAVAIL : مسیری برای تماس پیدا نشد . (طرف مقابل رجیستر نمی‌باشد .)
  - CANCEL : تماس گیرنده قبل از پاسخ یکی از مقاصد ، Hangup می‌کند.
  - ...
- options : به لینک‌های https://wiki.asterisk.org/wiki/display/AST/Asterisk+11+Application_Dial مراجعه شود .

  https://wiki.asterisk.org/wiki/display/AST/Asterisk+13+Application_Dial

VOIPING

برنامه‌ای هوشمند برای گرفتن موجودی و یا آخرین گردش‌ها از حساب‌های بانکی

- می‌خواهیم فقط با شماره گیری یک عدد 3 رقمی و بدون وارد کردن شماره حساب و یا رمز عبور و ... موجودی و یا چند گردش حساب بانکی خود را بشنویم .
- این عملیات بوسیله Dial کردن سرویس تلفن بانک ارائه شده توسط بانک مورد نظر انجام خواهد شد .
- این فرایند تقریبا در تمامی تلفن بانک‌های موجود در کشور قابل پیاده سازی می‌باشد .
- ما با استفاده از آپشن D در اپلیکیشن Dial اقدام به ارسال رشته‌های dtmf به صورت خودکار می‌نماییم تا اطلاعات مورد نیاز به سرور تلفن بانک ارسال شود .

VOIPING

اپلیکیشن Dial در CLI استریسک

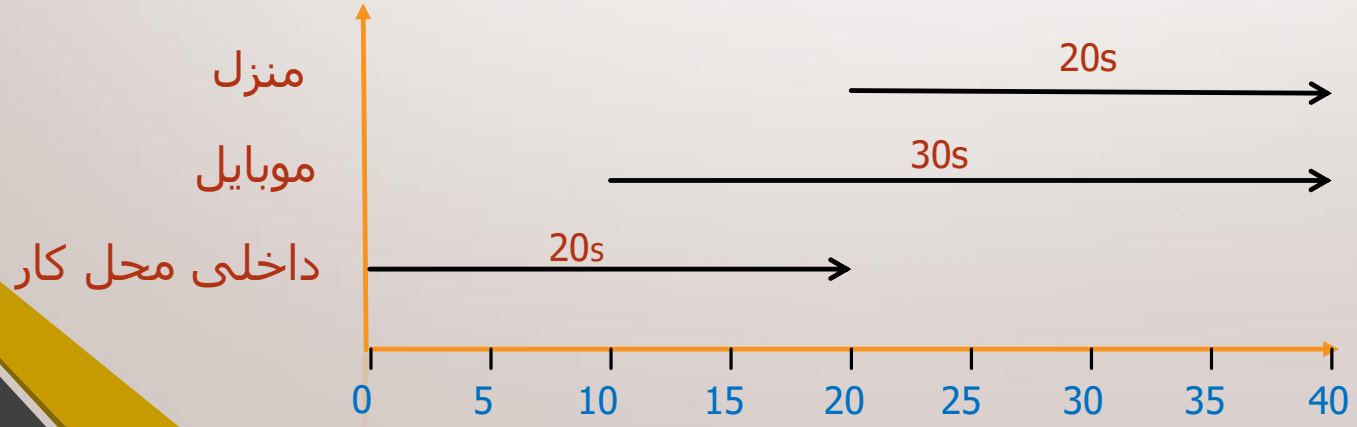
- برای استفاده از امکان Dial در CLI استریسک ، ماژول chan_oss.so باید در حالت فعال باشد .
  - module load chan_oss.so (راه‌کار موقت)
  - در فایل modules.conf ماژول فوق را در حالت پیش فرض Load قرار داده و سپس با دستور modules reload در CLI استریسک کلیه ماژول‌ها را دوباره بارگذاری نمایید . (راه‌کار دائم)
- فایل oss.conf باید در مسیر /etc/asterisk موجود باشد و تنظیمات لازم نیز انجام شده باشد .
- در صورت فعال بودن این ماژول ، شما دارای یک داخلی بدون شماره که نیاز به رجیستر شدن هم ندارد ، در CLI استریسک خواهید بود .
- این داخلی توانایی انجام کارهایی از قبیل ایجاد تماس ، پاسخ به تماس ، ارسال متن و .... خواهد داشت .
- در صورت فعال بودن کارت صدا و مهیا بودن ابزار لازم ، شما می‌توانید مانند یک تلفن با داخلی ها و شماره های مختلف صحبت نمایید .
- به این داخلی ، داخلی کنسول گفته می‌شود .
- از این امکان برای انجام تست‌ها و اجرای برنامه‌های Dialplan هم می‌توان استفاده نمود .
- دستور لازم برای عمل Dial در کنسول استریسک : console dial extension@context

آشنایی با Channel های Local در استریسک

- تا قبل از استریسک 12 , Local Channel ها در ماژولی به نام chan_local.so بارگذاری می‌شدند .
- از استریسک 12 ، chan_local به system core استریسک منتقل شده و لذا برای استفاده از آن نیازی به Load ماژول خاصی در استریسک نمی‌باشد .
- پس از اتصال مبدا و مقصد به یکدیگر توسط Local Channel ها اطلاعات این Local Channel ها همگی از بین می‌روند .
- برای جلوگیری از ، از بین رفتن اطلاعات Local Channel ها باید از modifier های مناسب (\n) استفاده شود .
- موارد استفاده :
  - بهینه سازی پروژه‌هایی که در آن‌ها از Call File استفاده می‌شود .
  - بهینه سازی پروژه‌های AMI ای که در آن‌ها از اکشن originate استفاده می‌شود .
  - برای تماس با چندین مقصد ولی با اطلاعات و زمان‌های انتظار متفاوت .
  - بهینه سازی اپلیکیشن Queue
  - در کل در کدنویسی‌های پیشرفته استفاده می‌شود .

سناریو :

- میخواهیم با سه شماره به صورت همزمان تماس بگیریم .
- پس از پاسخ هر کدام از شماره‌ها عمل Dial مابقی شماره‌ها متوقف شود .
- زمان شروع تماس با هر شماره و زمان انتظار برای پاسخ هر شماره متفاوت خواهد بود و مطابق نمودار زیر می‌باشد .



آشنایی با Function ها در Dialplan

- استریسک با معرفی Function ها ابزارهای برنامه نویسی را در Dialplan در کنار Application ها توسعه داد .
- برای مشاهده لیست Function ها و همچنین مشاهده توضیح مختصر برای هر یک از دستور زیر در CLI استریسک می‌توان استفاده نمود :
  - core show functions
- برای آشنایی با عملکرد هر Function به صورت مجزا از دستور زیر در CLI استریسک می‌توان استفاده نمود :
  - core show function FUNC-NAME
- برای هر Function در استریسک یک ماژول هم‌نام آن با فرمت func_xxxxx.so وجود دارد .
- Function ها همگی با حروف بزرگ قابل دسترسی می‌باشند .
- بر خلاف Application ها از بسیاری از Function ها می‌توان خروجی دریافت کرد .
- فرمت استفاده از Function ها FUNCTION(argument1,argument2, ...) می‌باشد .
- زمانی که از یک Function قرار است مقداری را به عنوان خروجی دریافت نماییم ، از فرمت ${FUNCTION(argument)} استفاده می‌شود .

آشنایی با Function ها در Dialplan

- SET() : SET assigns a value to a channel variable.
- CALLERID() : Gets or sets Caller*ID data on the channel.
- CDR() : Gets or sets a CDR variable.
- CURL() : Retrieve content from a remote web or ftp server .
- DB() : Read from or write to the Asterisk database.
- DB_DELETE() : Return a value from the database and delete it.
- EXTENSION_STATE() : Get an extension's state.
  - UNKNOWN | NOT_INUSE | INUSE | BUSY | INVALID | UNAVAILABLE | RINGING | RINGINUSE | HOLDINUSE | ONHOLD
- FILE() : Read or write text file.
- FILTER() : Filter the string to include only the allowed characters .
- GLOBAL() :Gets or sets the global variable specified.
  GROUP() : Gets or sets the channel group.
- GROUP_COUNT() : Counts the number of channels in the specified group.
- IF() : Check for an expression.
- LEN() : Return the length of the string given.
- RAND() : Choose a random number in a range.
- REGEX() : Check string against a regular expression.
- VOLUME() : Set the TX or RX volume of a channel.

VOIPING

فارسی سازی فایل‌ها و تغییر زبان پیش‌فرض استریسک

- مبحث فارسی سازی در دو قسمت انجام خواهد شد :
  - فارسی سازی فایل‌های صوتی
  - فارسی سازی نحوه خواندن اعداد
- استریسک برای خواندن یک فایل صوتی در صورت مشخص نشدن مسیر آن فایل به مسیر /var/lib/asterisk/sounds/{default-language} مراجعه کرده و در صورت پیدا نکردن فایل مربوطه به دایرکتوری مادر برگشته و آنجا به دنبال فایل صوتی می‌گردد . باز هم در صورت پیدا نکردن فایل مربوطه به دایرکتوری en مراجعه خواهد کرد .
- زبان پیش‌فرض در استریسک وابسته به تکنولوژی Channel می‌باشد .
- برای تغییر زبان پیش‌فرض در استریسک به فارسی باید :
  - پوشه حاوی فایل‌های صوتی به زبان فارسی (pr) در مسیر /var/lib/asterisk/sounds اضافه گردد .
  - عبارت language=pr برای کانال‌هایی از نوع sip,iax در سگمنت general و برای کانال‌های dahdi در سگمنت channels ، در فایل‌های مربوطه اضافه گردد .
    - sip_general_custom.conf ← sip.conf
    - iax_general_custom.conf ← iax.conf
    - chan_dahdi.conf
- استریسک برای خواندن اعداد می‌تواند از یکسری دستورالعمل که در فایل /etc/asterisk/say.conf می‌باشد پیروی کند .
- دستورات برای خواندن اعداد فارسی در فایل say.conf باید اضافه گردد .
- پیشنهاد می‌شود فایل say.conf موجود در بسته آموزشی را با فایل موجود در سرور استریسک عوض نمایید .
- پس از انجام کلیه تغییرات ، سرویس استریسک را reload نمایید .

VOIPING

اپلیکیشن‌های SayNumber , SayDigits , SayAlpha , SayPhonetic

SayDigits(number)

- خواندن اعداد number به صورت رقم به رقم .

SayNumber(number)

- خواندن عدد number به صورت کامل .

SayAlpha(string)

- خواندن string به صورت حرف به حرف .
- این دستور توانایی خواندن اعداد و حروف و کاراکترها را دارد .

SayPhonetic(string)

- خواندن کلماتی که با حروف موجود در string شروع می‌شوند .
- این دستور توانایی خواندن اعداد و حروف و کاراکترها را دارد .

اپلیکیشن DISA

DISA(password,[context,[cid,mailbox@[context],[options]]]])

- Direct Inward System Access
- رعایت مسائل امنیتی در استفاده از این اپلیکیشن بسیار با اهمیت می‌باشد .
- دسترسی مستقیم به یک context خاص از خارج مرکز تلفن .
- این context بوسیله آرگومان دوم مشخص می‌شود .
- در صورت مشخص نکردن context ، استریسک به صورت پیش فرض به دنبال یک context به نام disa می‌گردد .
- برای استفاده از این اپلیکیشن می‌توان از یک کلمه عبور و یا یک فایل برای گذرواژه‌ها استفاده کرد .
- اگر آرگومان اول با / شروع شود ، استریسک عمل اعتبار سنجی را با یک فایل انجام خواهد داد .
- در صورت عدم نیاز به کلمه عبور ، آرگومان اول را با مقدار no-password باید مقدار دهی کرد .
- می‌توان با مقدار دهی cid برای این کانال یک کالر آیدی جدید مقدار دهی کرد .
- در صورت مشخص کردن شماره یک mailbox ودر صورت پر بودن آن mailbox کاربر ابتدا چند صدای بوق کوتاه و سپس صدای بوق disa را می‌شنود .
- options
  - n : استریسک در ابتدای کار به کانال پاسخ نمی‌دهد .
  - p : شماره وارد شده توسط کاربر بعد از ورود کاراکتر # مورد خواندن قرار می‌گیرد .

اپلیکیشن Authenticate

Authenticate(password,[options,[maxdigits]])

- برای کاربر عملیات اعتبار سنجی صورت می‌گیرد .
- در صورت موفق بودن اعتبار سنجی Dialplan به خط بعدی منتقل خواهد شد و در صورت شکست Dialplan ادامه پیدا نخواهد کرد و کانال Hangup خواهد شد .
- تا قبل از hanggup شدن کانال ، کاربر می‌تواند تا سه مرتبه سعی به اعتبار سنجی نماید .
- کاربر برای اتمام عملیات اعتبار سنجی باید کاراکتر # را وارد نماید ، مگر اینه maxdigits مشخص شده باشد .
- اگر آرگومان اول با / شروع شود ، استریسک عمل اعتبار سنجی را با یک فایل انجام خواهد داد .
- برای اعتبار سنجی در این اپلیکیشن از دیتابیس خود استریسک هم می‌توان استفاده کرد .

اپلیکیشن‌های Read , System

Read(variable,filename&[filename2[&...]],[maxdigits,[options,[attempts,[timeout]]]]])

- از این اپلیکیشن برای گرفتن اطلاعات از کاربر استفاده می‌شود .
- این اطلاعات در متغییری که به عنوان آرگومان اول مشخص شده ، ذخیره خواهد شد .
- می‌توان قبل از گرفتن اطلاعات برای کاربر یک یا چند فایل صوتی پخش کرد .
- کاربر باید در آخر dtmf های خود کاراکتر # را وارد نماید مگر اینکه maxdigits مشخص شده باشد .
- می‌توان حداکثر 255 کاراکتر از کاربر به عنوان اطلاعات دریافت نمود .
- این اپلیکیشن متغییر READSTATUS را در کانال مقدار دهی می‌نماید :
  - INTERRUPTED , OK , ERROR , HANGUP , SKIPPED , TIMEOUT

System()

- از این اپلیکیشن برای اجرای یک دستور در محیط Linux استفاده می‌شود .
- دستورات در محیط Linux با دسترسی و قدرت کاربر asterisk اجرا خواهند شد .
- این اپلیکیشن یک متغییر به نام SYSTEMSTATUS را در کانال مقدار دهی می‌نماید :
  - FAILURE , SUCCESS
- از این اپلیکیشن خروجی نمی‌توان دریافت نمود .
- برای گرفتن خروجی بهتر است از function ای به نام SHELL استفاده کرد .

VOIPING

بررسی Expression در Dialplan

- هر عبارتی که بین $[...] قرارگیرد ، از لحاظ استریسک یک عبارت ریاضی و منطقی قلم داد شده و استریسک اقدام به محاسبه آن کرده .
- پس از پایان محاسبه استریسک مقدار نهایی را جایگزین $[...] خواهد کرد .
- برای محاسبات بلند و پیچیده استریسک محیط Dialplan را مناسب ندانسته و اینترفیس AGI خود را پیشنهاد می‌دهد .

```
exten => 111,1,noop()
same => n,set(x=$[1+1])
same => n,set(y=$[2*${x}])
```

کار با رشته های متنی در Dialplan

- توابع مورد نیاز برای کار با رشته ها :

LEN(var-name)

STRREPLACE(varname,find-string[,replace-string[,max-replacements]])

- انتخاب کاراکترها در یک رشته :

${varname:offset:length}

```
exten => 111,1,set(var=02174327)
        same => n,verbose(${var:3:2})   ;74
        same => n,verbose(${var:3})     ;74327
        same => n,verbose(${var:-2:1})  ;2
        same => n,verbose(${var:-5:-1}) ;7432
        same => n,verbose(${var:-5:-2}) ;743
        same => n,verbose(${var:-5:-3}) ;74
        same => n,verbose(${var:-5})    ;74327
```

اپلیکیشن‌های GotoIF , GotoIFTime , ExecIF

GotoIF()

- برای ارجاعات شرطی در Dialplan استفاده می‌شود .

GotoIf(condition?[[context1,]extension1,]priority1]:[[[context2,]extension2,]priority2])

GotoIFTime()

- برای ارجاعات شرطی زمانی در Dialplan استفاده می‌شود .

GotoIfTime(times,weekdays,mdays,months,[timezone]?[labeliftrue:[labeliffalse]])

<time range>= <hour>:<minute>-<hour>:<minute> | *
<days of week> = <dayname> | <dayname>-<dayname> | *
<dayname> = sun | mon | tue | wed | thu | fri | sat
<days of month> = <daynum> | <daynum>-<daynum> | *
<daynum> = 1 to 31
<hour> = 0 to 23
<minute> = 0 to 59
<months> = <monthname> | <monthname>-<monthname> | *
<monthname> = jan | feb | mar | apr | may | jun | jul | aug | sep | oct | nov | dec

ExecIF()

- برای اجرای شرطی یک اپلیکیشن استفاده می‌شود .

ExecIf(condition?appiftrue:[appiffalse])

VOIPING

بررسی Dial Patterns در استریسک

- X matches any digit from 0-9
- Z matches any digit from 1-9
- N matches any digit from 2-9
- [ ] matches any digit or letter in the brackets
- . matches one or more characters
- ! matches zero or more characters immediately

- برای استفاده از Dial Pattern ها در Dialplan همواره باید از کاراکتر _ درابتدای pattern استفاده کرد .
- از / برای مشخص کردن یک کالر آیدی خاص در Dial Pattern استفاده می‌شود .

- exten => _X,1,noop()
- exten => _X.,1,noop()
- exten => _Z.,1,noop()
- exten => _[123].,1,noop()
- exten => _[1-5].,1,noop()
- exten => _[a-f].,1,noop()
- exten => _4[89]./401,1,noop()
- exten => _4[89]./_4.,1,noop()

VOIPING

بررسی Dial Patterns در استریسک

- اولویت برای اجرای dialplan در Pattern ها بر اساس فاکتور های زیر صورت می‌گیرد :
  1. عبارات بدون الگو در اولویت اول می‌باشند .
  2. هر چه تعداد حالات یک الگو برای یک digit کمتر باشد ، اولویت اجرا بالاتر است .
  3. در صورت یکی بودن تعداد حالات ، اولویت با سورت معمولی می‌باشد . اول اعداد از 0 تا 9 و بعد حروف از a تا z .
- با شماره گیری عدد 6421 اولویت ها به ترتیب زیر خواهد بود .

```
exten => _6XX1,1,noop(1)
exten => _64XX,1, noop(2)
exten => _642X,1, noop(3)
exten => _6.,1, noop(4)
exten => _64NX,1, noop(5)
exten => _6[45]NX,1, noop(6)
exten => _6[34]NX,1, noop(7)
```

```
exten => _642X,1, noop(3)
exten => _64NX,1, noop(5)
exten => _64XX,1, noop(2)
exten => _6[34]NX,1, noop(7)
exten => _6[45]NX,1, noop(6)
exten => _6XX1,1, noop(1)
exten => _6.,1, noop(4)
```

برای مشخص شدن اولویت اجرایی در Dial Pattern ها می‌توان از دستور زیر استفاده کرد :

```
dialplan show exten@context
```

بررسی اپلیکیشن ChanSpy

ChanSpy([chanprefix,[options]])

- شنود صدای خروجی و دریافتی در channel های استریسک .
- نجوا در channel مورد نظر
- در هنگام شنود dtmf های زیر فعال می‌باشند :
  - # : برای کم و زیاد کردن صدای شنود ، از -4 تا 4
  - * : قطع شنود channel فعلی و شنود channel بعدی (پیمایش channel ها برای شنود)
  - X : هر عددی که گرفته شود و با # تمام شود آن عدد به انتهای chanprefix اضافه خواهد شد : (این قابلیت با آپشن d بازنویسی می‌شود)

ChanSpy(Agent) → 1234# → Chanspy(Agent/1234)

- Options :
  - **بدون آپشن :** فقط مد شنود برای صدای ارسالی و دریافتی
  - **b** : شنود فقط در channel های bridge شده
  - **B** : می‌توان در هر دو کانال هم شنود و هم نجوا کرد (شبیه به کنفرانس 3 نفره می‌شود)
  - **E** : پایان عملیات شنود در صورت hangup کردن شنود شونده
  - **d** :
    - 4 : مد شنود
    - 5 : مد نجوا
    - 6 : مد کامل (هم نجوا و هم شنود ، شبیه به کنفرانس 3 نفره)
  - **o** : فقط شنود صدای ارسالی در Channel
  - **q** : در ابتدای عملیات شنود نه صدای beep و نه نام channel انتخابی پخش نمی‌شود
  - **r** : ضبط صداهای channel انتخاب شده . در مسیر , /var/spool/monitor به صورت hiden ، مگر اینکه مسیر و نام فایل مشخص شود .
  - **w** : شنود کننده می‌تواند با شنود شونده نجوا کند . شنود کننده مکالمه شنود شونده و مخاطب آن را می‌شنود ولی صدای شنود کننده را فقط شنود شونده می‌شنود (کمک به پرسنل در پاسخ‌گویی به مشتریان)
  - **W** : نجوا کننده هیچ شنودی انجام نمی‌دهد و فقط میتواند در channel مورد نظر نجوا نماید . این نجوا را مخاطب channel مورد نظر نمی‌شنود . (یاداوری منشی به مدیرِ درحال مکالمه)

VOIPING

سناریوی ششم : بهبود عمل شنود در ISSABEL , FreePBX , Elastix

دسترسی :

دسترسی به اپلیکیشن شنود در سازمان باید کاملا مدیریت شده باشد .

- ✗ مدیریت به‌وسیله گذاشتن پسورد
- ✓ مدیریت به‌وسیله کالر آیدی

شنود و نجوای مستقیم :

باید در هر لحظه بتوان یک داخلی خاص را به صورت مستقیم شنود/نجوا کرد و نیازی به پیمایش نباشد .

امکانات شنود و نجوا :

1. شنود عادی
2. شنود و نجوا به عنوان supervisor **(w)**
3. نجوای منشی در کانال فعال مدیر **(W)**

بررسی Extension های از پیش تعریف شده

Operator extensiono : o
Assistant extension : a

Response timeout extension : t
Absolute timeout extension : T

Invalid entry extension : i

در صورت عدم یافتن digit معتبر در اپلیکیشن‌های background و waitexten کانال به این extension مسیردهی می‌شود .

Hangup extension : h

در صورت وجود این extension در context فعال ، زمانی که کانال hangup می‌شود ، کانال به این extension مسیر دهی می‌شود .
از این extension برای عملیاتی مثل فرستادن اطلاعات به پایگاه داده و ارسال ایمیل و ... استفاده می‌شود . در این extension هیچ صدایی در کانال پخش نمی‌شود .

Exception Catchall extension : e

جایگزین اکستنشن‌های **i,t,T** می‌باشد . در صورت عدم وجود **i,t,T** کانال به این extension مسیر دهی می‌شود .

VOIPING

بررسی Extension های از پیش تعریف شده

Start extension : s

1. زمان ورود کانال از خطوط شهری به dialplan
2. ماکروها

: failed

در auto-dialer ها مانند Call File ها استفاده می‌شود . در زمانی که از جریان context,extension,priority استفاده می‌شود چنانچه تماس پاسخ داده نشود و یا کانال مورد نظر اشغال باشد ، کانال به این extension مسیر دهی خواهد شد .

: fax

در صورت شناسایی سیگنال Fax کانال به این extension مسیر دهی خواهد شد .

: Talk

در اپلیکیشن BackGroundDetect استفاده می‌شود .

آشنایی با Macro و Subroutine ها

Macro :

- قطعه برنامه ایست مانند context ها برای جلوگیری از تکرار کد نویسی در dialplan .
- کارکرد function ها در زبان‌های برنامه نویسی را در Dialplan استریسک دارد .
- همانند context ها در dialplan تعریف می‌شود ، با این تفاوت که اول نام گذاری از عبارت -macro باید استفاده شود .
- مدیریت channel پس از اجرای macro به صورت اتوماتیک به محل صدا زدن macro باز می‌گردد .
- extension ای که در macro استفاده می‌شود معمولا **s** می‌باشد .
- به macro می‌توان آرگومان ارسال کرد .
- نحوه صدا زدن macro به صورت Macro(macroname,arg1,arg2…) می‌باشد .
- برای دسترسی به آرگومان‌ها در macro از عبارات ${arg1},${arg2},… استفاده می‌شود .
- در macro سه متغییر ${MACRO_PRIORITY} , ${MACRO_EXTEN} , ${MACRO_CONTEXT} که نشان دهنده محل صدا زدن آن macro می‌باشد از قبل مقدار دهی شده است .
- چنانچه در macro از Dial استفاده شود موقع hangup کردن callee ، به صورت اتوماتیک caller به اکستنشن h در صورت وجود منتقل می‌شود .
- macro ها در صدا زدن یکدیگر (macro در macro) ضعف دارند ، (ماکسیمم در 5 سطح) از همین رو استریسک استفاده از macro ها را پیشنهاد نمی‌کند و راه‌کاری دیگر را معرفی کرده است .

VOIPING

آشنایی با Macro و Subroutine ها

Subroutine :

- رفتار مشابه macro دارد .
- برای نام گذاری قطعه کد subroutine نیاز به قانون خاصی نیست .
- مدیریت channel پس از اجرای subroutine به صورت اتوماتیک به محل صدا زدن subroutine باز نمی‌گردد .
- برای بازگشت جریان برنامه باید از اپلیکیشن return() استفاده کرد .
- به subroutine ها می‌توان آرگومان ارسال کرد .
- در subroutine ها بر خلاف macro ها متغییرهای از پیش تعریف شده‌ای وجود ندارد ، لذا به هر اطلاعاتی که نیاز داریم باید در قالب یک آرگومان به subroutine ارسال نماییم .
- نحوه صدا زدن subroutine ها به صورت Gosub(subname,{extension},{priority}(arg1,arg2…)) می‌باشد .
- برای دسترسی به آرگومان‌ها در subroutine از عبارات ${arg1},${arg2},… استفاده می‌شود .
- در بازگشت از subroutine می‌توان به اپلیکیشن return() مقداری را به عنوان خروجی subroutine منتقل کرد و در dialplan تحت نام ${GOSUB_RETVAL} از آن استفاده کرد .

آشنایی با مسیر های Dial در FreePBX و ایزوهای مبتنی بر آن

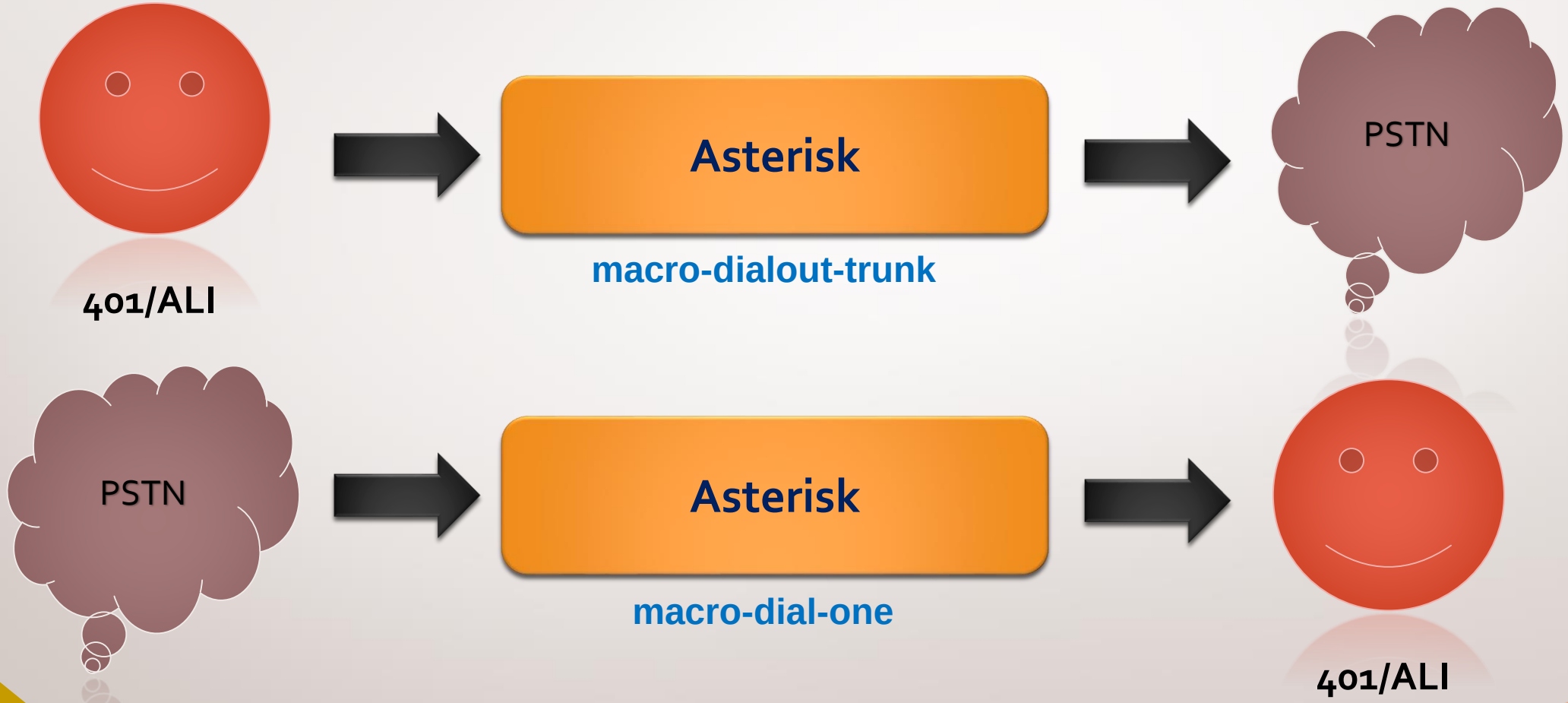
: macro-dialout-trunk

- برای تماس به سمت بیرون از IPPBX استفاده می‌شود .
- زمانی که داخلی ها اقدام به تماس با خارج از سازمان می‌کنند .
- تماس قرار است از سرور استریسک به سمت ترانک‌های خروجی منتقل شود .

: macro-dial-one

- برای تماس‌های وارده به IPPBX و اتصال به یک extension استفاده می‌شود .
- زمانی که تماسی وارد سرور استریسک شده و قرار است به یک extension متصل شود .

آشنایی با مسیر های Dial در FreePBX و ایزوهای مبتنی بر آن



با تمرین زیاد به آسانی یاد بگیریم

VoiPing.ir

مولف : محمد یزدی‌زاده