Un répondeur téléphonique sous Linux pour moins de 9 euros

Gazette Linux n°120 — Novembre 2005

Bob Smith

Article paru dans le n°120 de la Gazette Linux de novembre 2005.
Traduction française par Florent Revelut <florent POINT revelut CHEZ free POINT fr>.
Relecture de la traduction française par Joëlle Cornavin <jcornavi CHEZ club TIRET internet POINT fr>.
Article publié sous Open Publication License (http://linuxgazette.net/copying.html). La Linux Gazette n'est ni produite, ni sponsorisée, ni avalisée par notre hébergeur principal, SSC, Inc.

Table des matières

1. Introduction

Cet article décrit comment construire un répondeur téléphonique inspiré de Linux à l'aide d'un Winmodem (modem logiciel) bon marché. Nous expliquerons comment installer les pilotes et les bibliothèques, comment choisir et installer la carte modem la plus adaptée. Notre programme de répondeur téléphonique se compose de quelques centaines de lignes de code écrites en C dans un seul fichier. Si vous avez déjà utilisé un téléphone, vous ne devriez avoir aucun problème à comprendre le code.

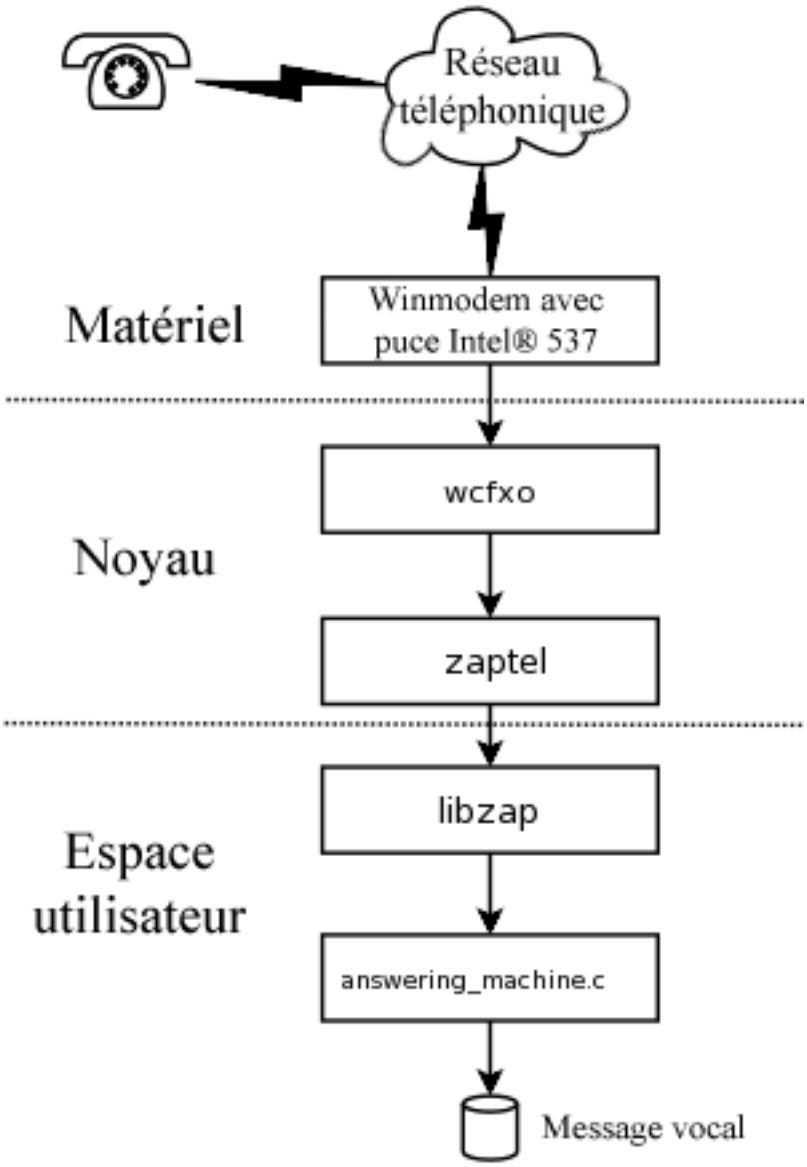
Les prérequis pour ce projet sont les suivants :

- Une carte modem reposant sur une puce Intel® 537 (*softmodem*, modem logiciel) ;
- Un emplacement PCI (*Peripheral Component Interconnect*, interconnexion de composants) qui ne partage pas d'interruptions.

Les cartes modem reposant sur des puces Intel® 537 sont disponibles pour moins de 9 euros actuellement. Comme le pilote que nous allons utiliser requiert cette puce, assurez-vous d'avoir le modem est correct avant de commencer.

2. Architecture d'une interface téléphonique sous Linux

Notre répondeur utilise des pilotes et des bibliothèques qui font partie du PBX (*Private Branch eXchange*, commutateur privé) Asterisk *open source*. Asterisk peut fonctionner en tant que répondeur téléphonique, mais c'est une application énorme et qui souffre d'une forte courbe d'apprentissage en terme d'installation et de configuration. Le paquetage Asterisk décompressé fait plus de 16 méga-octets, soit bien plus que les 5 kilo-octets de notre fichier source. L'architecture globale de notre répondeur téléphonique est présentée dans le diagramme ci-dessous.



Le pilote de la carte modem est le module wcfxo, qui s'interface vers un pilote de plus haut niveau appelé zaptel. La division du pilote en deux parties aide les développeurs Asterisk à minimiser la quantité de code qu'ils doivent écrire pour de nouveaux types de cartes d'interface téléphonique pour PC. Le pilote zaptel fournit à l'application de l'espace utilisateur un flux d'échantillons *mu-law* de 64 kb/s. zaptel comporte également des IOCTL IOCTL (*IO Control Routines*, routines de contrôle des entrées/sorties) pour décrocher la ligne, raccrocher ou indiquer que la ligne téléphonique sonne.

La bibliothèque Zapata (libzap) traite le flux de données de 64 kb/s pour extraire l'identifiant de l'appelant et les signaux DTMF (*Dual Tone MultiFrequency*, multifréquence à double tonalité). Les capacités de la `libzap` explique pourquoi notre répondeur est aussi simple.

Notre répondeur téléphonique correspond au strict minimum en tant qu'application téléphonique. Il extrait et affiche les informations d'identifiant de l'appelant pour les appels entrants. Si le téléphone sonne plus de quatre fois, il décroche et lit un message d'accueil. Celui-ci demande à l'appelant

d'appuyer sur la touche « 1 » pour laisser un message. Obliger l'appelant à appuyer sur un bouton élimine les messages de ce qu'on appelle les « systèmes d'appels automatiques » (*bulk dialers*). Le message vocal est enregistré sous forme de données encodées en *mu-law* dans un fichier dont le nom contient la date et l'heure.

3. Installation des pilotes et des bibliothèques

Notre répondeur téléphonique requiert les pilotes zaptel et la bibliothèque Zapata, disponibles tous les deux sur le site web d'Asterisk (http://www.asterisk.org/).

3.1. Installation des pilotes

Deux pilotes, wcfxo et zaptel, sont nécessaires. Ils se trouvent l'un et l'autre dans le paquetage zaptel. Vous pouvez obtenir ce paquetage via le lien Downloads (http://www.asterisk.org/download) sur la page d'accueil de Asterisk ou directement sur le site de Digium (http://www.digium.com/), qui héberge le site de téléchargement :

http://ftp.digium.com/pub/zaptel/zaptel-1.0.9.2.tar.gz

Désarchivez le fichier, faites un **make linux26** et un **make install** (en tant que *root*). Si vous utilisez udev, suivez scrupuleusement les directives du fichier `README.udev` (en anglais). En particulier, vous devez ajouter les lignes suivantes à votre fichier /etc/udev/rules.d/50-udev.rules :

Comme j'ai démarré le répondeur sous mon nom d'utilisateur (*`bobsmith`*), j'ai ajouté la ligne suivante au fichier de permissions udev (`/etc/udev/permissions.d/00-udev.permissions`) :

La dernière partie de la configuration de zaptel consiste à lui indiquer que nous avons une interface au téléphone « central téléphonique ». Modifiez le fichier `/etc/zaptel.conf` et ajoutez la ligne suivante à la fin :

Si vous ne vivez pas aux États-Unis, un autre changement dans le fichier `zaptel.conf` sera nécessaire. Vous devez décommenter la ligne contenant votre code de pays et faire de ce code de pays la zone par défaut.

3.2. Installation de la bibliothèque zapata

Du fait que Asterisk n'utilise plus la bibliothèque zapata, cette dernière a été déplacée dans le répertoire `old` sur le site de téléchargement :

http://ftp.digium.com/pub/zaptel/old/zapata-0.9.1.tar.gz

Désarchivez le fichier, faites un **make** et un **make install** (en tant que *root*). La bibliothèque Zapata ne nécessite aucune configuration.

3.3. Sélection et installation du modem Intel®

Comme indiqué plus haut, vous devez être sûr de disposer d'un modem compatible avec la puce Intel® 537. Ces modems devraient être faciles à trouver et relativement bon marché.

L'appeler « modem » n'est pas tout à fait correcte. Il s'agit en effet plus d'une « interface vers une ligne téléphonique » dont le flux de 64 kb/s apporte 8 000 octets par seconde, que le couple de pilotes wcfxo/zaptel apporte en blocs de 8 octets. Cela signifie concrètement un millier d'interruptions par seconde ! C'est cette charge sur les interruptions qui rend l'installation de la carte modem un peu délicate. Vous devez installer la carte dans un emplacement PCI qui n'est pas partagé avec un autre périphérique. Vérifiez dans la documentation de votre carte mère pour savoir quels emplacements utilisent quelles interruptions et comment les périphériques internes emploient ces interruptions. Si vous n'avez vraiment pas de chance, vous constaterez peut-être que l'interruption pour chaque emplacement est partagée et utilisée. Vous pouvez essayer de désactiver des périphériques internes, mais il y a un risque que wcfxo ne fonctionnera tout simplement pas sur certaines cartes mères.

L'autre problème, moins courant, est que le pilote wcfxo ne reconnaisse pas votre modem. Pour le résoudre, modifiez la table wcfxo_pci_tbl vers la fin du fichier `wcfxo.c` dans le répertoire de compilation de zaptel. Utilisez **lspci -nv** pour récupérer l'identifiant du fabricant, du produit et du sous-système. Les identifiants suivants de fabricant, de produit et de sous-système sont déjà reconnus par wcfxo :

- e159:0001 8085
- e159:0001 8086
- e159:0001 8087

3.4. Vérification du système

Si vous avez terminé l'installation matérielle et logicielle, vous pouvez vérifier votre système en quelques commandes. Installez les modules et examinez la sortie syslog obtenue :

Ne vous inquiétez pas du message « Sharing IRQ 5 ». Il vous informe que l'IRQ peut être partagée, mais cela n'implique pas qu'un autre périphérique Linux s'en sert en ce moment. Un **lsmod** devrait afficher les trois modules :

Vérifiez que le pilote wcfxo génère un millier d'interruptions par seconde et qu'elles ne sont pas partagées avec tout autre périphérique :

L'élément wcfxo dans `/proc/interrupts` devrait être une ligne à elle seule et le compteur d'interruption devrait avoir avancé de dix mille pendant les dix secondes d'attente ci-dessus. Si wcfxo n'a pas une ligne pour lui tout seul, vous devez déplacer le modem sur un autre emplacement PCI. Plus précisément, vous voulez quelque chose comme :

et non quelque chose comme :

Le pilote zaptel contient un utilitaire pour visualiser sa configuration. Servez-vous en pour vous assurer qu'il a une interface. Si ce test échoue, vérifiez que vous avez la ligne **fxsks=1** dans votre fichier de configuration.

4. Installation et utilisation du répondeur téléphonique

Le code pour le répondeur téléphonique est un seul fichier C qui est disponible ici (outils/lg120-C/answering_machine.c). Compilez le programme avec la commande suivante :

Lancez le programme avec la commande suivante :

Le programme s'attend à trouver un message d'accueil encodé en *mu-law* à 8 kHz appelé « leave_a_msg.ul » dans le répertoire en cours. Vous pouvez enregistrer l'annonce avec l'utilitaire de votre choix et ensuite convertir le fichier WAV en *mu-law* à l'aide de **sox**. La commande à utiliser est la suivante :

Sox interprète les fichiers portant l'extension `.ul` comme un format audio encodé en *mu-law* à 8 kHz. Vous pouvez entendre vos messages vocaux en utilisant la commande **play**. Par exemple :

5. Revue de code du répondeur téléphonique

Le travail difficile dans le répondeur téléphonique est effectué par la bibliothèque Zapata. Elle gère la détection de l'identifiant de l'appelant, des signaux DTMF, lit et écrit les fichiers audio *mu-law*. Voici le squelette du code du répondeur téléphonique :

Le code ci-dessus devrait vous donner une idée de la manière d'écrire des applications téléphoniques à l'aide de la bibliothèque Zapata. La bibliothèque a également des fonctions pour passer des appels sortants et mettre en place des conférences téléphoniques.

6. Étapes suivantes...

Pour un répondeur téléphonique simple, ce programme fonctionne étonnament bien. En fait, sa simplicité est sa meilleure qualité.

Si l'on devait ajouter des fonctionnalités supplémentaires à ce répondeur, on ajouterait probablement :

- L'accès à distance aux messages vocaux : nous devrions être en mesure de récupérer nos messages vocaux sur une ligne téléphonique. Les enregistrer dans un fichier nous donne déjà un accès distant. Cependant...
- Le routage de l'identifiant de l'appelant : l'appel serait routé en fonction du numéro de téléphone de l'appelant. Certains appelants accéderaient directement à la messagerie vocale.
- La captures des appels sortants : le répondeur téléphonique devrait être en mesure de journaliser le numéro appelé et la durée des appels sortants.
- L'enregistrement des appels en cours : ne serait-il pas pratique de pouvoir juste appuyer sur une touche pour que votre appel actuel soit enregistré ? Ce serait une méthode beaucoup plus fiable pour prendre des notes pendant un appel. Bien sûr, vous seriez amené à informer votre interlocuteur que vous enregistrez l'appel et vous feriez biper la machine toutes les dix secondes, ou similaire.
- Une interface *web* : il pourrait être agréable d'avoir une interface *web* pour configurer le répondeur. Quelque chose comme la bibliothèque Run Time Access pourrait se révéler utile. (Consultez http://www.runtimeaccess.com (http://www.runtimeaccess.com).)
- La remise par courrier électronique des messages vocaux : nous pourrions convertir les fichiers de courrier vocal du *mu-law* en WAV et nous les faire envoyer par courrier électronique.
- La transmission des messages/avertissement par récepteur de radio-messagerie : il s'agit d'utiliser la capacité de libzap à émettre des appels sortants pour informer un récepteur de radio-messagerie qu'il y

a un nouveau message ou de demander à l'application d'appeler notre téléphone portable et de lire le message vocal.

La bibliothèque Zapata utilise des entrées/sorties bloquantes, ce qui complique un peu l'emploi d'une boucle `select()` pour gérer une interface utilisateur lors de l'attente des messages entrants. Peut-être une application à plusieurs fils d'exécution (*threaded*) contournerait-elle le problème.

Bob est électronicien amateur et programmeur Linux. Il est un des auteurs de *Linux Appliance Design* qui doit être publié par No Starch Press®.