



Table des matières

1) Prise en main du logiciel avec article GLMF.....	3
2) Approfondissement sur sip.conf et extensions.conf.....	4
3) Étude des flux.....	4
4) Mettre en place la fonction "echo" au numéro 3666.....	8
5) Mettre en place une horloge parlante au numéro 3699.....	9
6) Mettre en place la lecture de la météo au numéro 3030.....	9
7) Jeu "trouver un nombre entre 1 et 99" au numéro 3535.....	11
8) Écouter NRJ au numéro 3636.....	12
9) Hardphone et transfert d'appel.....	12

On commence par se connecter dans la VM que le prof nous a donné :

```
sudo ssh test@10.255.150.114
```

Ensuite on modifie le mdp de la VM :

```
passwd test
```

Maintenant on se passe en root et on modifie le mdp :

```
passwd root
```

Maintenant on crée 2 utilisateurs :

```
root@holsteing-laforge : /home/test
# adduser aholsteing
Ajout de l'utilisateur « aholsteing » ...
Ajout du nouveau groupe « aholsteing » (1003) ...
Ajout du nouvel utilisateur « aholsteing » (1003) avec le groupe
« aholsteing » ...
Création du répertoire personnel « /home/aholsteing »...
Copie des fichiers depuis « /etc/skel »...
Entrez le nouveau mot de passe UNIX :
```

```
Retapez le nouveau mot de passe UNIX :  
passwd: password updated successfully  
Changing the user information for aholsteing  
Enter the new value, or press ENTER for the default  
    Full Name []:  
    Room Number []:  
    Work Phone []:  
    Home Phone []:  
    Other []:  
Cette information est-elle correcte ? [O/n]o
```

```
root@holsteing-laforge : /home/test  
# adduser slaforge  
Ajout de l'utilisateur « slaforge » ...  
Ajout du nouveau groupe « slaforge » (1004) ...  
Ajout du nouvel utilisateur « slaforge » (1004) avec le groupe « slaforge » ...  
Création du répertoire personnel « /home/slaforge »...  
Copie des fichiers depuis « /etc/skel »...  
Entrez le nouveau mot de passe UNIX :  
Retapez le nouveau mot de passe UNIX :  
passwd: password updated successfully  
Changing the user information for slaforge  
Enter the new value, or press ENTER for the default  
    Full Name []:  
    Room Number []:  
    Work Phone []:  
    Home Phone []:  
    Other []:  
Cette information est-elle correcte ? [O/n]o
```

Maintenant on ajoute les utilisateurs au groupe root :

```
root@holsteing-laforge : /home/test  
# adduser aholsteing root  
Ajout de l'utilisateur « aholsteing » au groupe « root »...  
Adding user aholsteing to group root  
Fait.  
  
root@holsteing-laforge : /home/test  
# adduser slaforge root  
Ajout de l'utilisateur « slaforge » au groupe « root »...  
Adding user slaforge to group root  
Fait.
```

On ferme notre session test et on se logue sur notre nouvel utilisateur crée

```
sudo ssh slaforge@10.255.150.114
```

On change notre hostname dans le fichier /etc/hostname

Avant de commencer le TP il ne faudra pas oublier de relancer le service asterisk après toutes modifications d'un des fichiers de configurations avec la commande suivante :

```
service asterisk restart
```

On peut aussi directement faire un «reload» dans le CLI d'asterisk

1) Prise en main du logiciel avec article GLMF

Dans la VM on commence par installer asterisk avec apt install :

```
apt install asterisk
```

Maintenant on passe à la configuration des fichiers :

1) On met le fichier /etc/asterisk/sip.conf de la façon suivante :

```
[71]
type=friend
host=dynamic
username=agent0
secret=agent0
disallow=all

[72]
type=friend
host=dynamic
username=agent1
secret=agent1
disallow=all

[general]
allow=alaw
```

Dans ce fichier on force de codec à utiliser la Loi A (Loi Européenne) et on crée des abonnés

2) Approfondissement sur sip.conf et extensions.conf

Pour cette partie nous allons devoir configurer deux postes softphones pour permettre une communication entre les deux machines

Pour ce faire nous devons modifier le fichier sip.conf pour configurer les deux postes et le fichier extensions.conf pour accepter le protocole SIP sur les numéros commençants par 6X et 7X

Fichier sip.conf :

```
[61]
type=friend
host=dynamic
username=61
secret=61
disallow=all
allow=alaw

[62]
type=friend
host=dynamic
username=62
secret=62
disallow=all
allow=alaw
```

Fichier extensions.conf :

```
[public]
exten => _7X,1,Dial(SIP/${EXTEN})
exten => _6X,1,Dial(SIP/${EXTEN})
```

Cette configuration est minimale pour que les deux machines puissent communiquer.

3) Étude des flux

Maintenant nous allons étudier sur le serveur asterisk les flux entre les différentes machines A et B

Avant de commencer les appels il nous faut installer un logiciel se connectant au serveur asterisk. Nous avons donc choisis de télécharger Zoiper. Nous devons télécharger le paquet .xz sur internet.

Une fois téléchargé on le décompresse pour l'installer

Une fois Zoiper installé on le lance et nous devons le configurer de la façon suivante :

On se connecte en tant que free user

On entre le nom de l'utilisateur suivis du nom de domaine et on se connecte. Le mot de passe et l'utilisateur que vous mettrez seront ceux que vous avez configurés dans le fichier extensions.conf

Vous pouvez configurer les périphériques et ensuite utiliser zoiper

Pour l'étude des flux nous allons nous connecter au serveur et utiliser wireshark sur l'interface du serveur

A appelle B, A raccroche :

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.214.10.1	10.255.150.114	RTCP	74	
Receiver Report Source description						
Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0						
Ethernet II, Src: HewlettP_2d:df:4c (d0:7e:28:2d:df:4c), Dst: 26:39:3e:b9:6c:69 (26:39:3e:b9:6c:69)						
Internet Protocol Version 4, Src: 10.214.10.1, Dst: 10.255.150.114						
User Datagram Protocol, Src Port: 8001, Dst Port: 11875						
Real-time Transport Control Protocol (Receiver Report)						
Real-time Transport Control Protocol (Source description)						
No.	Time	Source	Destination	Protocol	Length	Info
2	0.000177665	10.214.10.1	10.255.150.114	SIP/SDP	711	
Status: 200 OK						
Frame 2: 711 bytes on wire (5688 bits), 711 bytes captured (5688 bits) on interface 0						
Ethernet II, Src: HewlettP_2d:df:4c (d0:7e:28:2d:df:4c), Dst: 26:39:3e:b9:6c:69 (26:39:3e:b9:6c:69)						
Internet Protocol Version 4, Src: 10.214.10.1, Dst: 10.255.150.114						
User Datagram Protocol, Src Port: 5060, Dst Port: 5060						
Session Initiation Protocol (200)						
No.	Time	Source	Destination	Protocol	Length	Info
3	0.001406024	10.255.150.114	10.214.10.1	SIP	426	Request:
ACK sip:62@10.214.10.1						
Frame 3: 426 bytes on wire (3408 bits), 426 bytes captured (3408 bits) on interface 0						
Ethernet II, Src: 26:39:3e:b9:6c:69 (26:39:3e:b9:6c:69), Dst: HewlettP_2d:df:4c (d0:7e:28:2d:df:4c)						
Internet Protocol Version 4, Src: 10.255.150.114, Dst: 10.214.10.1						

User Datagram Protocol, Src Port: 5060, Dst Port: 5060
Session Initiation Protocol (ACK)

No.	Time	Source	Destination	Protocol	Length	Info
4	0.001913833	10.255.150.114	10.214.9.1	SSH	238	Server: Encrypted packet (len=172)

Frame 4: 238 bytes on wire (1904 bits), 238 bytes captured (1904 bits) on interface 0
Ethernet II, Src: 26:39:3e:b9:6c:69 (26:39:3e:b9:6c:69), Dst: HewlettP_2d:df:4c (d0:7e:28:2d:df:4c)
Internet Protocol Version 4, Src: 10.255.150.114, Dst: 10.214.9.1
Transmission Control Protocol, Src Port: 22, Dst Port: 47524, Seq: 1, Ack: 1, Len: 172
SSH Protocol

No.	Time	Source	Destination	Protocol	Length	Info
5	0.002858367	10.255.150.114	10.214.9.1	SIP/SDP	794	Status: 200 OK

Frame 5: 794 bytes on wire (6352 bits), 794 bytes captured (6352 bits) on interface 0
Ethernet II, Src: 26:39:3e:b9:6c:69 (26:39:3e:b9:6c:69), Dst: HewlettP_2d:df:4c (d0:7e:28:2d:df:4c)
Internet Protocol Version 4, Src: 10.255.150.114, Dst: 10.214.9.1
User Datagram Protocol, Src Port: 5060, Dst Port: 5060
Session Initiation Protocol (200)

No.	Time	Source	Destination	Protocol	Length	Info
6	0.003046777	10.214.9.1	10.255.150.114	TCP	66	47524 → 22 [ACK] Seq=1 Ack=173 Win=1444 Len=0 TSval=3237479719 TSecr=149607645

Frame 6: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: HewlettP_2d:df:4c (d0:7e:28:2d:df:4c), Dst: 26:39:3e:b9:6c:69 (26:39:3e:b9:6c:69)
Internet Protocol Version 4, Src: 10.214.9.1, Dst: 10.255.150.114
Transmission Control Protocol, Src Port: 47524, Dst Port: 22, Seq: 1, Ack: 173, Len: 0

No.	Time	Source	Destination	Protocol	Length	Info
7	0.004277984	10.255.150.114	10.214.9.1	SSH	198	Server: Encrypted packet (len=132)

Frame 7: 198 bytes on wire (1584 bits), 198 bytes captured (1584 bits) on

interface 0
Ethernet II, Src: 26:39:3e:b9:6c:69 (26:39:3e:b9:6c:69), Dst: HewlettP_2d:df:4c (d0:7e:28:2d:df:4c)
Internet Protocol Version 4, Src: 10.255.150.114, Dst: 10.214.9.1
Transmission Control Protocol, Src Port: 22, Dst Port: 47524, Seq: 173, Ack: 1, Len: 132
SSH Protocol

No.	Time	Source	Destination	Protocol	Length	Info
8	0.005348896	10.214.9.1	10.255.150.114	TCP	66	47524 → 22 [ACK] Seq=1 Ack=305 Win=1444 Len=0 TSval=3237479721 TSecr=149607646

Frame 8: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: HewlettP_2d:df:4c (d0:7e:28:2d:df:4c), Dst: 26:39:3e:b9:6c:69 (26:39:3e:b9:6c:69)
Internet Protocol Version 4, Src: 10.214.9.1, Dst: 10.255.150.114
Transmission Control Protocol, Src Port: 47524, Dst Port: 22, Seq: 1, Ack: 305, Len: 0

No.	Time	Source	Destination	Protocol	Length	Info
9	0.005458618	10.255.150.114	10.214.9.1	SSH	254	Server: Encrypted packet (len=188)

Frame 9: 254 bytes on wire (2032 bits), 254 bytes captured (2032 bits) on interface 0
Ethernet II, Src: 26:39:3e:b9:6c:69 (26:39:3e:b9:6c:69), Dst: HewlettP_2d:df:4c (d0:7e:28:2d:df:4c)
Internet Protocol Version 4, Src: 10.255.150.114, Dst: 10.214.9.1
Transmission Control Protocol, Src Port: 22, Dst Port: 47524, Seq: 305, Ack: 1, Len: 188
SSH Protocol

No.	Time	Source	Destination	Protocol	Length	Info
10	0.006452994	10.214.9.1	10.255.150.114	TCP	66	47524 → 22 [ACK] Seq=1 Ack=493 Win=1444 Len=0 TSval=3237479722 TSecr=149607646

Frame 10: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: HewlettP_2d:df:4c (d0:7e:28:2d:df:4c), Dst: 26:39:3e:b9:6c:69 (26:39:3e:b9:6c:69)
Internet Protocol Version 4, Src: 10.214.9.1, Dst: 10.255.150.114
Transmission Control Protocol, Src Port: 47524, Dst Port: 22, Seq: 1, Ack: 493, Len: 0

No.	Time	Source	Destination	Protocol	Length	Info
11	0.006545855	10.255.150.114	10.214.10.1	SIP/SDP	834	

Request: INVITE sip:62@10.214.10.1, in-dialog |

Frame 11: 834 bytes on wire (6672 bits), 834 bytes captured (6672 bits) on interface 0
Ethernet II, Src: 26:39:3e:b9:6c:69 (26:39:3e:b9:6c:69), Dst: HewlettP_2d:df:4c (d0:7e:28:2d:df:4c)
Internet Protocol Version 4, Src: 10.255.150.114, Dst: 10.214.10.1
User Datagram Protocol, Src Port: 5060, Dst Port: 5060
Session Initiation Protocol (INVITE)

No.	Time	Source	Destination	Protocol	Length	Info
12	0.007276513	10.214.10.1	10.255.150.114	SIP/SDP	711	

Status: 200 OK |

Frame 12: 711 bytes on wire (5688 bits), 711 bytes captured (5688 bits) on interface 0
Ethernet II, Src: HewlettP_2d:df:4c (d0:7e:28:2d:df:4c), Dst: 26:39:3e:b9:6c:69 (26:39:3e:b9:6c:69)
Internet Protocol Version 4, Src: 10.214.10.1, Dst: 10.255.150.114
User Datagram Protocol, Src Port: 5060, Dst Port: 5060
Session Initiation Protocol (200)

No.	Time	Source	Destination	Protocol	Length	Info
13	0.008212642	10.255.150.114	10.214.10.1	SIP	426	

Request: ACK sip:62@10.214.10.1 |

Frame 13: 426 bytes on wire (3408 bits), 426 bytes captured (3408 bits) on interface 0
Ethernet II, Src: 26:39:3e:b9:6c:69 (26:39:3e:b9:6c:69), Dst: HewlettP_2d:df:4c (d0:7e:28:2d:df:4c)
Internet Protocol Version 4, Src: 10.255.150.114, Dst: 10.214.10.1
User Datagram Protocol, Src Port: 5060, Dst Port: 5060
Session Initiation Protocol (ACK)

No.	Time	Source	Destination	Protocol	Length	Info
20	0.028636537	10.214.9.1	10.255.150.114	SIP	501	

Request: ACK sip:62@10.255.150.114:5060 |

Frame 20: 501 bytes on wire (4008 bits), 501 bytes captured (4008 bits) on interface 0
Ethernet II, Src: HewlettP_2d:df:4c (d0:7e:28:2d:df:4c), Dst: 26:39:3e:b9:6c:69 (26:39:3e:b9:6c:69)
Internet Protocol Version 4, Src: 10.214.9.1, Dst: 10.255.150.114

User Datagram Protocol, Src Port: 5060, Dst Port: 5060
Session Initiation Protocol (ACK)

No.	Time	Source	Destination	Protocol	Length	Info
21	0.028876943	10.255.150.114	10.214.9.1	SIP/SDP	811	

Request: INVITE sip:61@10.214.9.1, in-dialog |

Frame 21: 811 bytes on wire (6488 bits), 811 bytes captured (6488 bits) on interface 0
Ethernet II, Src: 26:39:3e:b9:6c:69 (26:39:3e:b9:6c:69), Dst: HewlettP_2d:df:4c (d0:7e:28:2d:df:4c)
Internet Protocol Version 4, Src: 10.255.150.114, Dst: 10.214.9.1
User Datagram Protocol, Src Port: 5060, Dst Port: 5060
Session Initiation Protocol (INVITE)

No.	Time	Source	Destination	Protocol	Length	Info
22	0.032310773	10.214.9.1	10.255.150.114	RTCP	86	

Receiver Report Goodbye

Frame 22: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
Ethernet II, Src: HewlettP_2d:df:4c (d0:7e:28:2d:df:4c), Dst: 26:39:3e:b9:6c:69 (26:39:3e:b9:6c:69)
Internet Protocol Version 4, Src: 10.214.10.1, Dst: 10.255.150.114
User Datagram Protocol, Src Port: 8001, Dst Port: 11875
Real-time Transport Control Protocol (Receiver Report)
Real-time Transport Control Protocol (Goodbye)

No.	Time	Source	Destination	Protocol	Length	Info
23	0.032326454	10.214.9.1	10.255.150.114	SIP/SDP	717	

Status: 200 OK |

Frame 23: 717 bytes on wire (5736 bits), 717 bytes captured (5736 bits) on interface 0
Ethernet II, Src: HewlettP_2d:df:4c (d0:7e:28:2d:df:4c), Dst: 26:39:3e:b9:6c:69 (26:39:3e:b9:6c:69)
Internet Protocol Version 4, Src: 10.214.9.1, Dst: 10.255.150.114
User Datagram Protocol, Src Port: 5060, Dst Port: 5060
Session Initiation Protocol (200)

No.	Time	Source	Destination	Protocol	Length	Info
24	0.032785097	10.255.150.114	10.214.9.1	SIP	403	

Request: ACK sip:61@10.214.9.1 |

Frame 24: 403 bytes on wire (3224 bits), 403 bytes captured (3224 bits) on interface 0

Ethernet II, Src: 26:39:3e:b9:6c:69 (26:39:3e:b9:6c:69), Dst: HewlettP_2d:df:4c (d0:7e:28:2d:df:4c)
Internet Protocol Version 4, Src: 10.255.150.114, Dst: 10.214.9.1
User Datagram Protocol, Src Port: 5060, Dst Port: 5060
Session Initiation Protocol (ACK)

No.	Time	Source	Destination	Protocol	Length	Info
25	0.033062124	10.255.150.114	10.214.9.1	SSH	238	Server: Encrypted packet (len=172)

Frame 25: 238 bytes on wire (1904 bits), 238 bytes captured (1904 bits) on interface 0
Ethernet II, Src: 26:39:3e:b9:6c:69 (26:39:3e:b9:6c:69), Dst: HewlettP_2d:df:4c (d0:7e:28:2d:df:4c)
Internet Protocol Version 4, Src: 10.255.150.114, Dst: 10.214.9.1
Transmission Control Protocol, Src Port: 22, Dst Port: 47524, Seq: 1205, Ack: 1, Len: 172
SSH Protocol

No.	Time	Source	Destination	Protocol	Length	Info
26	0.034376286	10.214.9.1	10.255.150.114	TCP	66	47524 → 22 [ACK] Seq=1 Ack=1377 Win=1444 Len=0 TSval=3237479750 TSecr=149607653

Frame 26: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: HewlettP_2d:df:4c (d0:7e:28:2d:df:4c), Dst: 26:39:3e:b9:6c:69 (26:39:3e:b9:6c:69)
Internet Protocol Version 4, Src: 10.214.9.1, Dst: 10.255.150.114
Transmission Control Protocol, Src Port: 47524, Dst Port: 22, Seq: 1, Ack: 1377, Len: 0

No.	Time	Source	Destination	Protocol	Length	Info
27	0.068188590	10.214.10.1	10.255.150.114	RTCP	86	Receiver Report Goodbye

Frame 27: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
Ethernet II, Src: HewlettP_2d:df:4c (d0:7e:28:2d:df:4c), Dst: 26:39:3e:b9:6c:69 (26:39:3e:b9:6c:69)
Internet Protocol Version 4, Src: 10.214.9.1, Dst: 10.255.150.114
User Datagram Protocol, Src Port: 8001, Dst Port: 12333
Real-time Transport Control Protocol (Receiver Report)
Real-time Transport Control Protocol (Goodbye)

On voit dans les captures que quand la machine A appelle la machine B elle passe d'abord par une connexion sur le serveur et fait la demande de dialogue. Le serveur répond qu'il a bien reçu la demande et va aller envoyer la requête à la machine B. La machine répond à la requête et les deux machines sont mises en communication. Lorsque la machine A demande à raccrocher elle envoie la demande au serveur qui lui envoie ensuite la requête à la machine B puis la communication se coupe.

A appelle B, B raccroche :

No.	Time	Source	Destination	Protocol	Length	Info
73	2.117072765	10.214.10.1	10.255.150.114	RTCP	86	Receiver Report Goodbye

Frame 73: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
Ethernet II, Src: HewlettP_2d:df:4c (d0:7e:28:2d:df:4c), Dst: 26:39:3e:b9:6c:69 (26:39:3e:b9:6c:69)
Internet Protocol Version 4, Src: 10.214.10.1, Dst: 10.255.150.114
User Datagram Protocol, Src Port: 8001, Dst Port: 16461
Real-time Transport Control Protocol (Receiver Report)
Real-time Transport Control Protocol (Goodbye)

Cette fois-ci rien ne change de l'appel précédent. Il n'y a que lors de la fin de l'appel qui est inversé par rapport au premier appel car cette fois-ci c'est la machine B qui demande la fin de la communication et non la machine A.

4) Mettre en place la fonction "echo" au numéro 3666

Pour cette partie nous devons mettre en place une fonction "echo" au numéro 3666

Tout d'abord qu'est-ce que c'est et à quoi elle sert ? La fonction echo permet de répéter tout ce que l'utilisateur dit. Son principe est qu'elle permet de vérifier le bon fonctionnement du micro (du casque jusqu'à la conversion de la voix en numérique). Elle permet de voir la latence entre le temps où l'on parle et que la machine reçoit.

Installation de la fonction "echo" :

Pour installer cette fonction on a seulement à ajouter 3 extensions dans le fichier extensions.conf :

```
exten => 3666,1,Playback(demo-echotest) ; Explication de ce qui se passe et le $
exten => 3666,2,Echo ; Fait le test d'écho.
exten => 3666,3,Playback(demo-echodone) ; Indique la fin de l'écho test
```

Les chiffres 1, 2 et 3 indiquent les étapes (l'ordre dans lequel s'exécute les extensions)

5) Mettre en place une horloge parlante au numéro 3699

Pour mettre en place une horloge parlante au numéro 3699 il faut d'abord installer les paquets voix d'asterisk en français :

```
apt install asterisk-core-sounds-fr  
apt install asterisk-core-sounds-fr-g722  
apt install asterisk-core-sounds-fr-gsm  
apt install asterisk-core-sounds-fr-wav
```

Maintenant il faut vérifier la présence de ces paquets dans /usr/share/asterisk/sounds/fr

Après on a plus qu'à terminer par ajouter les extensions dans le fichier extensions.conf :

```
exten => 3699,1,Answer  
    same => n,Playback(welcome)  
    same => n,SayUnixTime(,,Abdy HMS)  
    same => n,Playback(vm-goodbye)  
    same => n,Hangup()
```

6) Mettre en place la lecture de la météo au numéro 3030

Pour mettre en place la météo sur asterisk, nous devons installer espeak sur notre VM ainsi que toutes les librairies et dépendances.

```
Apt install espeak
```

Nous sommes allé récupérer un git destiné à espeak pour asterisk.

Il nous a mentionné des librairies manquantes.

On les a donc installées :

```
apt install asterisk-dev  
apt install asterisk-espeak  
apt install libespeak-ng-libespeak-dev  
apt install libsamplerate-dev
```

Une fois tout ça installé nous avons créé un script qui va récupérer les informations utiles grâce à la commande weather. Bien-sûr il faut également installer le paquet weather.

```
Apt install weather-util  
apt install weather-util-data
```

Pour la météo nous avons tout simplement crée un script qui sera exécuté toute les heures via la crontab : crontab -e - > 00 */1 * * * /home/aholsteing/meteo.sh

Afin de garder le fichiers de météo a jour. Le script de météo a pour but d'utiliser la commande weather de linux et remise en forme ensuite dans un fichier .txt qui sera ensuite lu par asterisk grâce au module espeak lorsqu'on appel le numéro 3030.

Voici le script qui envoie la météo dans le fichier meteo.txt qui est lui lu par asterisk :

```
#!/bin/bash
tempC=$(weather LFMU --headers=temperature | grep -Eo '\-?[1-9]* C' | sed 's/C$
Humi=$(weather lfmU | awk 'NR == 4 {print}' | sed 's/.*/:/')
tempN=$(weather lfmU | awk 'NR == 6 {print}' | sed 's/.*/:/' | sed 's/;.*//g')
echo -e "Il fait $tempC degré celsius à béziers" > meteo.txt
echo -e "Avec une humidité de $Humi" >> /home/aholsteing/meteo.txt
echo -e "Le météo est plutôt $tempN" >> /home/aholsteing/meteo.txt
chmod 766 meteo.txt
```

Ce script à pour but d'envoyer les informations vers un fichier texte qui sera lu ensuite par espeak avec une extension.

Voici à quoi ressemble le contenu du fichier .txt après la mise en forme via le script :

```
Il fait 8 degré celcius à Béziers
Avec une humidité de 93%
La météo est plutôt light rain
```

Voici l'extension utilisé pour faire lire le fichier meteo.txt par espeak de asterisk :

```
[public]
exten => 3030,1,Answer();
exten => 3030,2,System(/etc/asterisk/meteo.sh);
exten => 3030,3,Set(METEO=${FILE(/etc/asterisk/meteo.txt)});
exten => 3030,4,Espeak("${METEO}",any,fr);
```

7) Jeu "trouver un nombre entre 1 et 99" au numéro 3535

Avant de commencer il faut installer la fonction IVR. IVR (Interactive Voice Response) permet de créer des répondeurs automatiques. Cela permet notamment d'accueillir un client et de lui proposer plusieurs choix afin de le rediriger vers le service adapté à son appel.

Pour se faire on installe d'abord un paquet pour permettre à Asterisk de parler (ici on installe googleTTS). On commence par installer les dépendances :

```
apt-get install perl libwww-perl sox mpg123
```

Ensuite on se déplace dans le dossier /var/lib/asterisk/agi-bin et on y installe googleTTS :

```
cd /var/lib/asterisk/agi-bin  
wget https://raw.githubusercontent.com/zaf/asterisk-googleletts/master/googleletts.agi  
chmod +x googleletts.agi
```

Une fois googleTTs installé on configure le fichier extension.conf pour qu'Asterisk puisse parler. On commence avec un test :

```
;Premier IVR  
exten => 8000,1,Answer()  
exten => 8000,2,agi(googleletts.agi,"Par Toutatix je parle !",fr)  
exten => 8000,3,Hangup()
```

Ici si on appelle le numéro 8000 on va avoir Asterisk qui décroche puis qui va nous dire "Par Toutatix je parle (en français) pour ensuite raccrocher en dernière étape.

Une fois la fonction installée on a juste à créer un script qui va générer un nombre aléatoire entre 1 et 99 et l'ajouter dans les extensions.

8) Écouter NRJ au numéro 3636

Flux radio de NRJ : http://cdn.nrjaudio.fm/audio1/fr/30001/mp3_128.mp3

Pour pouvoir écouter n'importe quelle radio sur asterisk il nous faut tout d'abord avoir le flux de la radio voulu sous forme d'URL http. Ensuite il nous faudra utiliser la fonction MP3Player d'asterisk pour pouvoir écouter ce flux.

Une fois que nous avons le flux il nous suffit de mettre deux extensions dans le fichier extensions.conf :

```
exten => 3636,1,Answer()  
exten =>  
3636,2,MP3Player(http://cdn.nrjaudio.fm/audio1/fr/30001/mp3\_128.mp3)
```

Une fois les extensions mises on a juste à relancer asterisk et lancer un appel sur twinkle avec le numéro qui correspond à la radio (ici 3636)

9) Hardphone et transfert d'appel

Configuration et mise en place du transfert d'appel :

Dans le fichier **features.conf** se trouvant dans le repertoire **/etc/asterisk**,il faut repérer les lignes suivantes :

```
;blindxfer => #1;  
;atxfer => *2;
```

Ces ligne permettent de configurer une ou plusieurs touches pour transferer un appel.

Pour ma part j'ai dans ce fichier de conf mis comme **touches ## pour le transfert aveugle et ** pour le transfert supervisé**, donc au final les lignes ressemblent à ceci:

```
blindxfer => ##;  
atxfer => **;
```

Puis dans le fichier **extensions.conf** on rajoute l'option **tT** à l'application Dial()

Donc la ligne:

```
exten => _6XXX,1,Dial(SIP/${EXTEN},20)
```

Devient:

```
exten => _6XXX,1,Dial(SIP/${EXTEN},20,tT)
```

Maintenant, après avoir fais un reload d'Asterisk on peut au cours d'un appel appuyer sur les touches **##** pour un transfert aveugle ou ****** pour un transfert supervisé, une voix dira «Transfert» on n'aura juste à taper le numéro de téléphone de la personne à qui on souhaite transférer l'appel.