

Telefonía VoIP: Asterisk

Formación en soluciones de código abierto para pymes



Modulo 1

Introducción a Asterisk

Modulo1: Conceptos

- En este módulo veremos lo siguiente:
 - Qué es Asterisk?
 - Quién lo creó y porque?
 - Es Asterisk un sistema para mí?
 - Hay que pagar licencias?

Que es Asterisk

- Asterisk es una solución de centralita telefónica (PBX) desarrollada en código abierto:
 - Usuarios domésticos
 - Pequeñas y medianas empresas
 - Grandes empresas
 - Proveedores de VoIP
 - Compañías telefónicas

Qué o quién es Digium

- Son los principales desarrolladores y mantenedores de Asterisk
- Son proveedores de Soluciones hardware para telefonía IP (tarjetas)

Historia de Digium y Asterisk

- Mark Spencer creó Linux Support Services (LSS) en 1999 y necesitaba un sistema de telefonía
- Estudio el coste de diferentes soluciones propietarias y debido al coste de ellas, decidió desarrollar una.
- Le llamo Asterisk (el asterisco es el carácter comodín en Linux, y vale para todo)
- En 2002, se convirtió en Digium y dedicada exclusivamente a Asterisk

Mark Spencer



Asterisk y Digium hoy en día

- Cerca de dos millones de servidores puestos en producción
- Más de mil descargas diarias
- Miles de usuarios activos en las listas de correo
- Más de 200 empleados

Asterisk como PBX

- Asterisk es usado como PBX (Private Branch Exchange). Una PBX comparte recursos con usuarios (líneas de teléfono – usuarios)
- Permite a los usuarios realizar llamadas utilizando líneas de teléfono
- Como PBX dispone de buzón de voz, llamadas en grupo, contestador automático

Asterisk: más que una PBX

- El programa
- La plataforma
- El producto
- El proyecto
- La comunidad

Asterisk: El programa

- Es un sistema de telefonía
- Escalable, desde una pequeña empresa hasta miles de extensiones (más de 10.000)
- Se ejecuta en Linux, Mac OSX, Solaris, *nix
- Solución híbrida: VoIP y telefonía convencional

Asterisk: Que es lo que no es

- No es un producto de consumo
- No está preparado para funcionar después de ser descargado
- No es un proveedor de VoIP
- No es un proxy SIP

Asterisk: Licencia

- Cualquier producto de software viene con una licencia que dicta que es lo que se puede hacer con el, y lo que no nos está permitido.
- La licencia no es lo mismo que el Copyright

Asterisk: 3 formas de licencia

- Open Source, bajo el formato GPL
- Asterisk Business Edition
 - Estable y testeado a conciencia
 - Licencia comercial de Digium
 - Incluye documentación y un año de soporte
- OEM
 - Designados por terceros que generalmente incluyen hardware y desarrollos personalizados.

Modulo 2

Arquitectura de Asterisk

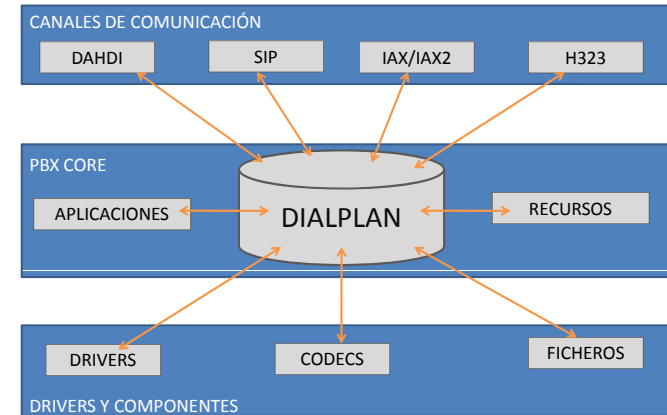
Modulo2: Conceptos

- En este módulo veremos lo siguiente:
 - Entender los componentes y la estructura de Asterisk
 - Reconocer los interfaces y la programación en Asterisk
 - Reconocer la estructura de directorios
 - Aprender acerca de los archivos de configuración más importantes de asterisk

Asterisk & Linux

- Asterisk corre en cualquier plataforma *nix
- Asterisk fue creado para su uso en Linux
 - El soporte de las tarjetas de Digium está limitado a Linux
 - Soporta a partir del kernel 2.6
- En la mayoría de los casos no requiere un hardware específico (servidor/ordenador)

Asterisk : el esquema



PBX Core

- El Core es el corazón de Asterisk y se encarga de:
 - Cargar los módulos dinámicamente
 - Leer los ficheros de configuración
 - Ejecutar aplicaciones
 - Procesar las peticiones del dialplan
 - Crear las instancias de los canales para las llamadas
 - Convertir entre formatos, códecs y protocolos

Asterisk: Arquitectura modular

- Asterisk tiene una arquitectura modular
- Cada módulo provee de una pieza de funcionalidad
- Los módulos pueden contener recursos, aplicaciones, comandos, drivers, códecs, etc.
- Los administradores de Asterisk deciden que módulos se cargan al inicio.

Asterisk: canales

- Los canales son los encargados de la entrada y salida de información (voz, video, texto, etc)
- Cada canal representa una tecnología o un protocolo (SIP, IAX, DAHDI, H.323)
- Cada canal es representado por un modulo (chan_sip.so, chan_iax2.so, etc.)
- Cada driver del canal es el encargado de traducir y convertir información para que la entienda asterisk.

Asterisk: aplicaciones

- Implementan funcionalidades en el sistema
 - Buzón de voz (para dejar y recibir mensajes)
 - Reproducción de ficheros de sonido
 - Audio conferencias
- Se cargan dinámicamente cuando el núcleo lo necesita, y viceversa
- Múltiples aplicaciones pueden ser invocadas durante una llamada, pero consecutivamente, no simultáneamente
- Se pueden pasar parámetros a las aplicaciones para modificar su comportamiento

Asterisk: Recursos

- Similares a las aplicaciones, pero se cargan estáticamente al inicio
- Proveen funcionalidades comunes que pueden ser usadas en cualquier momento
 - Música en espera
 - Grabación de llamadas
 - Parking de llamadas
 - Carga / descarga de ficheros de configuración

Asterisk: Códecs

- Codifican / decodifican
- Representan algoritmos para codificar y decodificar multimedia
 - Video, audio, etc.
 - La mayoría de ellos comprimen y descomprimen la información
- Los utiliza Asterisk para:
 - Interpretar el audio/video entrante
 - Codificar el audio/video saliente
- Codec_gsm.so, codec_alaw.so

Asterisk: CLI

- Es el interface para interactuar con Asterisk (Command Line Interface)
- Permite a los administradores:
 - Consultar el estado del sistema
 - Iniciar, parar o reiniciar el sistema
 - Actualizar valores en tiempo real
 - Mostrar información sobre comandos, aplicaciones, etc.
- Son interpretados por Asterisk, pero cada módulo tiene sus propios comandos

Asterisk: configuración

- Por defecto Asterisk carga su configuración de ficheros de texto
- Se puede configurar el sistema para una arquitectura ARA “Asterisk Realtime Architecture”
 - Podría leer la información de configuración de bases de datos, LDAP, etc.
- La mayoría de los usuarios utiliza los ficheros de configuración basados en texto

Asterisk: Ficheros de configuración

- Asterisk se configura editando ficheros de texto
 - Por defecto están almacenados en /etc/asterisk
 - Todos tiene la extensión .conf
 - Son ficheros de texto plano que pueden ser editados con cualquier editor de texto
- La instalación de Asterisk instala todos los ficheros de configuración necesarios, y en ellos vienen ejemplos documentados

Asterisk: Ficheros de configuración

- Asterisk utiliza bastantes ficheros de configuración, pero todos tienen la misma sintaxis
- Los ficheros de configuración se dividen en secciones, y estas tienen un nombre que tiene que ir entre corchetes []
- Los comentarios en los ficheros de configuración son los punto y coma (;)
- Los bloques de comentario comienzan por ;-- y finalizan por --;


```
[nombreseccion]
Parametro=1
; esto es un comentario
[otraseccion]
;-- esto es un bloque de comentario
que acaba en la linea inferior
--;
```

Asterisk: Ficheros de configuración

- Los cinco ficheros principales son:
 - asterisk.conf
 - sip.conf
 - iax.conf
 - chan_dahdi.conf
 - extensions.conf

asterisk.conf

- Es el fichero principal de configuración de Asterisk
- Asterisk espera que este situado en el directorio `/etc/asterisk`. Se puede especificar otra ubicación arrancando asterisk con el parámetro `-c`
- Raramente necesita ser editado o modificado
 - Un parámetro interesante para editar es `"nocolor=yes"`

sip.conf

- Configura el canal SIP para asterisk
- Se pueden definir valores generales bajo el contexto `[general]`
- Contiene la configuración para los dispositivos SIP
- También contiene la configuración para los proveedores SIP

iax.conf

- Configura el canal IAX para asterisk
- Se pueden definir valores generales bajo el contexto `[general]`
- Contiene la configuración para los dispositivos IAX
- También contiene la configuración para los proveedores IAX

chan_dahdi.conf

- Guarda la configuración para las tarjetas de comunicaciones (Analógicas o digitales)
- Almacena la configuración de los parámetros de las líneas de comunicaciones para las conexiones externas (trunks)
- Aquí configuramos el "caller ID" de las líneas de entrada, los grupos de llamada, etc.
- Trabaja en conjunción con la configuración de la tarjeta de comunicaciones almacenada en /etc/dahdi/system.conf

extensions.conf

- Almacena la configuración del dialplan (plan de llamadas)
- Le indica a Asterisk como proceder con las llamadas
- Es el fichero más crítico de asterisk, y el más importante, y en él reside la lógica de nuestro plan de llamadas.

Módulo 3

Instalar asterisk y sus componentes

Modulo 3: Conceptos

- En este módulo veremos lo siguiente:
 - Aprender a compilar e instalar Asterisk
 - Diferentes formas de parar y arrancar Asterisk
 - Utilizar CLI (Command Line Interface)

Asterisk: Componentes

- Libpri
 - Contiene soporte RDSI para líneas T1/E1/J1
- DAHDI
 - Contiene los drivers para las tarjetas analógicas y digitales
- Asterisk
 - Código fuente del programa
- Opcionalmente Asterisk-addons
 - Módulos opcionales o experimentales para asterisk

Asterisk: ficheros de sonido

- Asterisk tiene la voz de Alison en inglés. Se pueden descargar las voces en español, pero son habladas por Alison, con acento inglés
- Existen sitios en internet donde se pueden descargar voces en castellano, grabadas por profesionales y gratis.

Asterisk: componentes necesarios

- Requisitos mínimos de software
 - OpenSSL para la conexión remota
 - Ncurses para el CLI
 - Zlib (librería de compresión)
 - Compilador gcc, librerías y headers del kernel
- Otros componentes opcionales
 - Curl para interactuar con sitios web
 - Sendmail o postfix para en envío de los mensajes del buzón de voz

Asterisk: Ficheros y directorios

- /usr/sbin/asterisk
 - Ficheros binarios de asterisk
- /etc/asterisk
 - Ficheros de configuración de asterisk
- /usr/lib/asterisk/modules
 - Módulos para las aplicaciones, códecs, formatos y canales
- /var/lib/asterisk/sounds
 - Locuciones de Mensajes de asterisk
- /var/lib/asterisk/moh
 - Música en espera

Asterisk: Arrancar y parar

- El orden de arranque es:
 - DAHDI
 - Asterisk
- El orden de parada es:
 - Asterisk
 - DAHDI

Asterisk: Arrancar y parar

- Arrancar asterisk
 - /usr/sbin/asterisk
 - asterisk start
 - La mejor forma es mediante el ultimo comando, o con cualquier otro script de arranque del sistema operativo.

Asterisk: conectarse a CLI

- Cuando Asterisk esta en ejecución podemos conectarnos a él mediante CLI
 - Asterisk -r
 - Asterisk -rvvv (las v significan verbose, cuantas más v, mas información en pantalla)
 - Asterisk -rddd (las d significan debug, cuantas más d, mas información en pantalla)

Asterisk: comandos CLI

- help
- quit
- stop now
- core set verbose 5
 - Es lo mismo que asterisk .rvvvv
- ! Abre una ventana de comandos Unix. Pulsar Control +D o exit para regresar a CLI
- ! Comando Unix, ejecuta un comando Unix y regresa a la consola

Asterisk: comandos CLI

- Para obtener ayuda
 - help
 - help stop
 - core show applications
 - core show application dial
 - core show functions
 - core show function LEN
- !ls

Asterisk lab

- Libro de ejercicios

ASTERISK LAB

MODULO 3

INSTALACIÓN DE ASTERISK Y COMPONENTES

Módulo 4

Configurar una PBX básica

Modulo 4: Conceptos

- En este módulo veremos lo siguiente:
 - Entender el flujo de llamadas en asterisk
 - Crear extensiones básicas.

Asterisk: flujo de llamadas

- Cada conversación, externa o interna son dos llamadas
 - Del teléfono a asterisk
 - De asterisk al destinatario
- En las siguientes diapositivas veremos el flujo de las llamadas de una manera simplificada. Para ampliar conocimientos, consultar el libro "Asterisk, The Future of telephony"

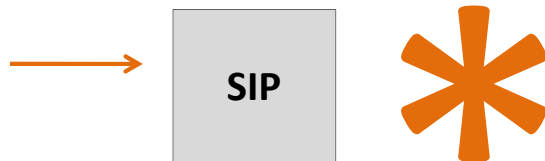
1.- El teléfono llama a asterisk

Diego descuelga el teléfono y marca 1000



2.- Driver Canal SIP

Hola, quiero hablar con la extensión 1000.
Se hablar G.711,GSM, etc



La llamada llega a un canal SIP de asterisk (mediante un mensaje INVITE)

3.- Canal SIP

Ok, hablaremos con el códec G.711 en el puerto ...



El canal SIP acepta la llamada y la comunicación queda establecida

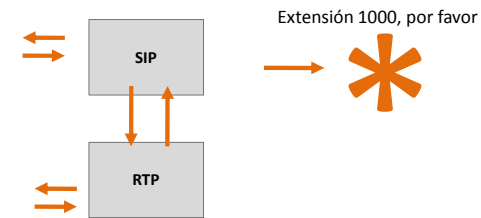
4.- Conexión establecida

El teléfono tiene ahora una llamada activa con un canal SIP



El canal SIP acepta la llamada y la comunicación queda establecida

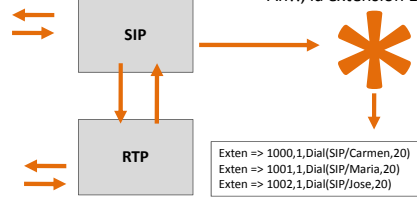
5.- la llamada entra en el Core



El canal SIP acepta la llamada y la comunicación queda establecida

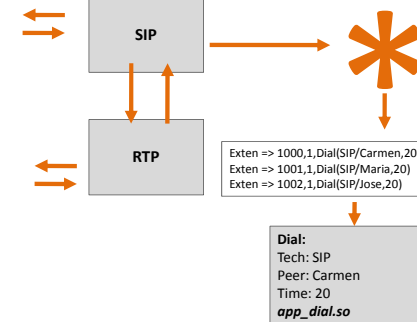
6.- El dialplan

Ah!!, la extensión 1000, consultamos el dialplan

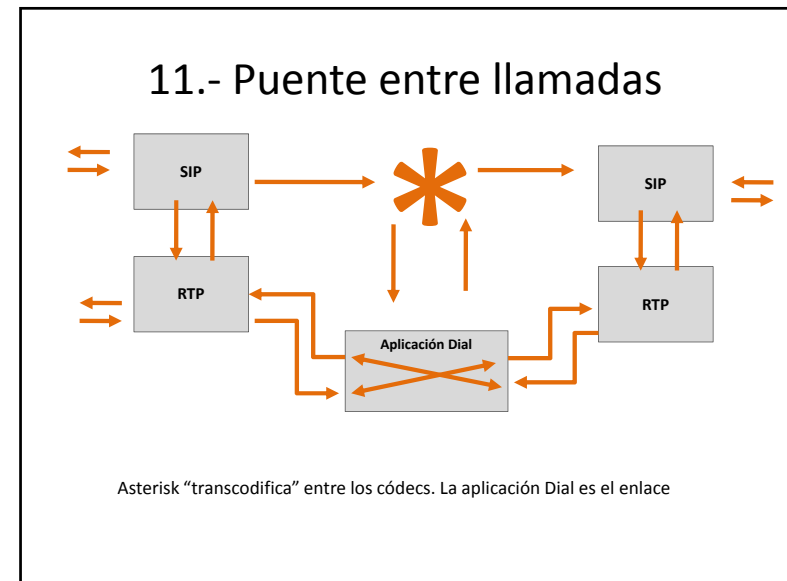
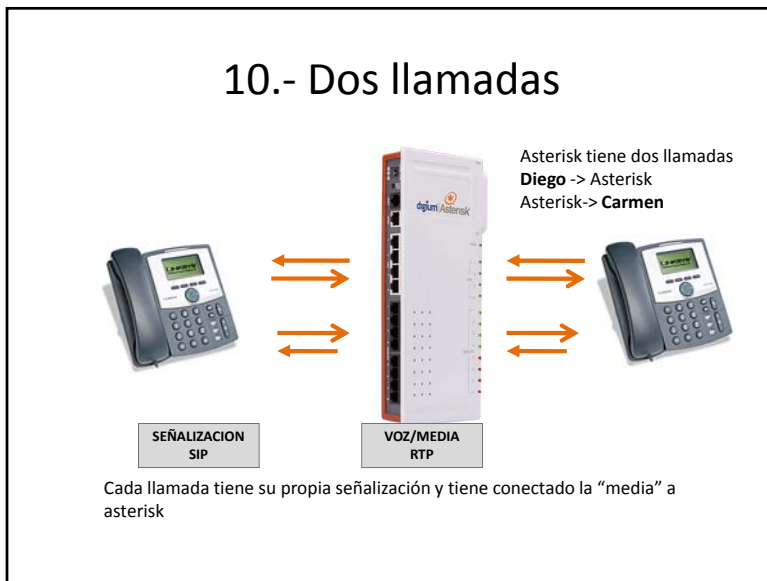
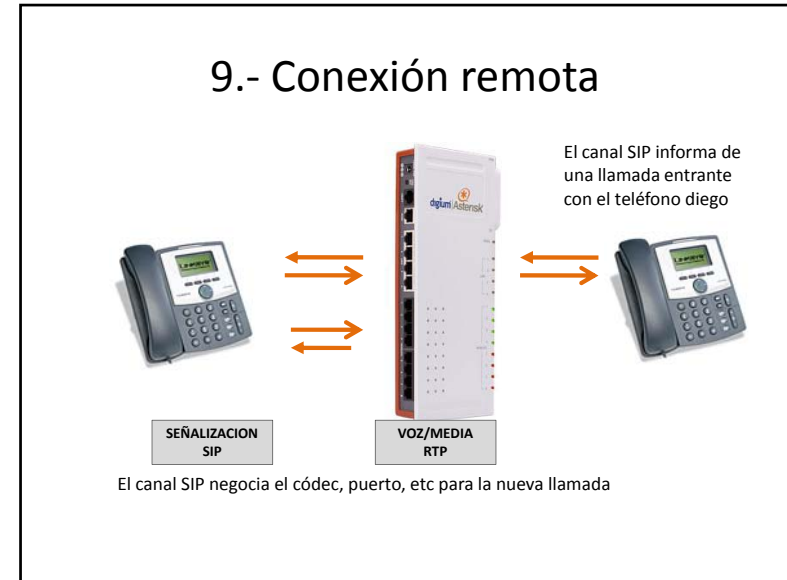
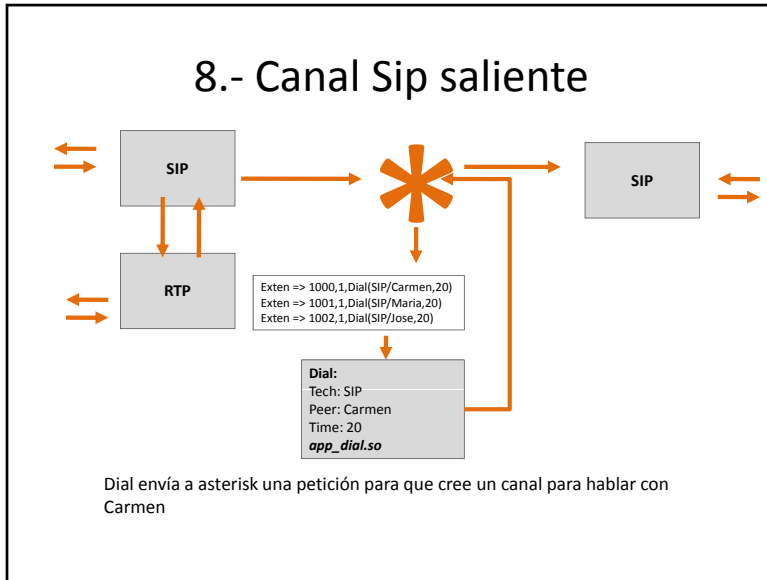


Asterisk busca la extensión en el dialplan

7.- la aplicación dial



LA extensión 1000 ejecuta la aplicación Dial, con SIP/Carmen como destino



12.- La llamada



Asterisk: Añadiendo un teléfono

- Lo primero que debemos hacer es crear el canal para el teléfono:
 - Si el teléfono está basado en SIP
 - Creamos la cuenta en el fichero sip.conf
 - Si el teléfono está basado en IAX/IAX2
 - Creamos la cuenta en iax.conf
 - Si el teléfono es un teléfono analógico
 - La línea analógica se configura en chan_dahdi.conf
 - La tarjeta analógica se configura en /etc/dahdi/system.conf

Asterisk: Añadiendo un teléfono SIP

- Editamos el fichero sip.conf
- Creamos la cuenta con los siguientes parámetros básicos

[mi_telefono]	; nombre o numero extensión
Type = friend	; el tipo de cuenta
Host = dynamic	; desde donde nos registramos
Context = users	; usaremos el contexto users
secret = password	; la contraseña

Asterisk: Añadiendo un teléfono SIP

- Que significan estos parámetros
- [mi_telefono]
 - Es el nombre de la cuenta. Cuando se configura un teléfono SIP, este es el usuario que se introduce en la configuración del teléfono
- Type = friend
 - Tipo de cuenta. "Friend" permite hacer y recibir llamadas
- Host = dynamic
 - Es la dirección IP desde la que se registra el teléfono. Si indicamos una, solamente podrá conectarse desde esa ip.
- Context = users
 - Cuando el teléfono hace una llamada, busca este contexto en el dialplan (extensions.conf)
- secret = password
 - La clave del dispositivo

Asterisk Lab

- Libro de ejercicios

ASTERISK LAB

MODULO 4

CONFIGURACIÓN DE ASTERISK

Módulo 5

El plan de llamadas

Módulo 5: conceptos

- En este módulo veremos lo siguiente:
 - Entender el dialplan
 - Uso básico de las aplicaciones del dialplan
 - Utilizar contextos, prioridades, extensiones para la realización básica de llamadas.

Asterisk: Que es el dialplan

- El dialplan de asterisk es mucho más potente y complejo que el dialplan de cualquier PDB convencional
- El dialplan es la parte mas dura de cualquier asterisk
- Provee la lógica de llamadas del sistema
- Consiste en una lista de instrucciones
 - Que hacer cuando una llamada entra?
 - Que hacer cuando una llamada sale?
 - Como el sistema interactúa entre los llamantes?

Dialplan: Estructura sintaxis

- Las instrucciones que forman parte del dialplan son organizadas en unidades lógicas
 - Contextos
 - Extensiones
 - Prioridades
 - Aplicaciones
- El dialplan es el fichero extensions.conf

Dialplan: Contexto

- Es el nivel superior de organización del dialplan
- Permite dividir el dialplan en bloques independientes
- Permite diferenciar las llamadas que puede hacer un usuario
- Sintaxis
 - Va siempre entre corchetes, no puede llevar espacios y caracteres [a-z,A-Z,0-9]
 - No puede tener una longitud superior a 80 caracteres

Dialplan : Contexto

- Un contexto puede contener otros contextos
- Para incluir un contexto utilizamos la palabra include

```
[users]
exten => 1000,1,Dial(SIP/1000)
exten => 1001,1,Dial(SIP/1001)
include => features
```

```
[features]
exten => 1009,1,VoiceMailMain()
```

- Se pueden realizar dialplans muy complejos utilizando includes

Dialplan: extensiones

- No es la extensión típica de una centralita convencional
- Son un conjunto de acciones que pueden finalizar en un teléfono
- Los identificadores son generalmente números
 - La extensión 1000 corresponde la teléfono de diego
- Sintaxis

```
exten => numero, prioridad, aplicación
exten => 1000,1,VoiceMailMain()
```

Dialplan: prioridades

- Cada acción de una extensión tiene un numero de prioridad
- Una extensión puede tener ilimitadas prioridades
- Cada proceso es ejecutado de forma secuencial
 - exten => 123,1, hacer algo ...
 - exten => 123,2, hacer otra cosa
 - exten => 123,3, hacer otra diferente
- Las prioridades deben comenzar siempre en 1
- No se pueden saltar números, siempre tienen que ser consecutivos
- Cuando asterisk encuentra un error en la prioridades (que no sean consecutivas) finaliza la llamada

Dialplan: prioridades

- Existe una prioridad especial, la "n", que incrementa en 1 el numero anterior
- Aunque se utilice la prioridad "n", la primera prioridad siempre tiene que ser 1
 - exten => 123,1, hacer algo ...
 - exten => 123,n, hacer otra cosa
 - exten => 123,n, hacer otra diferente

Dialplan: Aplicaciones

- Cada prioridad en el dialplan, está asociada a una aplicación
- Las aplicaciones mas comunes son
 - Contestar una llamada
 - Reproducir sonidos
 - Obtener información del numero llamante
 - Pasar la llamada a otro teléfono
 - Poner la llamada en espera
 - Buzón de voz
- Las aplicaciones pueden recibir parametros

Aplicaciones: Answer()

- Contesta a una llamada
 - exten => 123,1, Answer()
- No es necesario ponerla como primera prioridad, pero si es recomendable
- No tiene ningún parámetro

Aplicaciones: Playback()

- Playback() reproduce un sonido almacenado en el ordenador
- Ignora cualquier tono DTMF del teléfono
- Su primer parámetro es el nombre del fichero
- EL nombre no lleva extensión , porque asterisk es el encargado de seleccionar el mejor fichero en el caso de que los códec sean diferentes
- Sintaxis

```
exten => 123,1, Playback(hello-world)
```
- Por defecto los sonidos están almacenados en /var/lib/asterisk/sounds

Aplicaciones: Hangup()

- Finaliza la llamada actual
- Sintaxis

```
exten => 123,1, Hangup()
```
- Debe ser la ultima prioridad de cada extensión para evitar que queden llamadas abiertas.

Dialplan: aplicaciones

- Un ejemplo de nuestro primer dialplan

```
[users]
exten => 1009,1, Answer()
exten => 1009,2, Playback(hello-world)
exten => 1009,3, Hangup()
```

Asterisk Lab

- Libro de ejercicios

ASTERISK LAB
MODULO 5

PLAN DE LLAMADAS

Modulo 6

Plan de llamadas avanzado

Modulo 6: conceptos

- En este módulo veremos lo siguiente:
 - Extender el conocimiento sobre el dialplan
 - Crear menús de voz
 - Implementar los buzones de voz
 - Añadir una lógica simple al dialplan

Dialplan: interactividad

- Cuando un usuario interactúa con asterisk, se denomina IVR (Interactive Voice Response)
- La persona que llama puede introducir información presionando las teclas en el teléfono
- Los tonos que se escuchan son llamados tonos DTMF
- Asterisk puede interpretar esos tonos como decisiones del usuario
- DTMF:
 - Dual Tone Multi Frequency
 - Cada tecla crea dos tonos en diferentes frecuencias
 - Se suele denominar marcación por tonos

DTMF

	1209 Hz	1336 Hz	1447 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

Aplicaciones: Background()

- La aplicación Background() es similar a la aplicación Playback(), pero la primera acepta tonos DTMF
- Un tono DTMF para la reproducción del sonido
- Asterisk envía la llamada a la extensión que corresponda con los tonos DTMF
- Asterisk no envía la llamada hasta que no esta completamente seguro de que la extensión no es ambigua

Aplicaciones: Background()

- La aplicación Background() cuelga la llamada al finalizarse su locución.
- Un ejemplo


```
[users]
exten => 1000,1, Answer()
exten => 1000,2, Background(menu)
exten => 1,1, Playback(digits/1)
exten => 2,1, Playback(digits/2)
```
- Que pasa si el usuario pulsa 2?
- Que pasa si el usuario pulsa 1?
 - Asterisk no sabe si quieres la extensión 1000 o la extensión 1

Aplicaciones: Background() y Timeouts

- La aplicación Background() espera por dos tipos de timeout
 - Response timeout, el usuario no pulso ninguna tecla, y la locución del audio ha finalizado
 - Digit timeout, el tiempo de espera máximo en tre pulsaciones de teclas
- También hay un timeout absoluto
 - Es la duración máxima de la llamada

Aplicaciones: WaitExten()

- La aplicación WaitExten() se utiliza para tener un mayor control del timeout de la aplicación Background
- Recibe como parámetro el numero de segundos que esperamos por la respuesta
- Puede contener decimales y se expresa en segundos

```
exten => 1000,n, WaitExten(1.5)
```

Extensiones especiales

- Existen extensiones especiales para el caso de que el usuario no haga nada
- Estas extensiones son de una sola letra, por lo que no deberíamos de tener en nuestro dialplan ninguna extensión de una letra

Extensiones especiales

- Estas extensiones son
 - i, extensión invalida
 - t, response timeout de la extensión
 - T, timeout absoluto
 - h, la extensión para colgar
 - s, la extensión de inicio
 - o, la extensión del operador
 - a, la extensión del asistente

Extensiones especiales: i

- La extensión “i” significa invalida
- Cuando el usuario introduce una respuesta inválida a las aplicaciones Background o WaitExten, la llamada se envía a la extensión “i”

```
exten => i,1, Playback(invalid)
exten => i,n, Hangup()
```

Extensiones especiales: t

- La extensión “t” significa timeout
- Cuando el usuario espera mucho tiempo en dar una respuesta a las aplicaciones Background o WaitExten, la llamada se envía a la extensión “t”

```
exten => t,1, Playback(goodbye)
exten => t,n, Hangup()
```

Extensiones especiales: T

- La extensión “T” significa timeout absoluto
- Cuando la duración de la llamada excede del valor absoluto

```
exten => T,1, Playback(goodbye)
exten => T,n, Hangup()
```

Extensiones especiales: h

- La extensión “h” significa colgar
- Cuando un usuario cuelga en cualquier contexto, se está ejecutando la extensión “h”
- Es útil si queremos realizar alguna acción en el momento de que el usuario cuelgue
 - Por ejemplo, registrar la duración de la llamada en una base de datos

Extensiones especiales: s

- La extensión “s” significa iniciar(start)
- Cuando una llamada entra en un contexto sin una extensión específica, esta llamada se envía a la extensión “s”
 - Este es el caso más frecuente en las líneas analógicas

```
exten => s,1, Answer()
exten => s,2, Playback(hello-world)
```

Aplicaciones: Goto()

- Nos permite saltar de un lugar a otro del dialplan
- Puede tener entre 1 y tres parámetros
 - Prioridad (dentro de la misma extensión)
 - Extensión y prioridad (dentro del mismo contexto)
 - Contexto, extensión y prioridad (en cualquier lugar del dialplan)

Aplicaciones: Goto()

- Ejemplo de los tres casos

```
[goto-demo]
exten => 123,1, Playback(hello-world)
exten => 123,2, goto(1)
```

```
exten => 124,1, Playback(hello-world)
exten => 124,2, goto(123,1)
```

```
exten => 125,1, Playback(hello-world)
exten => 125,2, goto(goto-demo ,124,1)
```

Aplicaciones: Goto() y labels

- Ejemplo con labels

```
exten => 123,1, Answer()
exten => 123,n, WaitExten(1)
```

```
exten => 123,n(mus), Playback(musica)
exten => 123,n, Goto(mus)
```

Aplicaciones: Otras aplicaciones

- Wait()
 - Espera el numero de segundos especificado
- NoOp()
 - Significa “No Operation”
 - Se usa como herramienta de test para ver en el CLI los pasos que está dando en el dialplan
- Verbose
 - El nivel de verbose requerido para mostrar el mensaje
 - EL mensaje a mostrar en la pantalla

Aplicaciones: Dial()

- La aplicación Dial() permite unir un extremo (teléfono) con otro extremo (teléfono, centralita)
- Dial() inicia la llamada, y es la encargada del progreso (ocupado, no hay respuesta, etc)
- Si la llamada es contestada, Dial() une los dos extremos
- Dial() es la aplicación mas importante del dialplan

Aplicaciones: Dial()

- Dial() recibe cuatro parámetros
 - Dispositivo
 - SIP/1000, SIP/Diego, IAX/asterisk
 - DADHI/4/981373199
 - Timeout
 - Cuanto tiempo va a esperar antes de moverse a la siguiente prioridad en el dialplan
 - Opciones
 - Mensajes en pantalla, tonos específicos
 - Tecléea en CLI> core show application dial para ver una lista completa
 - URL
 - Envía una URL a el display del teléfono
 - Raramente es utilizado

Aplicaciones: Dial()

- Ejemplos

exten => 123,n, Dial(SIP/123)

exten => 123,n, Dial(SIP/123,30)

exten => 123,n,Dial(SIP/123,30,m)

exten => 123,n,Dial(SIP/123,,m)

Aplicaciones: VoiceMail()

- Si una llamada no es contestada, se puede desviar al buzón de voz.
- Existen dos aplicaciones
 - VoiceMail() , que permite a los usuarios dejar mensajes en el buzón de voz
 - VoiceMailMain() que permite a los usuarios autenticados recuperar sus mensajes y configurar opciones
- Los buzones de los usuarios se definen en el fichero voicemail.conf

Aplicaciones: VoiceMail()

- VoiceMail() tiene dos argumentos
 - El buzón en formato numero@contexto
 - Si no se especifica el contexto, se utiliza [default]
 - Las opciones
 - u – Reproduce el mensaje “unavailable”
 - b – Reproduce el mensaje “bussy”
 - s - no reproduce las instrucciones
- Ejemplo

exten => 123,n, VoiceMail(123@default,b)

Aplicaciones: VoiceMailMain()

- VoiceMailMain() tiene dos argumentos
 - El buzón en formato numero@contexto
 - Si no se especifica el contexto, se utiliza [default]
 - Las opciones
 - s - no solicitar la clave al usuario
- Ejemplo


```
exten => 123,n, VoiceMailMain(123@default)
```

Ficheros: voicemail.conf

- Los buzones de voz se definen en este fichero
- La sección [general] contiene opciones globales para todos los buzones
- La sección [zonemessages] define las zonas que :
 - Indican la franja horaria
 - Indican la forma de reproducir la hora, fecha, etc
- Al igual que el dialplan, el fichero voicemail.conf puede tener múltiples contextos.

Aplicaciones: Record()

- La aplicación Record() nos permite grabar un mensaje en el sistema para poder reproducirlo en nuestro dialplan
- La calidad no es perfecta, pero con un buen teléfono se consiguen calidades altas
- Ejemplo


```
exten => 123,n, Record (nombrefichero.wav)
```

Aplicaciones: Record()

- La aplicación Record() finaliza pulsando la tecla "#", y continua con la siguiente prioridad
- Soporta la grabación en todos los formatos que soporte asterisk
- Para consultar los formatos:

```
CLI> core show file formats
```

Asterisk Lab

- Libro de ejercicios

ASTERISK LAB

MODULO 6

PLAN DE LLAMADAS AVANZADO

GRACIAS