

## LABORATORIO DIDATTICO 1

### Installazione di un Communication Server con funzione di centralino telefonico IP (IP-PBX) e configurazione di base per chiamate interne a una rete privata

#### 1. Premessa

Le attività di installazione e configurazione di un communication server che svolge la funzione di IP-PBX (centralino telefonico in tecnica VoIP) sono suddivise in tre laboratori didattici:

- Laboratorio 1 -> installazione e configurazione di base del server Asterisk
- Laboratorio 2 -> installazione e configurazione di softphone e telefoni IP a standard SIP (Session Initiation Protocol)
- Laboratorio 3 -> analisi delle chiamate e introduzione alle problematiche di sicurezza

Questi laboratori possono essere realizzati anche utilizzando i pacchetti software gratuiti qui di seguito elencati.

- Sul PC che funge da Communication server / IP-PBX, che denominiamo **Server Asterisk**, è consigliabile installare e/o verificare che siano attivi i seguenti software:
  - AsteriskNOW
  - Server FTP (come VSFTP), per trasferire file e cartelle da/a un altro computer
  - Client TFTP, per fare il backup dei file di configurazione su un'altro computer
  - server SSH, per poter configurare da remoto il server (può essere già attivo di default)
  - editor di testo in ambiente LINUX, come per esempio **nano** (oppure vim, vi, ecc).
- Sui PC, sui tablet e/o sugli smartphone utilizzati come client vanno installati:
  - Softphone a standard SIP;
  - microfono e altoparlante o, meglio, cuffie con microfono;
  - webcam, per testare le comunicazioni in audio e video (opzionale);
  - Wireshark, analizzatore di protocollo (opzionale).
- Sul PC che si impiega per configurare da remoto il server Asterisk, che denominiamo **PC-ADMIN**, e su cui si salvano, come backup, i file di configurazione e altri file utili è consigliabile installare:
  - Softphone a standard SIP;
  - Wireshark, analizzatore di protocollo;
  - PuTTY, client SSH per Windows e Linux;
  - Server TFTP, per esempio Solarwinds TFTP Server ([www.solarwinds.com](http://www.solarwinds.com));
  - Client FTP, per esempio Filezilla (<https://filezilla-project.org>).

E' poi consigliabile completare la dotazione di laboratorio acquistando almeno un telefono IP hardware.

Per la sola comunicazione in audio è sufficiente un PC con cuffie dotate di microfono o un telefono IP hardware, mentre per la videocomunicazione tra gli interni si possono utilizzare softphone installati su PC dotati di webcam e microfono.

Come softphone, tra gli altri, si possono impiegano i seguenti pacchetti software (attualmente gratuiti in versione base, con alcune limitazioni rispetto alle versioni a pagamento):

- X-LITE free, <http://www.counterpath.com/x-lite> (ambiente Windows)
- 3CXPHONE for VoIP providers, [www.3cx.com/voip/softphone/](http://www.3cx.com/voip/softphone/) (ambiente Windows)

- Zoiper, [www.zoiper.com](http://www.zoiper.com) (preferibilmente in ambiente Linux, ma è anche disponibile per Windows e smartphone)
- Ekiga, <http://ekiga.org> (preferibilmente in ambiente Linux, ma è anche disponibile per Windows).

Per analizzare nel dettaglio l'evoluzione di una chiamata in audio e/o video si impiega l'analizzatore di protocollo Wireshark, che consente anche di evidenziare le problematiche di sicurezza insite nella telefonia su IP.

## 2. Installazione del server Asterisk

In questo LABORATORIO DIDATTICO si propone l'installazione di un IP-PBX, o meglio di un *communication server*, realizzato il software open source gratuito **Asterisk** ([www.asterisk.org](http://www.asterisk.org)) che, fra l'altro, consente chiamate telefoniche e videochiamate impiegando, per esempio, *softphone* (telefoni software) installati su PC dotati di webcam e/o di cuffia con microfono, su smartphone o tablet connessi in rete per esempio via Wi-Fi, nonché di telefoni IP che supportano il protocollo SIP.

Per semplificare l'installazione faremo ricorso a distribuzioni Linux che offrono Asterisk già compilato. A questo scopo vi sono almeno le seguenti possibilità:

- installare AsteriskNOW, la distribuzione Linux basata su CentOS e contenente Asterisk, scaricabile dal sito ufficiale di Asterisk ([www.asterisk.org/downloads/asterisknow](http://www.asterisk.org/downloads/asterisknow));
- installare il pacchetto Asterisk da un'altra distribuzione Linux, come Ubuntu, tramite il gestore pacchetti;
- installare Asterisk partendo dai sorgenti scaricabili dal sito [www.asterisk.org](http://www.asterisk.org) su un computer con sistema operativo Linux come CentOS ([www.centos.org](http://www.centos.org)).

Per evidenziare meglio i principi su cui si basa il funzionamento di Asterisk e la sua configurazione di base effettueremo l'installazione e le prove di comunicazione all'interno di una rete privata (una LAN), inoltre opereremo a linea di comando (CLI, *Command Line Interface*) concentrandoci sui seguenti aspetti:

- configurazione iniziale del server
- configurazione di base del piano di numerazione (*Dialplan*);
- trattamento delle chiamate;
- configurazione di un servizio di trasferimento di chiamata ad altro numero nel caso in cui l'utente chiamato non sia raggiungibile o sia occupato;
- implementazione di un servizio Text To Speech (TTS) usabile da Asterisk
- configurazione di un servizio di segreteria telefonica (voicemail o casella vocale), con indicazione sul telefono della presenza di messaggi nuovi;
- configurazione di softphone e telefoni IP operanti con protocollo SIP.

### AVVERTENZA

Per semplicità in questo laboratorio didattico non sono state messe in atto particolari politiche di sicurezza a protezione del server, dei dati in esso contenuti, dei client e delle informazioni scambiate fra essi, in quanto si è in una fase di sperimentazione e di test.

Nel caso in cui il server Asterisk fosse effettivamente impiegato come IP-PBX aziendale è invece necessario applicare adeguate politiche di sicurezza.

### 2.1 Installazione di AsteriskNOW

I passi con cui si può effettuare l'installazione di AsteriskNOW (qui inizialmente nella versione 3.0.1) sono essenzialmente i seguenti (per dettagli e modifiche successiva alla stesura di questo

laboratorio si veda la **Quick Start Guide**,  
<https://wiki.asterisk.org/wiki/display/AST/Installing+AsteriskNOW>):

1. Scaricare l'immagine ISO del software asteriskNOW dal sito <http://www.asterisk.org/downloads/asterisknow>
2. masterizzare l'immagine ISO su un DVD;
3. verificare i requisiti hardware e avviare il PC avendo cura che il boot sia da CD/DVD
4. seguire le istruzioni di installazione disponibili sul sito sotto la voce **Quickstart guide**
5. installare AsteriskNOW con FreePBX in modo da avere la possibilità di utilizzare anche l'interfaccia grafica *FreePBX* ([www.freepbx.org](http://www.freepbx.org)) e avere un sistema in cui è sufficiente personalizzare alcuni file di configurazione;
6. selezionare la **System Time Zone** (Rome);
7. impostare la **password di root** (amministratore; usare i criteri delle password sicure: minimo 8 caratteri contenenti lettere minuscole e maiuscole, numeri e caratteri speciali);
8. scegliere la versione di Asterisk (per esempio la 11) e di FreePBX da installare;
9. al termine dell'installazione togliere il DVD e riavviare il sistema;
10. effettuare il primo **login** come utente *root* (amministratore), inserendo la password configurata nella fase di installazione; è consigliabile poi creare un altro utente non amministratore con cui effettuare gli accessi successivi.

Dato che il layout della tastiera è impostato su quella americana (**us**), per cui diversi caratteri (segni di interpunzione, ecc.) non corrispondono con quelli della tastiera italiana, è consigliabile configurare il layout della tastiera italiana operando con i diritti di amministratore (*root*) nel seguente modo (FIGURA 1):

1. effettuare il login al server Asterisk, anche da remoto con PuTTY;
2. spostarsi nella directory */etc/sysconfig* (**cd /etc/sysconfig**);
3. aprire con un editor di testo (nano) il file *keyboard* (**nano keyboard**);
4. modificare **KEYTABLE = "us"** in **KEYTABLE = "it"** e **LAYOUT = "us"** in **LAYOUT = "it"**;
5. **LAYOUT = "it"**;
6. salvare (ctrl O), uscire dall'editor di testo (ctrl X) e riavviare il server Asterisk (**reboot**).

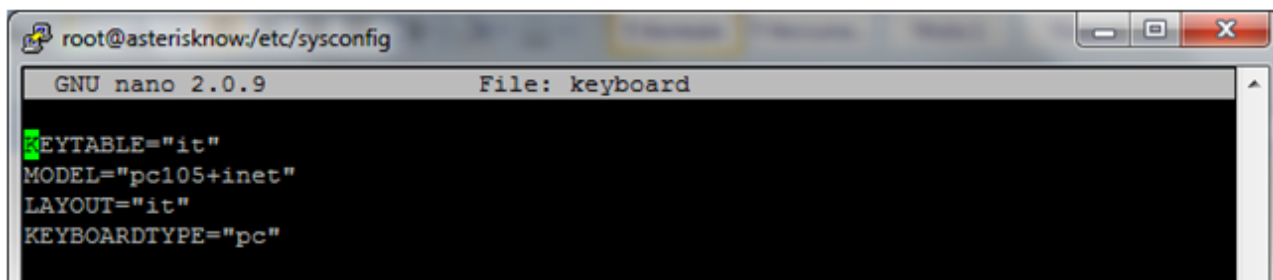


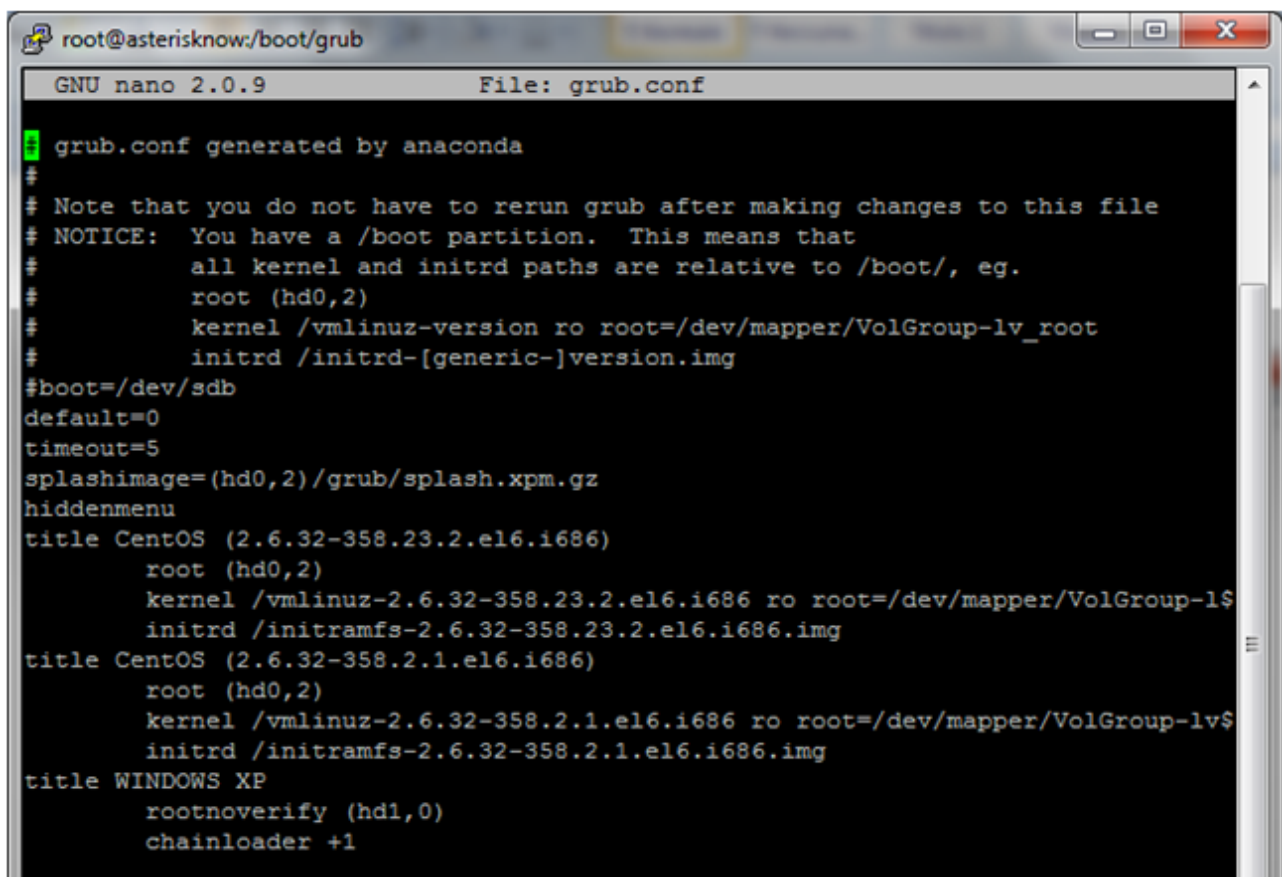
FIGURA 1 Configurazione della tastiera italiana

A partire dalla versione 3.0.x di AsteriskNOW è possibile aggiornare il server (upgrade) a una versione successiva senza dover reinstallare tutto, seguendo le indicazioni fornite dalla guida, che attualmente sono all'indirizzo  
<https://wiki.asterisk.org/wiki/display/AST/Upgrading+AsteriskNOW>

Se, come avviene con la versione AsteriskNOW 3.0.1, il sistema operativo CentOS 6.5 può non essere l'unico sistema operativo installato sul PC, è possibile modificare il sistema operativo con cui si avvia di default il server editando il file di configurazione del boot loader **grub** (FIGURA 2):

1. ci si sposta nella directory */boot/grub* (**cd /boot/grub**),

2. si apre con un editor di testo il file grub.conf (**nano grub.conf**) e si modifica il sistema operativo da avviare di default (identificato, partendo da 0, dal numero del TITLE); è anche possibile cambiare l'intervallo di tempo da aspettare prima di avviare il sistema di default.



```
root@asterisknow:/boot/grub
GNU nano 2.0.9 File: grub.conf
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE: You have a /boot partition. This means that
#           all kernel and initrd paths are relative to /boot/, eg.
#           root (hd0,2)
#           kernel /vmlinuz-version ro root=/dev/mapper/VolGroup-lv_root
#           initrd /initrd-[generic-]version.img
#boot=/dev/sdb
default=0
timeout=5
splashimage=(hd0,2)/grub/splash.xpm.gz
hiddenmenu
title CentOS (2.6.32-358.23.2.el6.i686)
    root (hd0,2)
    kernel /vmlinuz-2.6.32-358.23.2.el6.i686 ro root=/dev/mapper/VolGroup-lv$
    initrd /initramfs-2.6.32-358.23.2.el6.i686.img
title CentOS (2.6.32-358.2.1.el6.i686)
    root (hd0,2)
    kernel /vmlinuz-2.6.32-358.2.1.el6.i686 ro root=/dev/mapper/VolGroup-lv$
    initrd /initramfs-2.6.32-358.2.1.el6.i686.img
title WINDOWS XP
    rootnoverify (hd1,0)
    chainloader +1
```

FIGURA 2 Configurazione del boot loader nel caso di più sistemi operativi presenti sul PC

Con le versioni successive alla 3.0.1, invece, AsteriskNOW e CentOS vengono invece installati su un hard disk a essi dedicato.

## 2.2 Configurazione di un indirizzo IP statico e attivazione del client tftp

E' consigliabile assegnare un indirizzo IP statico al Server Asterisk, in modo da sapere sempre qual è il suo indirizzo IP. Con le ultime versione di AsteriskNOW ciò può essere fatto in fase di installazione del sistema.

Con la versione 3.0.1 di AsteriskNOW si può anche operare nel seguente modo:

1. effettuato il login sul server Asterisk, con i diritti di amministratore si digita **sudo setup** (FIGURA 3);
2. si seleziona **Network configuration**;
3. si deseleziona DHCP e si assegnano indirizzo IP, subnet mask, Default Gateway e server DNS (FIGURA 4).

N.B. Accertarsi che l'indirizzo statico scelto non sia tra quelli a disposizione del server DHCP della rete privata a cui il server Asterisk appartiene.

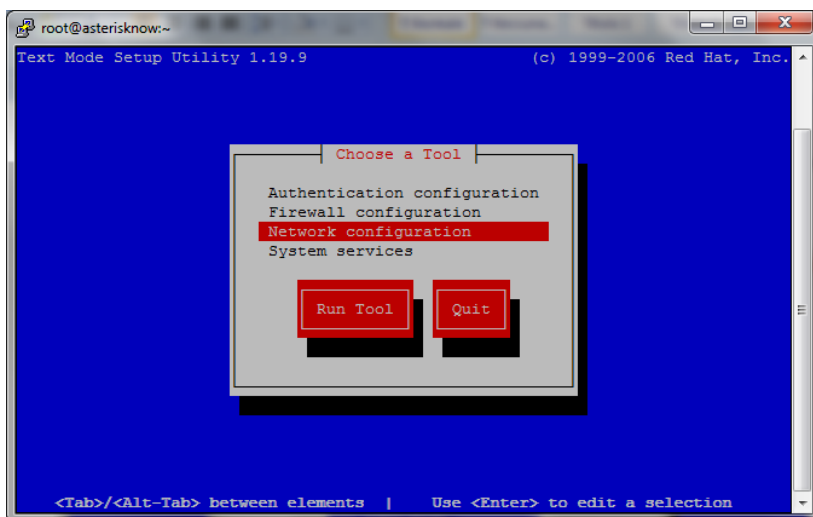


FIGURA 3 Setup iniziale

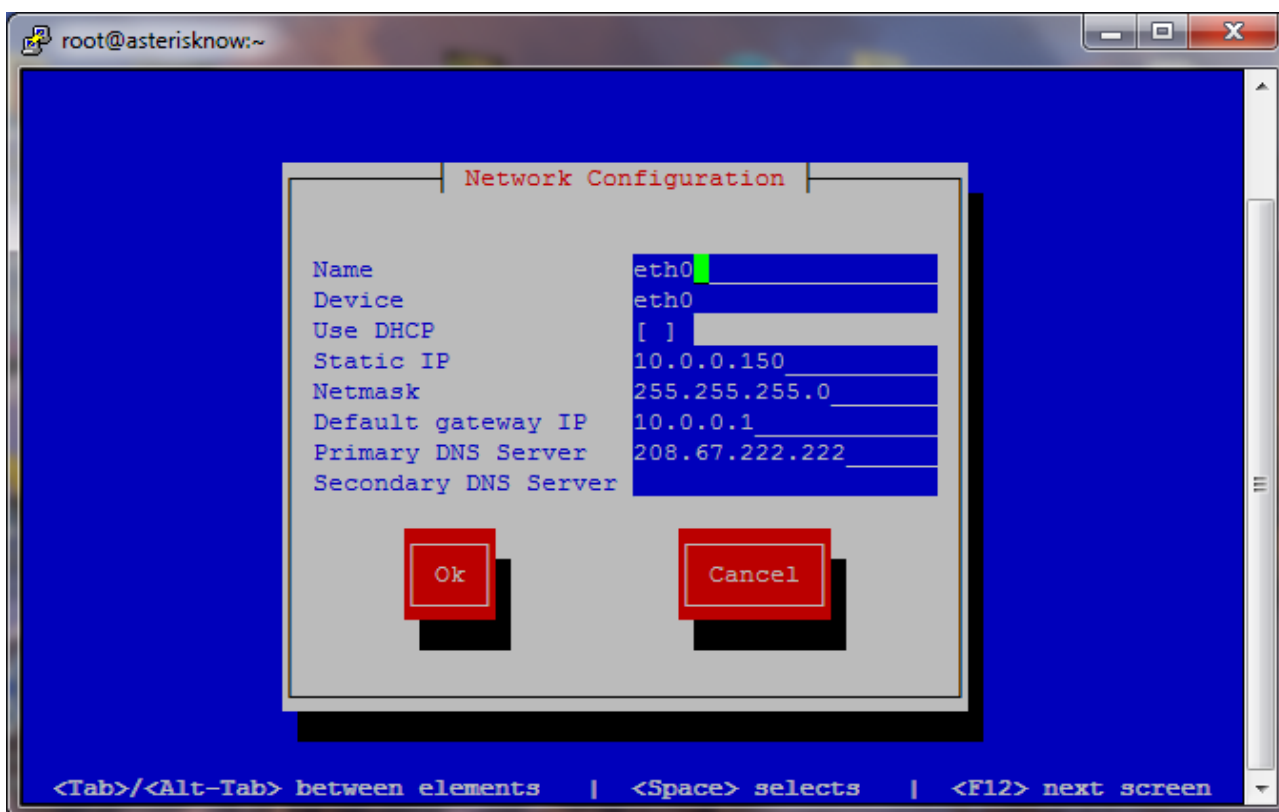


FIGURA 4 Configurazione dell'indirizzo statico

Per attivare i servizi desiderati, tra cui il client TFTP e, se installato, il server FTP (vsftpd), si seleziona *System services* e si marca (FIGURA 5):

- **tftp** per abilitare il *client tftp* (*Trivial File Transfer Protocol*) in modo da poter fare un backup della configurazione su un server tftp collegato in rete o ripristinare una configurazione salvata;
- **vsftpd** per abilitare il server FTP (da configurare, i file di configurazione sono nella directory */etc/vsftpd*), in modo da poter accedere ai file del server dal PC-ADMIN tramite un client FTP, come per esempio Filezilla.
- si seleziona (tasto TAB) **Ok** e **Save&Quit**

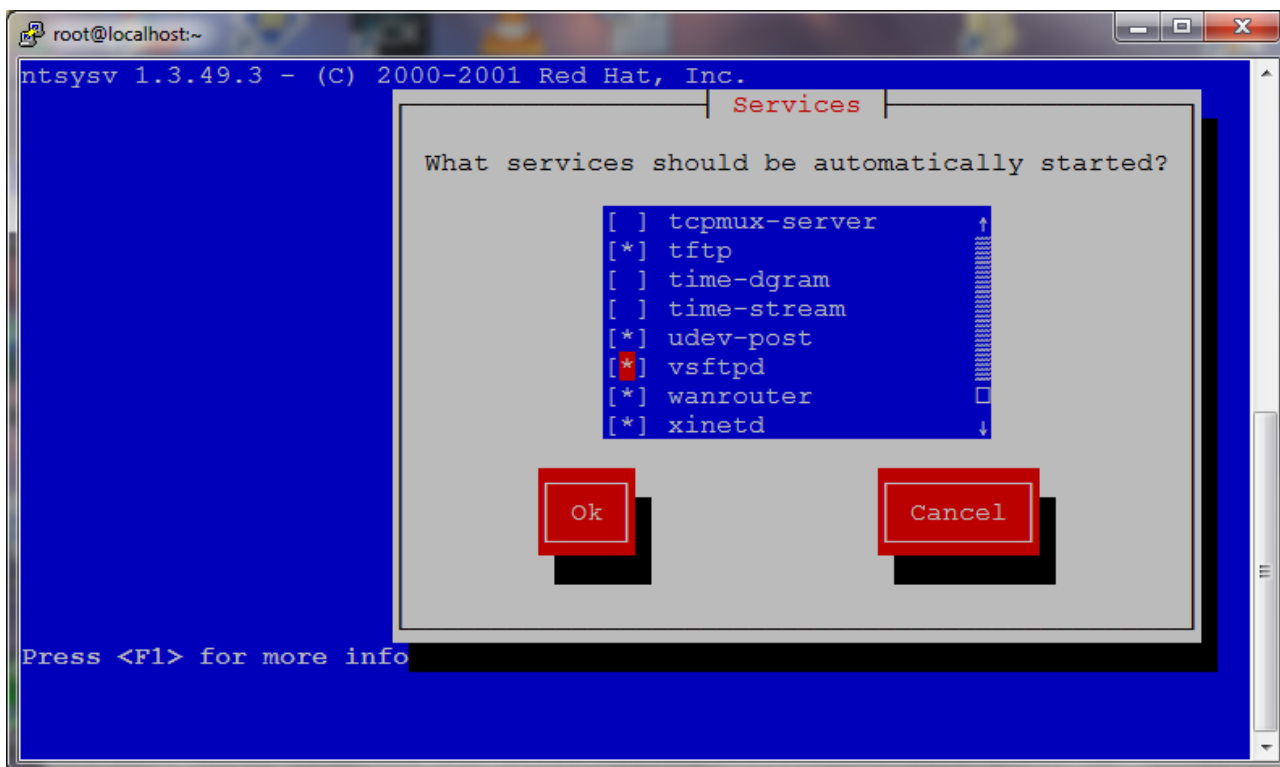


FIGURA 5 Attivazione del client TFTP e del server FTP (AsteriskNOW 3.0.1)

Nel caso in cui il comando setup non sia attivo è possibile installare il client tftp e il server vsftpd con il seguente comando: **sudo yum install tftp vsftpd**

Si verifica quindi che il server sia attivo con il comando: **sudo service vsftpd status** (è possibile avviare il servizio con **sudo service vsftpd start** o riavviarlo con **sudo service vsftpd restart**).

Andando nella directory /etc/vsftpd (cd /etc/vsftpd) è possibile editare i file di configurazione (**sudo nano vsftpd.conf**) in modo da personalizzare il servizio FTP (tra l'altro si vedano le istruzioni contenute come commenti nei file **ftpusers** e **user\_list**).

#### AVVERTENZA

Se si utilizza il servizio FTP, dopo aver copiato dei file sul server Asterisk è consigliabile verificare che essi abbiano i permessi corretti e che appartengano all'utente asterisk con il comando **ls -l** (oltre che tramite un client FTP come Filezilla, i permessi possono essere modificati con il comando **sudo chmod <nuovipermessi> <nomefile>** mentre l'utente può essere cambiato con il comando **sudo chown asterisk:asterisk <nomefile>**).

E' consigliabile quindi aggiornare il sistema operativo CentOS con il comando **sudo yum update**. Riavviare il PC con il comando **reboot** (oppure **shutdown -r now**)

### 3 Principali file di configurazione di AsteriskNOW

Con AsteriskNOW per la configurazione di base che andiamo a fare è consigliabile editare almeno i seguenti file:

- **sip\_custom.conf**, file incluso in **sip.conf** e contenente le impostazioni da configurare sui softphone e sui telefoni IP affinché possano registrarsi ed essere serviti;

- **extensions\_custom.conf**, file incluso in extensions.conf, per la configurazione del *Dialplan* e il trattamento delle chiamate,
- **voicemail.conf**, file di configurazione per le caselle vocali (segreteria telefonica).

Se invece si opera con una distribuzione diversa, o installando direttamente da sorgente tutti i pacchetti di Asterisk, allora i principali file da editare possono essere direttamente i seguenti:

- **sip.conf**,
- **extensions.conf**
- **voicemail.conf**.

### 3.1 Primo backup dei file di configurazione

Sul PC-ADMIN collegato in rete, con cui si amministra il server Asterisk da remoto, e di cui è noto l'indirizzo IP (per esempio 10.0.0.59) si installa un server TFTP (*Trivial File Transfer Protocol*), per esempio *Solarwinds TFTP server* (attualmente scaricabile dal sito [www.solarwinds.com](http://www.solarwinds.com)). Aprendo il prompt dei comandi, dal PC si fa un ping per verificare la raggiungibilità del server Asterisk (avente per esempio indirizzo IP 10.0.0.150): C:\>ping 10.0.0.150.

Si apre quindi il server TFTP e lo si configura cliccando su (FIGURA 6):

- *File, Configure, General* per
  - visualizzare (ed eventualmente cambiare) la directory (Storage) in cui il server TFTP copia i file ricevuti o da cui preleva i file da trasmettere, che di default è C:\>TFTP-Root;
  - consentire la modifica del nome del file ricevuto nel caso in cui sia uguale a uno già esistente nella directory TFTP-Root (opzionale).
- *File, Configure, Security* per abilitare sia la trasmissione sia la ricezione di file (*Send and Receive files*); è anche possibile decidere se i file possono essere trasmessi e ricevuti da qualsiasi computer oppure da un computer avente un ben preciso indirizzo IP (come per esempio il 10.0.0.150 se l'unico PC con cui si scambiano files con protocollo TFTP è il server Asterisk).

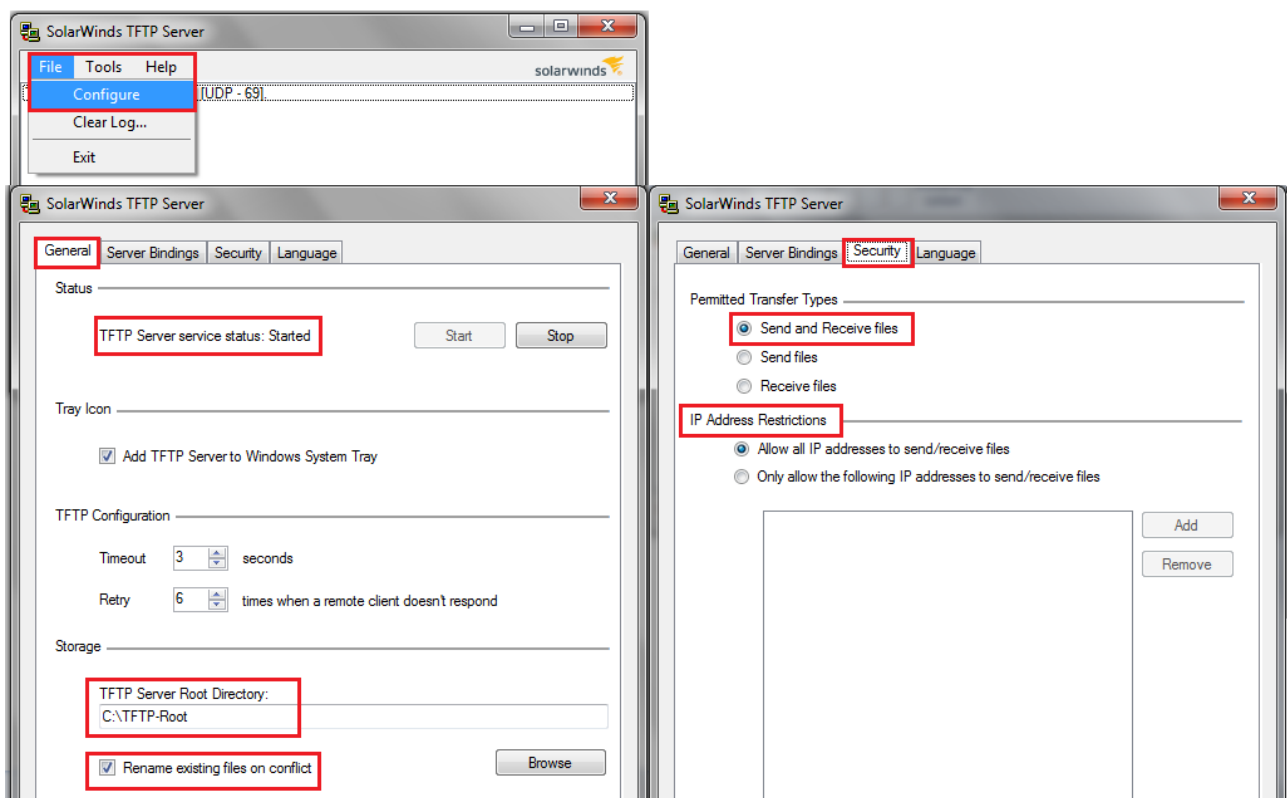


FIGURA 6 Configurazione del server TFTP Solarwinds

Dal PC-ADMIN si apre PuTTY (FIGURA 7), si effettua il login con protocollo SSH e ci si porta sulla directory /etc/asterisk (**cd /etc/asterisk**), si apre il client TFTP e con il comando **put <nomefile>** si fa un primo backup dei file di configurazione che editeremo digitando (FIGURA 8):

```
[root@asterisknow ~]# cd /etc/asterisk
[root@asterisknow asterisk]# tftp 10.0.0.59
tftp> put sip_custom.conf
tftp> put extensions_custom.conf
tftp> put voicemail.conf
tftp> quit
[root@asterisknow asterisk]#
```

Se non si è abilitata l'opzione per rinominare i file ricevuti in caso di conflitto, per non sovrascrivere versioni successive dei file di backup è possibile, per esempio, creare in TFTP-Root delle cartelle aventi come nome la revisione o la data in cui si è operato il backup e spostare in esse i file di backup.

Per scaricare un file dal server TFTP sul server Asterisk, ripristinando così una configurazione che era stata salvata o preparata sul PC-ADMIN, si digita invece il comando **get <nome file>**:

```
[root@asterisknow backup]# tftp 10.0.0.59
tftp> get sip_custom.conf
tftp> get extensions_custom.conf
tftp> get voicemail.conf
tftp> quit
```

Nel caso in cui i file siano stati rinominati è necessario ripristinare il nome originario prima di scaricare nuovamente i file sul Server Asterisk.

Si ricorda che per applicare le modifiche apportate ai file di configurazione di Asterisk va riavviato il servizio asterisk con il comando<sup>1</sup> **service asterisk restart**.

---

<sup>1</sup> In alternativa è possibile operare il caricamento della nuova configurazione impiegando la console di asterisk, apribile digitando da terminale **Asterisk -rv** e immettendo i comandi di reload appropriati.

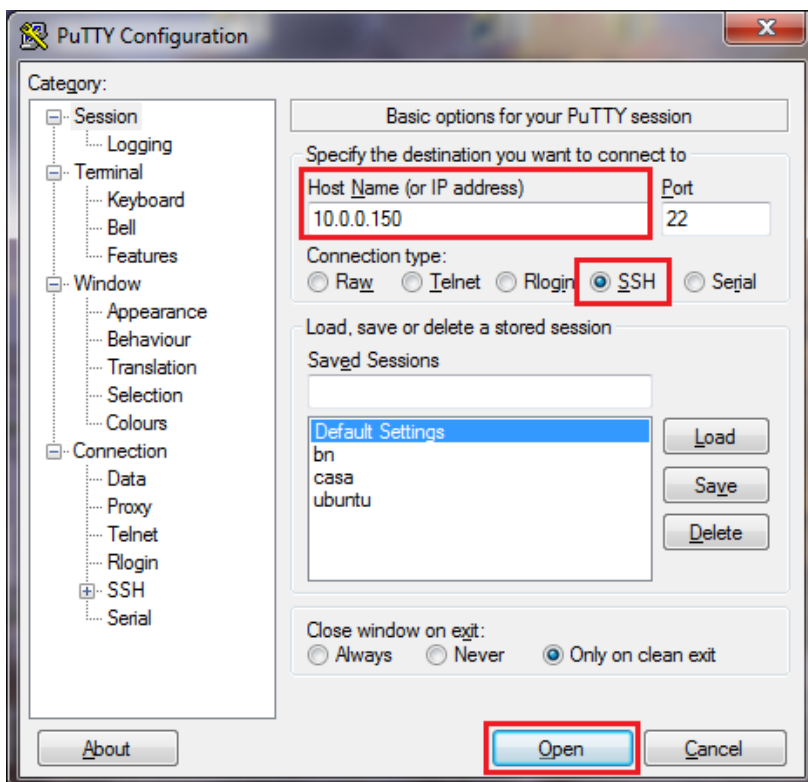


FIGURA 7 Client SSH PuTTY

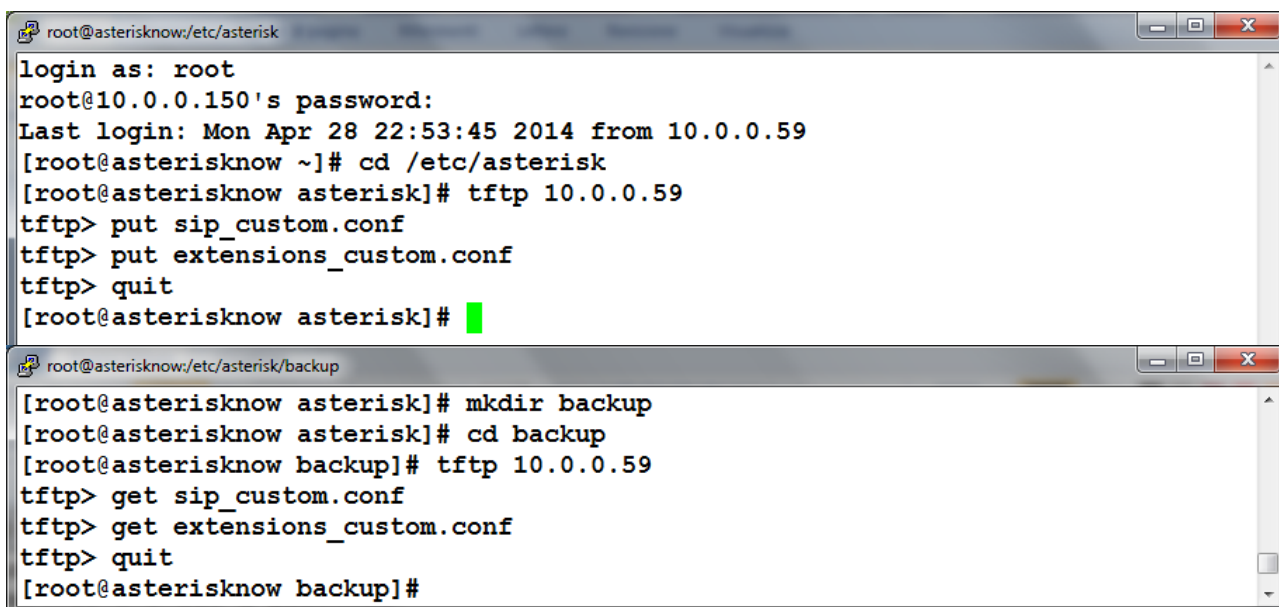


FIGURA 8 Put: invio del file al server TFTP (upload); Get: si scarica il file dal server TFTP (download)

Se si desidera utilizzare un editor di testo grafico, come wordpad, notepad, gedit, ecc. possiamo fare l'upload dei file di configurazione nella directory di storage del server TFTP (per esempio TFTP-Root), farne una copia di riserva, ed editarli (per esempio con notepad) per inserire le configurazioni e le istruzioni desiderate.

Al termine della configurazione, si salvano i file e li si scarica (get) sul server Asterisk via TFTP come visto in precedenza. Per applicare le modifiche si riavvia il servizio asterisk (con i diritti di amministratore) con il comando: **sudo service asterisk restart**.

Il servizio TFTP è semplice e veloce da utilizzare e da configurare per trasferire i file di configurazione.

Per poter caricare e scaricare file e cartelle anche di grosse dimensioni (voci, file musicali, ecc.) dal/al server Asterisk può risultare conveniente utilizzare il protocollo FTP, messo a disposizione dal servizio **vsftpd** sul server Asterisk. Se il servizio vsftpd è stato correttamente configurato, è così possibile trasferire file e cartelle utilizzando sul PC-ADMIN un client FTP, come per esempio FileZilla (FIGURA 9).

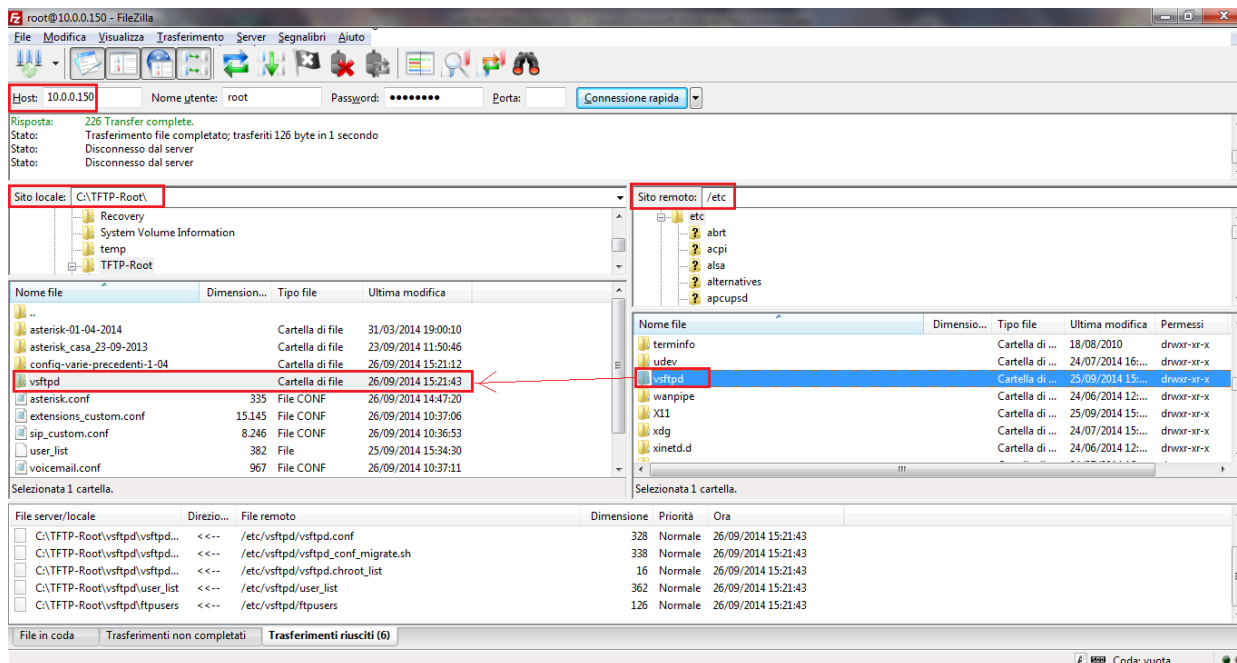


FIGURA 9 Trasferimento di una cartella (la directory vsftpd posta /etc) da Server Asterisk a PC-ADMIN con protocollo FTP.

### 3.2 Configurazione dei file sip\_custom.conf e sip.conf

Il file di configurazione **sip\_custom.conf**, che nella distribuzione AsteriskNOW è incluso nel file **sip.conf**, indica ad Asterisk quali telefoni IP deve servire e come essi si devono presentare per potersi registrare.

E' conveniente editare il file **sip\_custom.conf** e non operare direttamente sul file **sip.conf** per maggiore chiarezza e per evitare sovrascritture nel caso in cui si voglia poi impiegare anche la GUI (*Graphical User Interface*) *FreePBX*.

Invece se non si intende utilizzare mai la GUI *FreePBX* o se si installa Asterisk dai sorgenti o da pacchetti di altre distribuzioni le configurazioni sotto indicate per sip\_custom.conf possono essere inserite direttamente nel file **sip.conf**.

#### 3.2.1 Configurazione del file sip.conf

Iniziamo verificando che il file **sip\_custom.conf** sia effettivamente stato dichiarato come incluso nel file **sip.conf** e che sia attivato il servizio *Message Waiting Indicator* (MWI) operando nel seguente modo (FIGURA 10):

- ci colleghiamo ed effettuiamo il login al server Asterisk
- ci spostiamo nella directory /etc/asterisk (**cd /etc/asterisk**)
- con i diritti di amministratore apriamo il file sip.conf con un editor di testo, per esempio *nano* (**sudo nano sip.conf**);

- verifichiamo che sia presente la dichiarazione **#include sip\_custom.conf**;
- attiviamo il servizio *Message Waiting Indicator* (MWI), che consente ai telefoni SIP di sapere se vi sono nuovi messaggi nella propria casella vocale (segreteria telefonica), inserendo nella sezione **[general]** del file **sip.conf** la riga seguente: **mwi\_from=asterisk**

```

root@localhost:/etc/asterisk
GNU nano 2.0.9 File: sip.conf Modified
;-----;
; Do NOT edit this file as it is auto-generated by FreePBX. All modifications to ;
; this file must be done via the web gui. There are alternative files to make ;
; custom modifications, details at: http://freepbx.org/configuration_files ;
;-----;
[general]
;
mwi_from=asterisk
; abilita l'indicazione di messaggi in attesa nella casella vocale (Message Waiting Indicator)
;
; These files will all be included in the [general] context
;
#include sip_general_additional.conf

;sip_general_custom.conf is the proper file location for placing any sip general
;options that you might need set. For example: enable and force the sip jitterbuffer.
;If these settings are desired they should be set the sip_general_custom.conf file.
;
; jbenable=yes
; jbforce=yes
;
;It is also the proper place to add the lines needed for sip nat'ing when going
;through a firewall. For nat'ing you'd need to add the following lines:
; nat=yes , externip= , localhost= , and optionally fromdomain= .
;
#include sip_general_custom.conf

;sip_nat.conf is here for legacy support reasons and for those that upgrade
;from previous versions. If you have this file with lines in it please make
;sure they are not duplicated in sip_general_custom.conf, if so remove them
;from sip_nat.conf as sip_general_custom.conf will have precedence.
#include sip_nat.conf

;sip_registrations_custom.conf is for any customizations you might need to do to
;the automatically generated registrations that FreePBX makes.
;
#include sip_registrations_custom.conf
#include sip_registrations.conf

; These files should all be expected to come after the [general] context
;
#include sip_custom.conf
#include sip_additional.conf

^G Get Help    ^O WriteOut    ^R Read File    ^Y Prev Page    ^K Cut Text    ^C Cur Pos
^X Exit        ^J Justify     ^W Where Is     ^V Next Page    ^U UnCut Text  ^T To Spell

```

FIGURA 10 File sip.conf con inclusione di sip\_custom.conf e abilitazione del servizio MWI.

### 3.2.2 Configurazione del file sip\_custom.conf

Il file **sip\_custom.conf** consente di personalizzare le impostazioni per i propri telefoni e softphone SIP. Esso è presente nella directory **/etc/asterisk**, e localmente va aperto ed editato (con i diritti di amministratore) con un editor di testo, come nano (**sudo nano sip\_custom.conf**).

Come prima cosa abilitiamo la videocomunicazione e impostiamo come lingua per i messaggi vocali l'italiano, operando nel seguente modo:

- inseriamo la riga **videosupport=yes** per abilitare la comunicazione anche in video;
- dal PC-ADMIN cerchiamo e scarichiamo da Internet le voci per asterisk in italiano, per esempio in formato audio **.gsm**, (attualmente per esempio dal sito <http://www.segreterietelefoniche.it/asterisk.php>), voci utilizzate per gli annunci vocali della segreteria telefonica (voicemail);

- creiamo una directory denominata **it** posta in **/var/lib/asterisk/sounds** con i seguenti comandi:
  - **cd /var/lib/asterisk/sounds**, si va nella directory sounds
  - **mkdir it**, si crea la cartella it
  - **cd it**, si entra nella cartella it
- trasferiamo il file compresso delle voci (via FTP) nella directory **it**, lo decomprimiamo qui con il comando **tar -xvf <nome\_file>**
- si visualizza il contenuto della directory (FIGURA 11) con il comando **ls**
- si verifica che i file copiati abbiano i permessi corretti con il comando **ls -l**
- se non è già presente, operando in modo analogo creiamo in **it** la directory **digits** (**mkdir digits**) dove copiamo e decomprimiamo i file audio per le cifre, le ore, i minuti, ecc.
- come esempio si veda il contenuto della directory **en** (english) posta in **/var/lib/asterisk/sounds** che contiene i file per gli annunci in lingua inglese (si noti che i file per la lingua italiana possono mantenere lo stesso nome di quelli in inglese presenti nella directory **en**)
- Dopo aver eseguito queste operazioni è possibile inserire nel file **sip\_custom.conf** la seguente riga: **language=it**

```

*****
; Configurazione dei telefoni SIP contenuta in sip_custom.conf
*****
;
;
videosupport=yes ; consenti le videochiamate
;
language=it ; cerca le voci in lingua italiana (se presenti nella directory /var/lib/asterisk/sounds/it)
;
;
+++++

```

```

root@localhost:/var/lib/asterisk/sounds/it# cd /var/lib/asterisk/sounds/
root@localhost/sounds# cd it
root@localhost/it# ls
0001429-itounds.tar.gz  conf-userwilljoin.gsm      queue-thereare.gsm        vm-from-phonenum.gsm    vm-received.gsm
agent-alreadyon.gsm    conf-waitforleader.gsm     queue-youarenext.gsm      vm-goodbye.gsm          vm-rec-name.gsm
agent-incorrect.gsm    demo-abouttotry.gsm       screen-callee-options.gsm vm-helpexit.gsm         vm-rec-temp.gsm
agent-loggedoff.gsm    demo-congrats.gsm         seconds.gsm               vm-INBOX.gsm            vm-rec-unv.gsm
agent-loginok.gsm      demo-echodone.gsm         spy-agent.gsm             vm-incorrect.gsm        vm-reenterpassword.gsm
agent-newlocation.gsm  demo-echotest.gsm         spy-h323.gsm              vm-incorrect-mailbox.gsm vm-repeat.gsm
agent-pass.gsm         demo-enterkeywords.gsm     spy-iax2.gsm              vm-instructions.gsm    vm-review.gsm
agent-user.gsm         demo-instruct.gsm         spy-iax.gsm               vm-intro.gsm            vm-saved.gsm
auth-incorrect.gsm     demo-moreinfo.gsm         spy-mgcp.gsm              vm-isonphone.gsm       vm-savedto.gsm
auth-thankyou.gsm      demo-nogo.gsm             spy-sip.gsm               vm-isunavail.gsm       vm-savefolder.gsm
beeperr.ulaw          demo-nomatch.gsm          spy-skinny.gsm            vm-last.gsm             vm-savemessage.gsm
beeperr.ulaw          demo-thanks.gsm           spy-zap.gsm               vm-leavemsg.gsm        vm-saveoper.gsm
beep.gsm              digits                     ss-noservice.gsm         vm-login.gsm            vm-sorry.gsm
beep.ulaw             dir-instr.gsm             transfer.gsm              vm-mailboxfull.gsm     vm-star-cancel.gsm
conf-adminmenu.gsm     dir-intro-fn.gsm          tt-allbusy.gsm           vm-message.gsm         vm-starmain.gsm
conf-enteringno.gsm    dir-intro.gsm             tt-monkeys.gsm           vm-messages.gsm        vm-tempgreeting2.gsm
conf-errormenu.gsm     dir-nomatch.gsm          tt-monkeysintro.gsm      vm-minutes.gsm         vm-tempgreeting.gsm
conf-getchannel.gsm    dir-nomore.gsm           tt-somethingwrong.gsm    vm-mismatch.gsm        vm-tempremoved.gsm
conf-getconfno.gsm     hello-world.gsm          tt-weasels.gsm           vm-msginstruct.gsm     vm-then-pound.gsm
conf-getpin.gsm        hours.gsm                 vm-advopts.gsm           vm-msgsaved.gsm        vm-theperson.gsm
conf-hasjoin.gsm       invalid.gsm               vm-and.gsm               vm-newpassword.gsm     vm-tocallback.gsm
conf-hasleft.gsm       minutes.gsm               vm-caldiffnum.gsm        vm-newuser.gsm         vm-tocallnum.gsm
conf-invalid.gsm       pbx-invalid.gsm           vm-changeto.gsm          vm-next.gsm             vm-tocancel.gsm
conf-invalidpin.gsm    pbx-invalidpark.gsm      vm-Cust1.gsm             vm-nobodyavail.gsm     vm-tocancelmsg.gsm
conf-kicked.gsm        pbx-transfer.gsm         vm-Cust2.gsm             vm-nobox.gsm           vm-toexternumber.gsm
conf-leaderhasleft.gsm privacy-incorrect.gsm     vm-Cust3.gsm             vm-no.gsm              vm-toforward.gsm
conf-locked.gsm        privacy-prompt.gsm       vm-Cust4.gsm             vm-nomore.gsm          vm-tohearenv.gsm
conf-lockednow_2.gsm   privacy-thankyou.gsm     vm-Cust5.gsm             vm-nonumber.gsm        vm-tomakecall.gsm
conf-lockednow.gsm     privacy-unident.gsm      vm-deleted.gsm           vm-num-i-have.gsm      vm-tooshort.gsm
conf-muted.gsm         priv-callee-options.gsm vm-delete.gsm            vm-nuovi.gsm           vm-toreply.gsm
conf-noempty.gsm       priv-callpending.gsm     vm-dialout.gsm           vm-nuovo.gsm           vm-torecord.gsm
conf-onlyone.gsm       priv-introsaved.gsm      vm-enter-num-to-call.gsm vm-old.gsm             vm-undeleted.gsm
conf-onlyperson.gsm    priv-recording.gsm       vm-extension.gsm         vm-onefor.gsm          vm-undelete.gsm
conf-otherinparty.gsm  queue-callswaiting.gsm   vm-Family.gsm            vm-options.gsm         vm-unknown-caller.gsm
conf-placeintoconf.gsm queue-holdtime.gsm       vm-first.gsm             vm-opts.gsm            vm-vecchi.gsm
conf-thereare.gsm      queue-less-than.gsm      vm-for.gsm               vm-passchanged.gsm     vm-vecchio.gsm

```

FIGURA 11 file per le voci in lingua italiana copiati e scompattati nella directory it, contenente anche la directory digits

Si passa ora alla configurazione dei parametri con cui si devono presentare i telefoni/softphone SIP e delle opzioni relative.

Per facilitare la configurazione è possibile inserire nel file un *template* (modello precompilato), contenente le configurazioni comuni a tutti i telefoni/softphone SIP, ed eventualmente aggiungere per ciascun telefono SIP solo delle ulteriori configurazioni particolari.

Il *template* può essere inserito nel file **sip\_custom.conf** operando nel seguente modo:

- assegniamo un nome al template, per esempio **base**, che va posto fra parentesi quadre ([base]) ed è seguito dal punto esclamativo tra parentesi tonda (!).  
Esempio di nome di template: **[base](!)**
- seguono le configurazioni comuni (si faccia riferimento al file di configurazione riportato nell'ESEMPIO 1)

N.B è conveniente inserire dei commenti alle varie istruzioni in modo da chiarire ciò che si sta facendo. Ogni riga di commento deve essere preceduta da un punto e virgola:

**; commento da inserire**

Fatto ciò si assegna a ciascun telefono/softphone SIP un nome, ponendolo fra parentesi quadra e facendolo seguire dal nome del template fra parentesi tonda.

Nel caso si voglia assegnare a un telefono un nome dato dal numero con cui può essere chiamato, per esempio 100, si inserirà: **[100](base)**

Nelle righe sottostanti il nome (numero) si inseriscono le eventuali configurazioni aggiuntive (si veda l'esempio sotto riportato).

Va sottolineato che con Asterisk il nome che identifica un telefono/softphone SIP non deve essere necessariamente il numero dell'interno associato al telefono. Tra l'altro, invece del numero è possibile utilizzare:

- un nome che identifichi il tipo di telefono SIP (xlite, 3cx, zoiper, ecc.);
- l'indirizzo MAC del telefono, che fisicamente identifica l'hardware su cui è installato;
- un altro identificativo ritenuto significativo (per esempio il nome dell'utente che lo utilizza).

Nel file di configurazione proposto nell'ESEMPIO 1 utilizzeremo sia numeri di telefono interni (per esempio **100**) sia nomi di softphone/telefoni SIP (per esempio **xlite-1**) sia altri identificativi (per esempio **segretaria** per ricordare chi lo usa).

---

## ESEMPIO 1

Esempio di configurazione di base da inserire nel file sip\_custom.conf.

```
.*****
; Configurazione dei telefoni/softphone SIP contenuta in sip_custom.conf
;*****
videosupport=yes ; consenti le videochiamate
language=it ; usa i messaggi vocali in italiano posti nella directory /var/lib/asterisk/sounds/it
;+++++
; template per la configurazione di base comune a tutti i telefoni IP
[base](!)
      ; (!) indica che e' un template in cui si mette la configurazione
      ; comune a tutti i telefoni IP
type = friend
```

```

; friend indica che Asterisk accetta connessioni sia in entrata sia in uscita dal telefono,
; che si puo' presentare con uno username o con un indirizzo IP
context = lab-tele
; [lab-tele] e' la posizione nel file extensions_custom.conf (il Dialplan)
; della numerazione per gli interni e delle procedure con cui sono trattate le chiamate
host = dynamic
; indica che il telefono IP si deve registrare presso Asterisk
secret = av14RT67uO0
; password (sicura) di autenticazione dei telefoni
; VA CONFIGURATA ANCHE NEI TELEFONI IP; NON USARE QUESTA
disallow = all
; inibisci l'uso di tutti i codec e consenti solo quelli indicati di seguito
allow = alaw
; consenti l'uso del codec audio PCM a standard Europeo (legge A di compressione)
allow = gsm ; consenti l'uso del codec audio gsm
allow = h263 ; consenti l'uso del codec video H.263
allow = h263p ; consenti l'uso del codec video H.263+
;+++++
; Di seguito si forniscono i nomi dei vari telefoni IP ed eventuali altre caratteristiche +
;+++++
[100](base) ; (base) = usa le configurazioni contenute nel template [base]
; denomina 100 il telefono, cosi' da ricordare il suo numero nel file
; extensions_custom.conf in alternativa si puo' usare un qualsiasi altro nome
; VA CONFIGURATO ANCHE NEL TELEFONO SIP
username = Docente
; indica lo username dell'utente (opzionale)
callerid = "Docente" <100>
; per maggiore chiarezza indica l'associazione
; fra username e numero di telefono dell'utente (opzionale)
; secret = 12qQ467PO ; decommentando questa riga e' possibile usare una password diversa
; da quella del template per autenticare il telefono. NON USARE QUESTA PASSWORD
mailbox=100 ; casella vocale dell'utente
;+++++ NOTA BENE ++++++
; il numero della mailbox serve al telefono SIP per verificare +
; se vi sono messaggi in attesa nella casella vocale associata al numero +
; e configurata in voicemail.conf, svolgendo cosi' la funzione detta +
; MWI - Message Waiting Indicator che deve essere attivata inserendo +
; nella parte [general] di sip.conf la riga mwi_from=asterisk +
;+++++
[101](base); denomina 101 il telefono, come il suo numero
username = studente
callerid = "studente" <101>
mailbox=101
;+++++
[102](base); denomina 102 il telefono, come il suo numero
username = itp
callerid = "itp" <102>
mailbox=102
;+++++
[xlite-1](base) ; denomina xlite-1 il telefono, per indicarne il tipo
username = lab-tele

```

```

callerid = "lab-tele" <xlite-1>
mailbox=108
;+++++
[xlite-2](base)
username = Vicario
callerid = "Vicario" <xlite-2>
mailbox=109
;+++++
[segretaria](base); denomina segretaria il telefono, come l'utente che lo utilizza
username = segretaria
callerid = "segretaria" <segretaria>
mailbox=200
;+++++
[xlite-3](base)
username = Dirigente
callerid = "Dirigente" <xlite-3>
mailbox=300
;+++++
[001](base); denomina 001 il telefono del centralino
username = Centralino
callerid = "Centralino" <001>
mailbox=001
.*****

```

### 3.3 Configurazione e personalizzazione del Dialplan tramite il file `extensions_custom.conf`

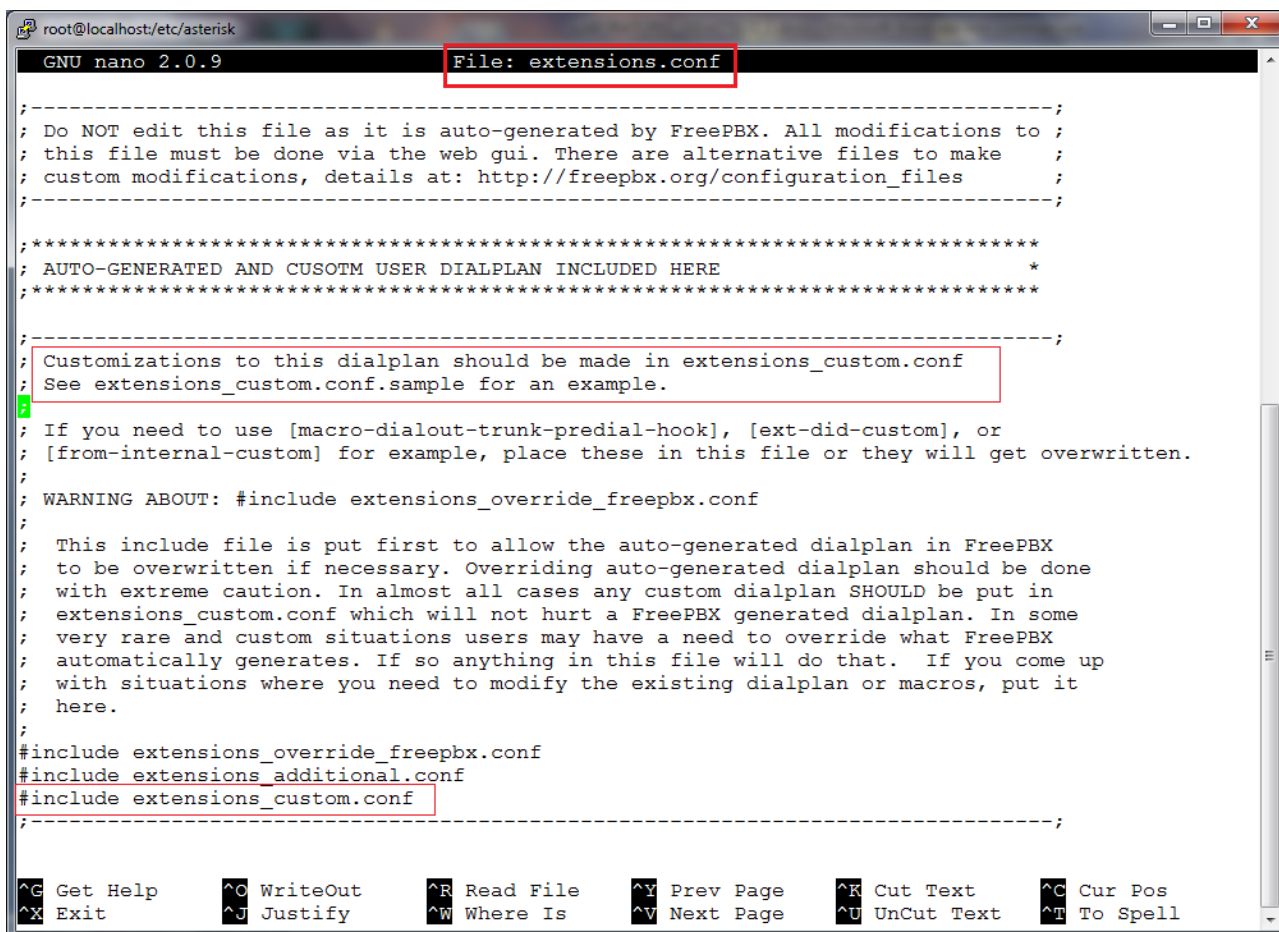
Il file **extensions.conf** contiene tutta la configurazione che consente di definire il *Dialplan*, cioè le modalità con le quali asterisk effettua il trattamento delle chiamate in ingresso e in uscita.

Per maggiore chiarezza, con AsteriskNOW è possibile personalizzare il *Dialplan* tramite il file **extensions\_custom.conf**, che deve essere dichiarato come incluso nel file **extensions.conf** (FIGURA 12).

Esempi di configurazioni che possono essere inserite nel file **extensions\_custom.conf** sono:

- assegnazione dei numeri di telefono o dei nomi con cui si chiamano gli interni,
- definizione del trattamento delle chiamate su ciascun numero/nome invocando delle opportune *Application* di asterisk;
- definizione dei numeri di telefono da digitare per ascoltare i messaggi presenti nelle caselle vocali;
- definizione di servizi accessori (come registrazione di messaggi vocali da utilizzare in seguito), ecc.

E' importante ricordare che con Asterisk la definizione del trattamento delle chiamate e dei servizi è fatta via software (tramite appositi comandi, applicazioni, macro, ecc.), per cui si possono facilmente personalizzare i servizi, così come se ne possono sviluppare di nuovi.



```
root@localhost:/etc/asterisk
GNU nano 2.0.9 File: extensions.conf

;-----;
; Do NOT edit this file as it is auto-generated by FreePBX. All modifications to ;
; this file must be done via the web gui. There are alternative files to make ;
; custom modifications, details at: http://freepbx.org/configuration_files ;
;-----;

;*****
; AUTO-GENERATED AND CUSOTM USER DIALPLAN INCLUDED HERE
;*****

;-----;
; Customizations to this dialplan should be made in extensions_custom.conf
; See extensions_custom.conf.sample for an example.
;
; If you need to use [macro-dialout-trunk-predial-hook], [ext-did-custom], or
; [from-internal-custom] for example, place these in this file or they will get overwritten.
;
; WARNING ABOUT: #include extensions_override_freepbx.conf
;
; This include file is put first to allow the auto-generated dialplan in FreePBX
; to be overwritten if necessary. Overriding auto-generated dialplan should be done
; with extreme caution. In almost all cases any custom dialplan SHOULD be put in
; extensions_custom.conf which will not hurt a FreePBX generated dialplan. In some
; very rare and custom situations users may have a need to override what FreePBX
; automatically generates. If so anything in this file will do that. If you come up
; with situations where you need to modify the existing dialplan or macros, put it
; here.
;
#include extensions_override_freepbx.conf
#include extensions_additional.conf
#include extensions_custom.conf
;-----;

^G Get Help    ^O WriteOut    ^R Read File   ^Y Prev Page   ^K Cut Text     ^C Cur Pos
^X Exit        ^J Justify     ^W Where Is    ^V Next Page   ^U UnCut Text   ^T To Spell
```

FIGURA 12 File extensions.conf con la dichiarazione di inclusione del file extensions\_custom.conf

Le caratteristiche principali da tenere presente per la creazione del *Dialplan* sono le seguenti

- I file di configurazione della numerazione e del trattamento delle chiamate, **extensions.conf** e **extensions\_custom.conf**, sono suddivisi in **context**.
- Un **context** viene identificato da un nome racchiuso fra parentesi quadra, [nome\_context], e raggruppa le istruzioni per la numerazione e il trattamento delle chiamate per i telefoni di un certo ambito; un context non interagisce con altri context (ambiti) se non attraverso esplicite istruzioni.
- Un **context** comprende un certo numero di **extensions**.
- Una **extension** (**exten**) definisce come si effettua il trattamento di una chiamata o di un servizio, tramite la definizione di un nome/ numero che la identifica e di un certo numero di istruzioni che sono:
  - numerate sequenzialmente in modo da definirne la priorità (**priority**)
  - eseguite sequenzialmente richiamando volta per volta l'applicazione contenuta nell'istruzione stessa (**Application** con i relativi parametri).
- Un context può includere altri context attraverso una dichiarazione di inclusione.
- Un context finisce solo quando inizia il context successivo.
- I commenti vanno inseriti facendoli precedere dal punto e virgola (;)

Un formato generale con cui può essere organizzato il file **extensions\_custom.conf** può quindi essere il seguente.

```

[nome_primo_context]
exten => nome_exten, priorità, Applicazione(parametri)
.....
exten => nome_exten, priorità, Applicazione(parametri)

[nome_secondo_context]
exten => nome_exten, priorità, Applicazione(parametri)
.....
exten => nome_exten, priorità, Applicazione(parametri)
include => [nome_altro_context] ; questo context include un altro context

[nome_terzo_context]
exten => nome_exten, priorità, Applicazione(parametri)
.....
exten => nome_exten, priorità, Applicazione(parametri)

[nome_altro_context]
exten => nome_exten, priorità, Applicazione(parametri)
.....
exten => nome_exten, priorità, Applicazione(parametri)

```

Per le istruzioni di una stessa **extension** *successive alla prima*, è possibile usare la notazione abbreviata:

**same => n, Applicazione(parametri)**

In questo modo si velocizza la stesura del dialplan in quanto è necessario indicare il numero/nome dell'extension e la priorità (1) solo sulla prima istruzione.

Un altro vantaggio è che nel caso in cui si aggiungano o tolgano delle istruzioni non è necessario rinumerarle perché ciò viene fatto automaticamente.

Nell'ESEMPIO 2 si riporta un estratto di un file `extensions_custom.conf` che realizza un semplice Dialplan con cui:

- si denomina `[lab-tele]` un context
- si effettua un trattamento di base delle chiamate
  - si assegna il numero di telefono (il 100) con cui è possibile chiamare un utente, denominando con tale numero una extension;
  - si invoca l'applicazione **Dial(..)** che ha come parametri
    - il protocollo da utilizzare per le chiamate (SIP)
    - il nome del telefono da chiamare (già configurato in `sip_custom.conf`),
    - per quanto tempo bisogna attendere che il chiamato risponda (in secondi), cioè per quanto tempo il telefono deve squillare;
  - se il chiamato non risponde si fa aprire ad Asterisk la chiamata (Answer);
  - si attende un secondo con l'applicazione `Wait(1)`,
  - si invoca l'applicazione `VoiceMail(numero_casella_vocale@context_voicemail)` che consente di registrare un messaggio nella casella vocale dell'utente (configurata in **voicemail.conf**); il context della voicemail può essere omissso se nel file `voicemail.conf` si definisce solo il context di nome `[default]`.
- si include un altro *context* denominato `[servizi]` in cui, fra l'altro, vi sono le extensions che permettono di chiamare le caselle vocali per ascoltare i messaggi registrati;
- per chiamare la casella vocale di un utente si definisce un'extension così strutturata:

- la prima istruzione denomina l'extension con asterisco seguito dal numero dell'utente (\*100);
- la seconda istruzione invoca l'applicazione **VoiceMailMain(numero\_casella\_vocale,s)** che ha come parametri
  - il numero della casella vocale (100), con eventualmente il nome del context se non è quello di default,
  - l'opzione **s** che in fase di test permette di *non* chiedere la password (configurata nel file voicemail.conf)

## ESEMPIO 2

Dialplan di base e relativo file extension\_custom.conf

```

*****
;
;                                DIALPLAN                                *
;                                PIANO DI NUMERAZIONE INTERNO            *
;                                E PROCEDURE PER IL TRATTAMENTO DELLE CHIAMATE *
;                                *****
; i commenti sono inseriti dopo il ;
[lab-tele] ; assegniamo il nome lab-tele al context creato per il lab. di telecomunicazioni
; segue l'estensione che definisce il trattamento delle chiamate verso un interno
; a cui è assegnato il numero 100
;+++++
; 100 è il numero assegnato a un utente, corrispondente al numero dell'extensions;      +
; le istruzioni da eseguire sono numerate per definirne la priorità (1, 2, 3, ..);      +
; ogni istruzione comprende una Applicazione, con gli eventuali parametri,            +
; che può eseguire una certa operazione                                                +
;+++++
;
exten => 100,1,NoOp() ; la prima istruzione, con priorità 1, può essere usata per definire il numero
;                                ; dell'extension che permette di raggiungere un telefono
;                                ; e non esegue nessuna applicazione
exten => 100,2,Dial(SIP/100,20)
; la seconda istruzione indica come si raggiunge il telefono di un utente quando si digita
; il numero dell'extension (il 100), eseguendo l'applicazione Dial(SIP/100,20) dove:
; Dial indica l'applicazione che permette di creare un canale in uscita verso un telefono,
; SIP indica il protocollo da usare;
; /100 è il nome del telefono SIP da chiamare, configurato in sip_custom.conf,
; nome che in questo caso coincide con il numero dell'extension;
; 20 è il tempo di attesa della risposta, espresso in secondi
exten => 100,3,Answer()
; se l'utente non risponde Asterisk apre la chiamata con l'applicazione Answer()
exten => 100,4,Wait(1) ; aspetta 1 secondo
exten => 100,5,VoiceMail(100)
; fai partire la registrazione di un messaggio nella casella vocale usando l'applicazione
; VoiceMail(numero) contenente il numero della casella vocale assegnata all'utente
; configurata nel file voicemail.conf all'interno del context [default]
exten => 100,6,hangup() ; riaggancia
;+++++
include => servizi ; includi il context di servizi vari
;+++++
[servizi]

```

```

; Numerazione per leggere le caselle di posta dei vari utenti
; costituita da: asterisco (*) seguito dal numero dell'interno
exten => *100,1,NoOp(lettura della casella vocale dell'utente avente interno 100)
; per chiamare la casella vocale dell'utente 100 si deve digitare *100
exten => *100,2,VoiceMailMain(100)
; VoiceMailMain() è l'applicazione per riprodurre i messaggi vocali registrati nella casella vocale
; identificata dal numero 100
; se si usa l'opzione s non viene richiesta la password : VoiceMail(100,s);
; il context della voicemail in voicemail.conf è [default]
exten => *100,3,hangup()

```

Come ulteriore esempio facciamo in modo che:

- l'utente a cui è assegnato il numero 100 possa anche essere chiamato digitando un nome, nell'esempio *docente*, definendo un'apposita extension, così denominata, che esegue le stesse operazione della 100;
- si utilizza la forma abbreviata **same => n,Application()** per le istruzioni successive alla prima, in modo da velocizzare la stesura del Dialplan.

### ESEMPIO 3

Assegnazione di un nome per chiamare un utente e impiego della forma abbreviata *same*.

```

[lab-tele]
exten => 100,1,NoOp()
exten => 100,2,Dial(SIP/100,20)
exten => 100,3,Answer()
exten => 100,4,Wait(1) ; aspetta 1 secondo
exten => 100,5,VoiceMail(100)
exten => 100,6,hangup() ; riaggancia
; -----
exten => docente,1, NoOp() ; l'utente può essere chiamato digitando su un softphone SIP
; il nome docente
same => n,Dial(SIP/100,20)
; docente viene raggiunto tramite il telefono SIP denominato 100 (in sip_custom.conf),
; si aspetta la risposta per 20 secondi
same => n,Answer() ; se l'utente non risponde asterisk apre la chiamata
same => n,VoiceMail(100) ; registra il messaggio vocale nella casella vocale n. 100
same => n,hangup() ; riaggancia
;
; seguono altre extensions
;
; ++++++
include => servizi ; includi il context di servizi vari
; ++++++
[servizi]
exten => *100,1,NoOp(lettura della casella vocale dell'utente avente interno 100)

```

```
; per chiamare la casella vocale dell'utente 100 (docente) si deve digitare *100
same => n, VoiceMailMain(100,s) ; con l'opzione s non viene richiesta la password
;
exten => *102,1,NoOp(lettura della casella vocale dell'utente avente interno 102)
; per chiamare la casella vocale dell'utente 102 (itp) si deve digitare *102
same => n, VoiceMailMain(102,s)
```

Definiamo poi (ESEMPIO 4) due nuove extension, per esempio denominate 102 e itp, con le quali si può raggiungere il telefono SIP denominato 102 dell'utente itp, avente casella vocale n. 102.

#### ESEMPIO 4

Aggiunta di nuove extensions

```
; queste extensions sono poste dopo l'extension docente ma prima di include => servizi
;+++++
; 102 -> Numero assegnato a ITP ; itp -> nome con cui chiamare l'utente ITP;
;+++++
exten => 102,1,Dial(SIP/102,20) ; 102 è il numero per chiamare l'utente, che viene raggiunto
; tramite il telefono SIP denominato 102
;per velocizzare la stesura del dialplan invochiamo l'applicazione Dial già nella prima istruzione
same => n,Answer() ; asterisk apre la chiamata
same => n,VoiceMail(102) ; registra il messaggio nella casella vocale 102
same => n,hangup() ; riaggancia
;+++++
; extension per la chiamata con nome +
;+++++
exten => itp,1,Dial(SIP/102,20); l'utente può anche essere chiamato componendo il nome itp
same => n,Answer() ; rispondi
same => n,VoiceMail(102) ; casella vocale
same => n,hangup() ; riaggancia
```

Prima di effettuare delle chiamate di prova configuriamo il file Voicemail.conf

### 3.4 Configurazione delle caselle vocali tramite il file Voicemail.conf

Il file **voicemail.conf**, contenuto nella directory `/etc/asterisk`, può essere editato per configurare i numeri/nomi delle caselle vocali assegnate ai vari utenti, in cui, con l'applicazione **VoiceMail(numero\_casella, opzioni)**, vengono registrati i messaggi vocali nel caso l'utente non risponda e da cui possono essere ascoltati i messaggi con l'applicazione **VoiceMailMain(numero\_casella, opzioni)**.

In AsteriskNOW il file `voicemail.conf` include due altri file (**vm\_general.inc** e **vm\_email.inc**), i quali consentono configurazioni avanzate che per semplicità qui non trattiamo.

Anche il file `voicemail.conf` è suddiviso in *context*.

Il nome del *context* che si configura va inserito come parametro nell'applicazione **VoiceMail()** a meno che non si scelga di configurare solo il context di nome [*default*], cosa che per semplicità scegliamo di fare.

Nel context sono inseriti i numeri delle caselle vocali con i relativi parametri secondo il seguente formato (ESEMPIO 5):

**numero => password, Cognome Nome, indirizzo mail (opzionale) |time zone (tz) (opzionale)**

### ESEMPIO 5

Esempio di file voicemail.conf per AsteriskNOW

```
*****
;
; Esempio di configurazione di base del file voicemail.conf
;
*****
[general]
;
;
externnotify=/var/lib/asterisk/agi-bin/vmnotify-newvm.php
#include vm_general.inc
#include vm_email.inc
envelope=yes
moveheard=yes
;
[default]
; usando il context [default] non e' necessario specificare il context nella funzione VoiceMail()
; in extensions_custom.conf che puo' riportare solo il numero es. Voicemail(103)
; Il formato e':
; numero (nome) della casella vocale => password, Cognome Nome, mail opz. |time zone (tz)
100 => 1659, docente, nome.cognome@provider.it |tz=Europe ; password e nome utente
102 => 4390, itp, nome.cognome@provider.it ; password e nome utente
.....
```

Per consentire l'ascolto di messaggi vocali registrati da parte di alcuni softphone, può essere utile cambiare il formato da **.wav** a **.gsm** editando il file `vm_general.inc` (FIGURA 13).

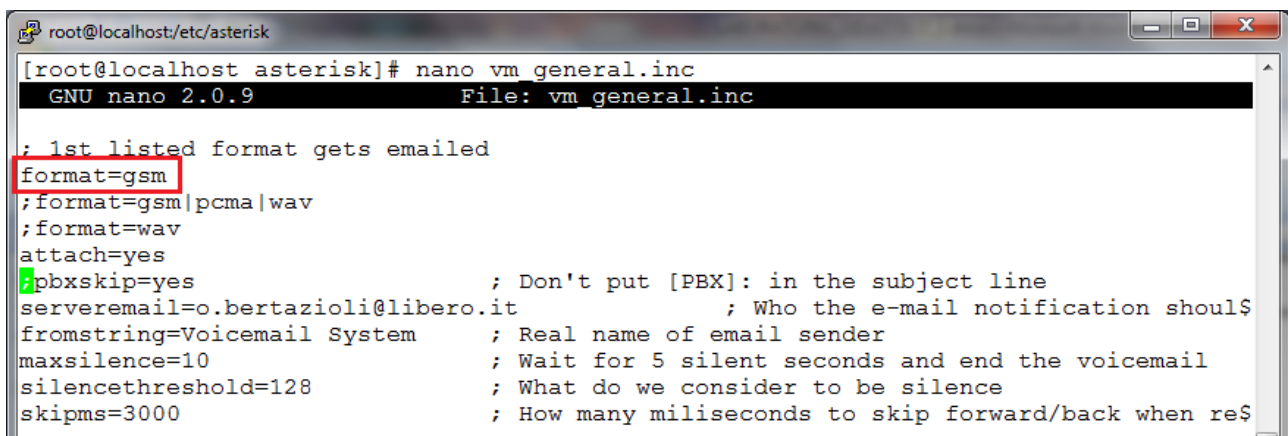


FIGURA 13 Definizione del formato dei file audio

### 3.4 La console di Asterisk

Asterisk ha una propria console a linea di comando (CLI, Command Line Interface) con la quale si può interagire direttamente con il servizio (daemon) asterisk.

E' possibile aprire (-r) la console di Asterisk in modalità *verbose* (v) con il comando `sudo asterisk -rv`

Tra l'altro la console può essere utilizzata per seguire i passi con cui Asterisk gestisce le chiamate, per verificare se vi sono istruzioni errate o non operative, per ottenere un aiuto in linea, digitando comandi quali (ESEMPIO 6):

- core show help
- core show applications
- core show application dial
- core show application voicemail
- core show application VoicemailMain

---

## ESEMPIO 6

### Aiuti in linea tramite la console di asterisk

```
root@localhost:~# asterisk -rv
Asterisk 11.12.1, Copyright (C) 1999 - 2013 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 11.12.1 currently running on localhost (pid = 2017)
```

```
localhost*CLI> core show help
! Execute a shell command
acl show Show a named ACL or list all named ACLs
ael reload Reload AEL configuration
ael set debug {read|tokens|mac} Enable AEL debugging flags
agi dump html Dumps a list of AGI commands in HTML format
agi exec Add AGI command to a channel in Async AGI
agi set debug {on|off} Enable/Disable AGI debugging
agi show commands [topic] List AGI commands or specific help
aoc set debug enable cli debugging of AOC messages
cc cancel Kill a CC transaction
cc report status Reports CC stats
```

```
root@localhost:~#
localhost*CLI> core show applications
-- Registered Asterisk Applications --
AddQueueMember: Dynamically adds queue members.
ADSIProg: Load Asterisk ADSI Scripts into phone
AELSub: Launch subroutine built with AEL
AGI: Executes an AGI compliant application.
Answer: Answer a channel if ringing.
Authenticate: Authenticate a user
BackGround: Play an audio file while waiting for digits of an extension to go to.
BackgroundDetect: Background a file with talk detect.
Bridge: Bridge two channels.
Busy: Indicate the Busy condition.
```

```

root@localhost:~
localhost*CLI> core show application dial

-- Info about application 'Dial' --

[Synopsis]
Attempt to connect to another device or endpoint and bridge the call.

[Description]
This application will place calls to one or more specified channels. As soon
as one of the requested channels answers, the originating channel will be
answered, if it has not already been answered. These two channels will then
be active in a bridged call. All other channels that were requested will then
be hung up.
Unless there is a timeout specified, the Dial application will wait
indefinitely until one of the called channels answers, the user hangs up,
or if all of the called channels are busy or unavailable. Dialplan executing
will continue if no requested channels can be called, or if the timeout
expires. This application will report normal termination if the originating
channel hangs up, or if the call is bridged and either of the parties in the
bridge ends the call.

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

[Arguments]
Technology/Resource
    Specification of the device(s) to dial. These must be in the format
    of 'Technology/Resource', where <Technology> represents a particular
    channel driver, and <Resource> represents a resource available to that
    particular channel driver.
Technology2/Resource2
    Optional extra devices to dial in parallel
    If you need more than one enter them as Technology2/Resource2&Te
    chnology3/Resource3&....
timeout
    Specifies the number of seconds we attempt to dial the specified
    devices
    If not specified, this defaults to 136 years.

```

```

root@localhost:~
localhost*CLI> core show application voicemail

-- Info about application 'VoiceMail' --

[Synopsis]
Leave a Voicemail message.

[Description]
This application allows the calling party to leave a message for the specified
list of mailboxes. When multiple mailboxes are specified, the greeting will
be taken from the first mailbox specified. Dialplan execution will stop if
the specified mailbox does not exist.
The Voicemail application will exit if any of the following DTMF digits are
received:
    0 - Jump to the 'o' extension in the current dialplan context.
    * - Jump to the 'a' extension in the current dialplan context.
This application will set the following channel variable upon completion:
${VMSTATUS}: This indicates the status of the execution of the VoiceMail
application.
    SUCCESS
    USEREXIT
    FAILED

[Syntax]
VoiceMail(mailbox[@context][&mailbox[@context][&...]][,options])

[Arguments]
options
    b: Play the 'busy' greeting to the calling party.

```

```
root@localhost:~  
localhost*CLI> core show application VoiceMailMain  
  
== Info about application 'VoiceMailMain' ==  
  
[Synopsis]  
Check Voicemail messages.  
  
[Description]  
This application allows the calling party to check voicemail messages. A  
specific <mailbox>, and optional corresponding <context>, may be specified.  
If a <mailbox> is not provided, the calling party will be prompted to enter  
one. If a <context> is not specified, the 'default' context will be used.  
The VoiceMailMain application will exit if the following DTMF digit is entered  
as Mailbox or Password, and the extension exists:  
* - Jump to the 'a' extension in the current dialplan context.  
  
[Syntax]  
VoiceMailMain([mailbox] [@context] [,options])  
  
[Arguments]  
options  
p: Consider the <mailbox> parameter as a prefix to the mailbox that  
is entered by the caller.  
  
g(#): Use the specified amount of gain when recording a voicemail  
message. The units are whole-number decibels (dB).  
  
s: Skip checking the passcode for the mailbox.
```

---

### 3.5 Backup dei file di configurazione

Se la configurazione dei file è stata fatta direttamente sul server Asterisk con un editor di testo come *nano*, effettuiamo il backup dei file di configurazione di AsteriskNOW sul PC-ADMIN con il server tftp (con indirizzo IP 10.0.0.59):

- dal PC-ADMIN ci colleghiamo al server asterisk (10.0.0.150) con un client SSH (PuTTY, ecc.);
- effettuato il login, digitiamo i seguenti comandi:

```
[root@asterisknow ~]# cd /etc/asterisk  
[root@asterisknow asterisk]# tftp 10.0.0.59  
tftp> put sip_custom.conf  
tftp> put extensions_custom.conf  
tftp> put voicemail.conf  
tftp> quit  
[root@asterisknow asterisk]#
```

Se invece i file sono stati editati sul PC-ADMIN, essi possono essere copiati nella directory */etc/asterisk* del server Asterisk via TFTP utilizzando il comando **get** <nome file> (si veda il paragrafo 3.1).

Terminata la configurazione e il backup dei file **sip\_custom.conf**, **extensions\_custom.conf** e **voicemail.conf**, si riavvia il servizio con il comando **sudo service asterisk restart**.

Come indicato nel LABORATORIO DIDATTICO 2, si possono quindi configurare dei telefoni/softphone SIP, con cui si effettuano delle chiamate di prova in audio ed eventualmente in audio e video, per esempio fra i telefoni/softphone 100 (docente) e 102 (itp), verificando che sia possibile registrare messaggi vocali nelle caselle ed ascoltarli chiamando la casella vocale.

### 3.5 Servizi Text To Speech per Asterisk

Un servizio *Text To Speech* (TTS) consente di convertire delle stringhe di testo in un segnale audio che può essere inviato a un utente nel corso di una chiamata.

In fase di sperimentazione e di test questo servizio consente di evitare la registrazione di messaggi di benvenuto, avvisi, ecc., che possono essere semplicemente scritti e fatti riprodurre come audio.

Esistono diverse applicazioni per i servizi TTS, anche opensource come per esempio **Festival**, che però vanno installati, opportunamente configurati e, se commerciali, acquistati.

Una soluzione molto semplice per sperimentare l'implementazione di un servizio TTS da usare con Asterisk è invece quella che fa uso di un servizio di Google denominato *googletts* (google text to speech) resa possibile da uno script in Perl scaricabile da Internet, attualmente agli indirizzi

<http://zaf.github.io/asterisk-googletts/>

<https://github.com/zaf/asterisk-googletts>

Per abilitare il servizio TTS tramite googletts è sufficiente verificare che siano installate, oppure installare, le seguenti dipendenze:

- perl
- perl-libwww-perl
- sox
- mpg123
- perl-CGI
- perl-LWP-UserAgent-Determined

Esse si possono installare con i comandi (FIGURA 14):

- sudo yum install perl perl-libwww-perl sox mpg123
- sudo yum install perl-CGI
- sudo yum install perl-LWP-UserAgent-Determined

E' inoltre necessario che il server Asterisk sia collegato a Internet con una connessione a larga banda.

```

root@localhost:~# yum install perl perl-libwww-perl sox mpg123
Loaded plugins: fastestmirror, kmod, security
Determining fastest mirrors
base
extras
pbx
pbx/primary
pbx
schmooze-commercial
schmooze-commercial/primary
schmooze-commercial
updates
Setting up Install Process
Package 4:perl-5.10.1-136.el6.i686 already installed and latest version
Package perl-libwww-perl-5.833-2.el6.noarch already installed and latest version
Package sox-14.2.0-6.el6.i686 already installed and latest version
Package matching mpg123-1.13.6-17.el6.i686 already installed. Checking for update.
Nothing to do
[root@localhost ~]#

```

FIGURA 14 Verifica che le dipendenze richieste per il servizio TTS siano installate e aggiornate

Fatto ciò è sufficiente:

- scaricare dal sito citato il file compresso attualmente denominato **asterisk-googletts-0.6.tar.gz**
- creare una directory, per esempio denominata **tts** (**mkdir tts**) in cui copiare (**cp asterisk-googletts-0.6.tar.gz /var/lib/asterisk/agi-bin/tts/**) entrare nella directory **tts** (**cd /var/lib/asterisk/agi-bin/tts**) e scompattare il file con il comando **tar -xvf asterisk-googletts-0.6.tar.gz**
- entrare nella directory **asterisk-googletts-0.6** e copiare nella directory **/var/lib/asterisk/agi-bin/** il file **googletts.agi**  
**cp googletts.agi /var/lib/asterisk/agi-bin/**

E' anche possibile personalizzare lo script aprendolo e modificando la lingua di default (da *en* in *it*) e la versione di sox (si veda l' ESEMPIO 7).

#### ESEMPIO 7 Esempio di personalizzazione dello script per il TTS con googletts

```

# ----- #
# User defined parameters: #
# ----- #
# Default language      #
my $lang = "it";
# Output audio sample rate  #
my $samplerate = 8000;
# Output speed factor      #
my $speed = 1.2;
# Use of cache mechanism    #
my $usecache = 1;
# Cache directory path      #
my $cachedir = "/var/lib/asterisk/temp";
# SoX Version              #
my $sox_ver = 14;
# Verbose debugging messages #
my $debug = 0;
# ----- #
my %AGI;

```

```

my @text;
my $ua;
my $res;
my $fh;
my $tmpname;
my $filename;
my $fexten;
my @result;
my @soxargs;
my $name;
my $intkey = "";
my $tmpdir = "/var/lib/asterisk/temp";
my $maxlen = 2048;
my $url = "http://translate.google.com/translate_tts";
my $sox = `/usr/bin/which sox`;
my $mpg123 = `/usr/bin/which mpg123`;

```

Lo script googletts.agi può essere utilizzato all'interno del file **extensions\_custom.conf** (si veda l'ESEMPIO 8) scrivendo in una extension un'espressione avente il seguente formato:

```

.....
same => n,AGI(googletts.agi,"Testo da riprodurre come audio ",lingua da usare)

```

La lingua italiana viene indicata come **it** (si veda l'esempio seguente)

```

same => n,AGI(googletts.agi,"Testo da riprodurre come audio ",it)

```

### 3.6 Esempi di impiego di alcune Application nel Dialplan extensions\_custom.conf

Analizziamo l'impiego delle seguenti Application utilizzabili nelle extensions di un Dialplan (file extensions\_custom.conf):

- **Playback(nomefile, ...)**

Consente di riprodurre un file audio presente nella directory /var/lib/asterisk/sounds o in una sottodirectory di cui si dà il percorso. Avvertenze: non va messa l'estensione del file e il formato deve essere tra quelli supportati da Asterisk

- **Background(nomefile, ..);** riproduce un file audio presente nella directory /var/lib/asterisk/sounds in background, per esempio mentre si attende che l'utente digiti un numero
- **Read(variabile,nomefileaudio,max\_num\_cifre, ..);** legge un certo numero di cifre digitate da un utente e le assegna alla variabile indicata, le cifre vengono lette dopo che viene riprodotto il file audio indicato, è possibile specificare il numero di cifre da leggere oppure mettere come num\_cifre 0 e in questo caso la lettura deve essere terminata dall'utente digitando il tasto # .
- **Record(nomefile.formato);** consente di registrare un messaggio audio in un file avente il nome e l'estensione (.wav, .gsm, .sln, ..) indicati.
- **SayDigits(digits/numeri);** riproduce i numeri indicati; se i numeri sono contenuti in una variabile, per esempio *nome-messaggio* il loro valore va passato utilizzando il seguente formato **\${nome-messaggio}**;
- **SayAlpha(stringa);** riproduce le lettere che compongono una stringa, la quale può anche essere contenuta in una variabile se si usa la sintassi **\${variabile}**
- **GotoIf(condizione?[<etichetta\_vero>]: [<etichetta\_falso>];** a seconda che la condizione sia vera o falsa va all'extension indicata in etichetta\_vero oppure in quella indicata in etichetta\_falso.

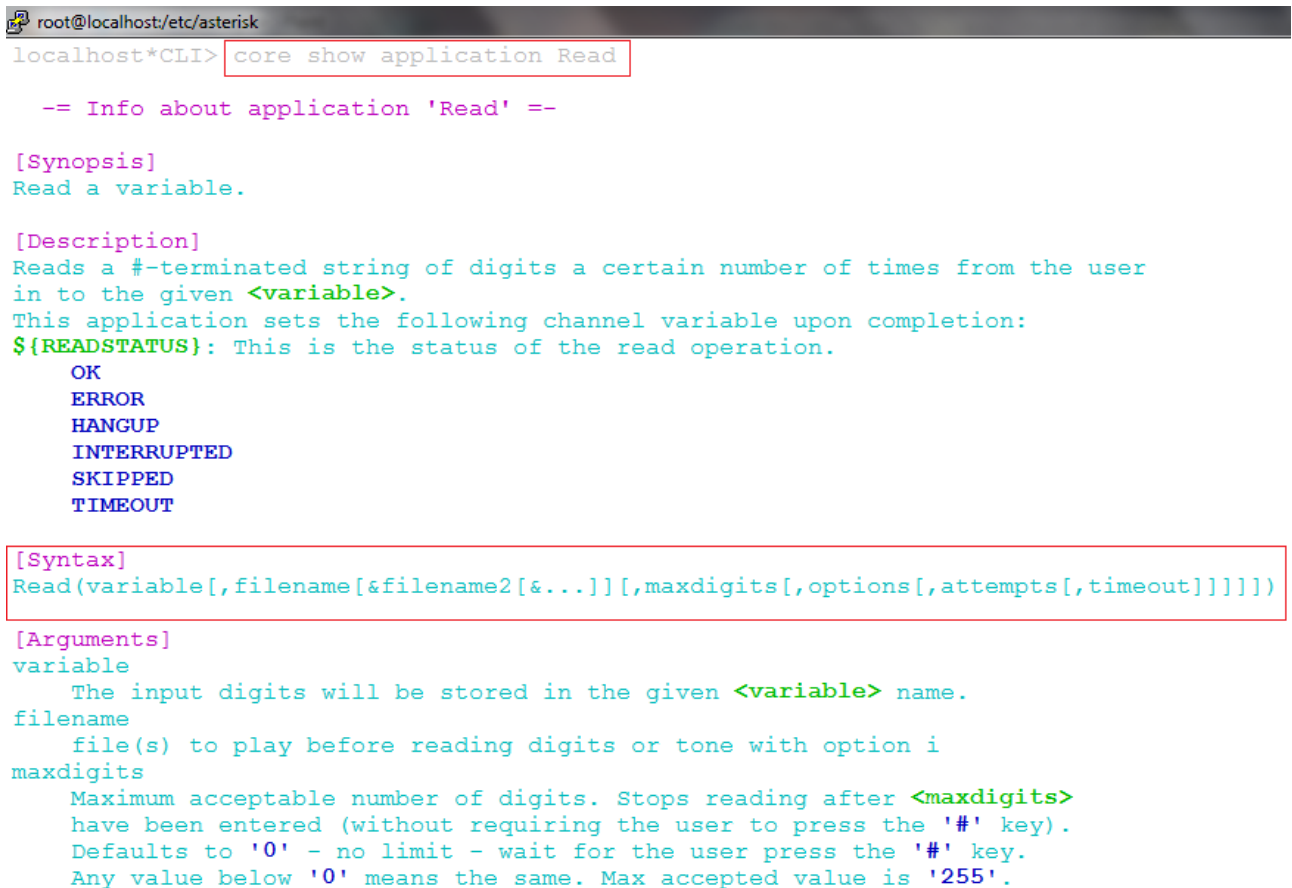
NB: Una volta definita una variabile e assegnatole un valore/stringa, con la sintassi `${variabile}` si dice ad Asterisk di utilizzare il valore o la stringa contenuta nella variabile.

E' possibile ottenere un help in linea aprendo la console (CLI) di Asterisk (sudo **asterisk -rv**) e digitando il comando `core show application <nome dell'applicazione>`, come per esempio (FIGURA 15):

```
localhost*CLI> core show application playback
```

```
localhost*CLI> core show application GotoIf
```

.....



```
root@localhost/etc/asterisk
localhost*CLI> core show application Read

-- Info about application 'Read' --

[Synopsis]
Read a variable.

[Description]
Reads a #-terminated string of digits a certain number of times from the user
in to the given <variable>.
This application sets the following channel variable upon completion:
${READSTATUS}: This is the status of the read operation.
    OK
    ERROR
    HANGUP
    INTERRUPTED
    SKIPPED
    TIMEOUT

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

[Arguments]
variable
    The input digits will be stored in the given <variable> name.
filename
    file(s) to play before reading digits or tone with option i
maxdigits
    Maximum acceptable number of digits. Stops reading after <maxdigits>
    have been entered (without requiring the user to press the '#' key).
    Defaults to '0' - no limit - wait for the user press the '#' key.
    Any value below '0' means the same. Max accepted value is '255'.
```

```

root@localhost:/etc/asterisk
=====
Connected to Asterisk 11.13.1 currently running on localhost (pid = 2713)
localhost*CLI> core show application playback

-- Info about application 'Playback' --

[Synopsis]
Play a file.

[Description]
Plays back given filenames (do not put extension of wav/alaw etc). The playback
command answer the channel if no options are specified. If the file is
non-existent it will fail
This application sets the following channel variable upon completion:
${PLAYBACKSTATUS}: The status of the playback attempt as a text string.
    SUCCESS
    FAILED
See Also: Background (application) -- for playing sound files that are
interruptible
WaitExten (application) -- wait for digits from caller, optionally play music
on hold

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

```

FIGURA 15 Informazioni sulle applicazioni Read e Playback fornite dalla CLI di Asterisk

### 3.7 Esempi di servizi e di trattamento delle chiamate in extensions\_custom.conf

Negli esempi seguenti si esemplifica l'uso di alcune Application per effettuare registrazioni ed ascolti di propri messaggi vocali, trasferimenti di chiamata, differenziazione dei messaggi di cortesia per le caselle vocali, impiego delle subroutine

#### ESEMPIO 8

**Extension per implementare un servizio che consente di registrare e riascoltare dei messaggi vocali.**

```

[servizi]
; ++++++
; Extension che consente di registrare un messaggio del context servizi, incluso in lab-tele
; ++++++
exten => 400,1,NoOp() ; numero da digitare per registrare i messaggi
same => n,Wait(1)
same => n,AGI(googletts.agi,"Registrazione di un messaggio.",it) ; text to speech riprodotto
same => n,Wait(1)
same => n,AGI(googletts.agi,"Dopo il beep digitate un numero che costituisce il nome del
messaggio.",it)
same => n,AGI(googletts.agi,"Terminate il numero con cancelletto.",it)
same => n,Read(nome-messaggio,beep,0)
same => n,AGI(googletts.agi,"Il numero scelto come nome è",it)
same => n,SayDigits(digits/${nome-messaggio})
same => n,Wait(1)
same => n,AGI(googletts.agi,"Il messaggio viene registrato dopo il segnale acustico.",it)
same => n,AGI(googletts.agi,"Terminate il messaggio con cancelletto.",it)
; NOTA BENE

```

```

; fare una directory messaggi-registrati in /var/lib/asterisk/sounds;
; i messaggi vengono registrati in quella directory
same => n,Record(messaggi-registrati/${nome-messaggio}.sln16)
same => n,Wait(1)
same => n,AGI(googletts.agi,"Riascolto del messaggio registrato numero ${nome-messaggio}.",it)
;
same => n,Playback(messaggi-registrati/${nome-messaggio})
same => n,Wait(1)
same => n,AGI(googletts.agi,"Arrivederci.",it)
same => n,Hangup()
;
;+++++
; Estensione per ascoltare un messaggio registrato in precedenza      +
;+++++
exten => 500,1,NoOp() ; numero da digitare per riascoltare i messaggi
same => n,AGI(googletts.agi,"Digita il numero del messaggio registrato che vuoi ascoltare dopo il
beep.",it)
same => n,AGI(googletts.agi,"Premi cancelletto per indicare la fine del numero.",it)
same => n,Read(nome-messaggio,beep,0)
same => n,Wait(1)
same => n,Playback(messaggi-registrati/${nome-messaggio})
same => n,AGI(googletts.agi,"Grazie per aver chiamato.",it)
same => n,Hangup()

```

### 3.7.1 Trasferimento di chiamata

Con Asterisk è semplice realizzare un servizio di trasferimento di chiamata ad altro numero in caso di mancata risposta di un utente.

Nell'ESEMPIO 9 si riportano le extensions aventi numero **300** e nome **dirigente** inserite nel file `extensions_custom.conf`, con le quali nel caso l'utente 300/dirigente non risponda dopo 20 secondi vengono intraprese le seguenti azioni:

- si fa aprire la chiamata da Asterisk con l'applicazione `Answer()`;
- si fa partire un annuncio vocale che avvisa del trasferimento della chiamata ad un altro utente (tramite il servizio TTS di googletts);
- si aspetta un secondo con l'applicazione `Wait(1)`;
- si effettua la chiamata a un altro telefono SIP (denominato `xlite-2` nell'esempio) e si aspetta per 20 secondi;
- nel caso neanche questo utente risponda si fa partire la segreteria telefonica, `VoiceMail`, per la registrazione nella casella vocale dei messaggi.

N.B. I telefoni SIP utilizzati devono essere già configurati nel file `sip_custom.conf` (si veda l'ESEMPIO 1) e devono altresì essere state configurate le caselle vocali nel file `voicemail.conf` (ESEMPIO 5).

#### Esempio 9

Trasferimento di chiamata a seguito della mancata risposta dell'utente e di utilizzo di `googletts.agi` per riprodurre in lingua italiana il testo posto fra "" "".

```

root@localhost/etc/asterisk
GNU nano 2.0.9 File: extensions_custom.conf Modified
; ++++++;
; Dirigente
; ++++++
exten => 300,1,Dial(SIP/xlite-3,20)
; il telefono IP xlite-3 puo' essere chiamato usando come interno il 300
; attendi 20 s la risposta
;
same => n,Answer() ; rispondi
same => n,Wait(1) ; aspetta 1 secondo
same => n,AGI(googletts.agi,"La chiamata verrà trasferita al Collaboratore Vicario",it)
same => n,Wait(1) ; aspetta un secondo
same => n,Dial(SIP/xlite-2,20) ; passa la chiamata al telefono del Vicario
; se non risponde fai partire la segreteria telefonica
same => n,Wait(1)
same => n,AGI(googletts.agi,"L'interno chiamato non risponde lasciare un messaggio nella casella vocale",it)
same => n,VoiceMail(300)
same => n,Hangup()
;
; ++++++
exten => dirigente,1,Dial(SIP/xlite-3,20)
; il telefono IP xlite-3 puo' essere chiamato usando come il nome dirigente
; invece di un numero, attendi 20 s la risposta
same => n,Answer() ; rispondi
same => n,Wait(1) ; aspetta 1 secondo
same => n,AGI(googletts.agi,"La chiamata verrà trasferita al Collaboratore Vicario",it)
same => n,Wait(1) ; aspetta un secondo
same => n,Dial(SIP/xlite-2,20) ; passa la chiamata al telefono del Vicario
; se non risponde fai partire la segreteria telefonica
same => n,Wait(1)
same => n,AGI(googletts.agi,"L'interno chiamato non risponde lasciare un messaggio nella casella vocale",it)
same => n,VoiceMail(300)
same => n,Hangup()
;
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^N Next Page ^U UnCut Text ^T To Spell

```

### 3.7.2 Esempio di utilizzo dell'Application GotoIf(..)

Se si desidera implementare una soluzione più completa che permette di differenziare i messaggi in relazione allo stato di non disponibile (assente) o occupato di un interno è possibile scrivere l'extension dell'ESEMPIO 10, che fa uso dell'applicazione GotoIf(...).

#### ESEMPIO 10

Differenziazione dei messaggi di occupato e assente

```

; ++++++
; Dirigente +
; ++++++
; -----
; VERSIONE CON DIFFERENZIAZIONE DEI MESSAGGI DI OCCUPATO E ASSENTE
; -----
exten => 300,1,Dial(SIP/xlite-3,20)
; il telefono IP xlite-3 del Dirigente puo' essere chiamato usando come interno il 300
; attendi 20 s la risposta
; same => n, sayalpha("${DIALSTATUS}"); questa funzione permette di ascoltare lo stato della
; chiamata in inglese; decommentare per provarla
same => n,GotoIf("${DIALSTATUS}" = "BUSY"?occupato:assente)
; se lo stato è busy (occupato) vai alla riga n(occupato)
; se lo stato è unavailable (non disponibile) vai a n(assente)
; se l'utente e' occupato in altra conversazione chiedi se si vuole trasferire la chiamata
; oppure se si vuole lasciare un messaggio nella casella vocale

```

```

same => n(occupato),AGI(googletts.agi,"Buongiorno, il Dirigente scolastico è impegnato in
un'altra conversazione.",it)
same => n,AGI(googletts.agi,"Premere uno per trasferire la chiamata al Collaboratore vicario.",it)
same => n,AGI(googletts.agi,"Premere due per lasciare un messaggio nella casella vocale",it)
same => n,AGI(googletts.agi,"Grazie per aver chiamato.",it)
; Text To Speech che riproduce il testo tra " " sfruttando il servizio googletts
same => n,Wait(1)
same => n,Read(numero,beep,1,0) ; funzione che assegna alla variabile numero il valore del tasto
; premuto dopo il beep
; same => n,Saydigits(${numero}) ; decommentare per riascoltare il numero letto
same => n,GotoIf(${numero} = 1)?trasferisci:registra)
; se numero=1 allora vai a n(trasferisci) altrimenti vai a n(registra)
; E' stato premuto il tasto 1 -> la chiamata viene trasferita
same => n(trasferisci),AGI(googletts.agi,"La chiamata verrà trasferita al Collaboratore vicario",it)
same => n,Wait(1)
same => n,Dial(SIP/xlite-2,20) ; passa la chiamata al vicario e aspetta 20 s
; se anche questo interno non risponde fai partire la segreteria telefonica
same => n,AGI(googletts.agi,"L'interno chiamato non risponde. Lasciate un messaggio nella casella
vocale del Dirigente",it)
same => n,Wait(1)
same => n,VoiceMail(300,b) ;
; E' stato premuto il tasto 2
same => n(registra),VoiceMail(300,b) ; con l'opzione b (busy) si fa ascoltare un messaggio di
; utente occupato -> vm-isonphone.gsm
; oppure si può mandare in esecuzione il messaggio busy.sln
; se è presente in /var/spool/asterisk/voicemail/default/300
; il messaggio busy.slin può essere registrato componendo il numero della
; casella vocale (*300), quindi digitando 0 e a seguire 2
same => n,hangup() ; riaggancia
; l'utente chiamato non è disponibile (e' assente)
same => n(assente),NoOp()
same => n,Dial(SIP/310,20) ; chiama sullo smartphone che ha nome 310
same => n,Wait(1)
same => n(registra),VoiceMail(300,u)
; con l'opzione u (unavailable) si fa ascoltare un messaggio di
; utente non disponibile -> unavailable.gsm
; oppure si può mandare in esecuzione il messaggio unavail.sln
; se è presente in /var/spool/asterisk/voicemail/default/300
; il messaggio può essere registrato componendo il numero della casella vocale (*300)
; quindi digitando 0 e a seguire 2
same => n,hangup() ; riaggancia
; -----
; estensione per la chiamata con nome
; -----
;
;
exten => dirigente,1,Dial(SIP/xlite-3,20)
; il telefono IP xlite-3 puo' essere chiamato componendo il nome dirigente invece del numero 300
same => n,GotoIf("${DIALSTATUS}" = "BUSY"?occupato:assente)
same => n(occupato),AGI(googletts.agi,"Buongiorno, il Dirigente scolastico è impegnato in
un'altra conversazione.",it)
same => n,AGI(googletts.agi,"Premere uno per trasferire la chiamata al Collaboratore vicario.",it)

```

```

same => n,AGI(googletts.agi,"Premere due per lasciare un messaggio nella casella vocale",it)
same => n,AGI(googletts.agi,"Grazie per aver chiamato.",it)
same => n,Wait(1)
same => n,Read(numero,beep,1,0)
same => n,GotoIf(${numero} = 1)?trasferisci:registra)
same => n(trasferisci),AGI(googletts.agi,"La chiamata verrà trasferita al Collaboratore vicario",it)
same => n,Wait(1)
same => n,Dial(SIP/xlite-2,20) ; passa la chiamata al vicario e aspetta 20 s
same => n,AGI(googletts.agi,"L'interno chiamato non risponde. Lasciate un messaggio nella casella vocale del Dirigente",it)
same => n,Wait(1)
same => n,VoiceMail(300,b)
same => n(registra),VoiceMail(300,b) ;
same => n,hangup() ; riaggancia
same => n(assente),NoOp()
same => n,Dial(SIP/310,20) ; chiama sullo smartphone identificato dal numero 310
same => n,VoiceMail(300,u)
; con l'opzione u si manda in esecuzione un messaggio di utente non disponibile
same => n,hangup() ; riaggancia
;+++++
; Smartphone del Dirigente +
;+++++
exten => 310,1,Dial(SIP/310,20); assegna il 310 allo smartphone identificato sempre dal 310
; se l'utente chiamato non risponde chiama sul fisso identificato dal nome xlite-3
same => n,Dial(SIP/xlite-3,20) ; chiama sul fisso
; se non risponde fai partire la casella vocale
same => n,VoiceMail(300,u)
same => n,hangup() ; riaggancia
;+++++

```

### 3.7.3 Esempio di utilizzo delle subroutine nel file extensions\_custom.conf

Una soluzione più elegante è quella che definisce una **subroutine** per la gestione delle caselle vocali, differenziando i messaggi fra libero e occupato, e che fa uso dell'applicazione **Gosub(parametri)** come mostrato nell'ESEMPIO 11.

---

#### ESEMPIO 11

Uso delle subroutine per semplificare la stesura del Dialplan

```

;-----
; esempio di una delle extensions del context [lab-tele]
;-----
exten => 102,1,NoOp() ; 102 e' il numero di itp
same => n, set(TELEFONO=102)
; denomina TELEFONO la variabile da passare alla subroutine per chiamare il numero
same => n, set(CASVOC=102)
; denomina CASVOC la variabile da passare alla subroutine per la casella vocale
same => n,GoSub(subcasella,casella,1(${TELEFONO},${CASVOC}))
; la chiamata vera e propria viene fatta nella subroutine [subcasella]
; la casella vocale può anche essere passata tramite l'extension

```

```

;esempio: same => n,GoSub(subcasella,casella,1({TELEFONO},{EXTEN}))
;-----
; estensione per la chiamata con nome
;-----
exten => itp,1,NoOp(); chiamata con nome
; la chiamata vera e propria viene fatta nella subroutine [subcasella]
same => n, set(TELEFONO=102)
same => n, set(CASVOC=102)
same => n,GoSub(subcasella,casella,1({TELEFONO},{CASVOC}))
; .....
;-----
; seguono le altre extensions del context [lab-tele]
;-----
;.....
;
;+++++
; subroutine per gestire i messaggi nelle caselle vocali
; +++++
[subcasella]; subroutine per la gestione della chiamata e della casella vocale
; la subroutine accetta i parametri passati dall'application GoSub()
; con la denominazione ARG1, ARG2,
exten => casella,1,NoOp(); denomina casella questa extension
same => n,set(TELEFONO={ARG1})
; per maggiore chiarezza assegniamo alla variabile TELEFONO il contenuto di ARG1
same => n,set(CASVOC={ARG2})
; per maggiore chiarezza assegniamo alla variabile CASVOC il contenuto di ARG2
same => n,Dial(SIP/{TELEFONO},20) ; chiama l'interno desiderato
; se l'utente non risponde o rifiuta la chiamata chiedi se si vuole chiamare il centralino
; oppure lasciare un messaggio nella casella vocale
same => n,AGI(googletts.agi,"Buongiorno, l'interno chiamato non risponde.",it)
same => n,AGI(googletts.agi,"Dopo il beep, premere uno per trasferire la chiamata al
Centralino",it)
same => n,AGI(googletts.agi,"Premere due per lasciare un messaggio nella casella vocale",it)
same => n,AGI(googletts.agi,"Grazie per aver chiamato.",it)
same => n,Read(numero,beep,1,0)
; funzione che assegna alla variabile numero il valore del tasto premuto dopo il beep
same => n,GotoIf({${numero} = 1}?trasferisci:registra)
; se numero=1 allora vai a n(trasferisci) altrimenti vai a n(registra)
same => n(trasferisci),NoOp()
same => n,Dial(SIP/001,20) ; passa la chiamata al centralino (TEL. 001) e aspetta 20 s
; se anche questo interno non risponde fai partire la segreteria telefonica
same => n,AGI(googletts.agi,"L'interno chiamato non risponde. Lasciate un messaggio nella casella
vocale",it)
same => n,VoiceMail({CASVOC},u)
same => n,Hangup(); chiudi la chiamata
; E' stato premuto il tasto 2
same => n(registra),NoOp()
; si fanno ascoltare messaggi diversi a seconda che l'utente sia occupato o non disponibile
; ${DIALSTATUS} restituisce lo stato di occupato oppure no. E' occupato se rifiuta la chiamata

```

```
same => n,GotoIf("${DIALSTATUS}" = "BUSY"?occupato:assente)
; se l'utente rifiuta la chiamata vai a n(occupato) mentre se non risponde vai a n(assente)
same => n(occupato),VoiceMail(${CASVOC},b)
; con l'opzione b (busy) si fa ascoltare il messaggio di utente occupato
same => n,Hangup(); chiudi la chiamata
same => n(assente),VoiceMail(${CASVOC},u)
; con l'opzione u (unavailable) si fa ascoltare il messaggio di utente non disponibile
same => n,Hangup() ; chiudi la chiamata
;
```

---

## Bibliografia

Oltre a una vasta documentazione reperibile su Internet, un testo di sicuro interesse (disponibile anche in formato PDF) è il seguente:

Russell Bryant, Leif Madsen, and Jim Van Meggelen  
Asterisk™: The Definitive Guide, Fourth Edition  
O'Reilly