Bip visuel mini-Howto (Visual bell mini-Howto)

Alessandro Rubini, rubini@linux.it
Traduction Jean-Albert Ferrez, Jean-Albert.Ferrez@epfl.ch
v2.2, 11 Novembre 1997

Ce document explique comment utiliser termcap pour configurer un bip visuel sur son système et comment désactiver à la demande le bip sonore.

Contents

1 Introduction

La console Linux émet un bip sonore lorsqu'un caractère BEL (code ASCII 7) est produit. Bien que ce soit un choix raisonnable dans la plupart des cas, bien des utilisateurs n'aiment pas que leur ordinateur fasse du bruit. Ce mini-howto est censé expliquer comment indiquer aux applications de ne pas produire de caractère BEL. Il explique également comment faire en sorte que le noyau et le serveur X Window ne sonnent pas en présence d'un caractère BEL. Notez que la plupart de ce document concerne la console texte, étant donné que la configuration du serveur X est triviale pour ceux qui travaillent dans un environnement graphique.

A mon avis, la meilleure attitude à adopter face à un ordinateur bruyant consiste à opérer au niveau du matériel, le mien n'a même pas de haut-parleur.

2 "Spekearectomy", ablation du haut-parleur

L'ablation du haut-parleur est de loin la solution la plus brillante au problème du bip sonore. Comme son nom l'indique, elle consiste à supprimer le bip à la source. L'opération est simple, elle ne nécessite pas d'anesthésie.

Les PC sont habituellement équipés d'un bouton ridicule servant à réduire la fréquence du CPU. Ce bouton n'est jamais utilisé dans un environnement multitâche, puisque l'on n'a jamais besoin de ralentir la machine pour utiliser des jeux basés sur des boucles de temporisation. Malheureusement, on ne peut pas utiliser ce bouton pour augmenter la fréquence du CPU, mais on peut l'utiliser pour activer/désactiver le haut-parleur. Parfois, ce dernier est utile même si vous préférez un ordinateur silencieux, par exemple pour signaler la fin d'une longue compilation. Pour modifier la fonction de ce bouton, il suffit de le déconnecter de la carte-mère et de le brancher en série avec le haut-parleur.

Cela dit, les possesseurs de portables n'ont pas accès au haut-parleur, pas plus qu'ils ne disposent d'un bouton inutile à réassigner. La solution pour ces utilisateurs réside dans la configuration de leurs logiciels, comme indiqué ci-dessous.

3 Configuration du bip console par console

Depuis la version 1.3.43, Martin Mares a modifié `console.c`, y ajoutant la possibilité de configurer la hauteur et la durée du bip. Chaque console peut être configurée pour avoir un bip de hauteur et/ou durée différente. Ce but est atteint au moyen de séquences d'échappement interprétées par le gestionnaire de la console. Vous pouvez configurer vos fichiers `~/.profile` ou `~/.login` pour choisir un bip particulier pour chaque console (ou, le cas échéant, pas de bip du tout).

Les séquences d'échappement fonctionnent comme suit :

- ESC-`[10;`xx`]` sélectionne la hauteur du bip en Hertz. La valeur doit être dans la plage 21-32766, sinon le résultat est indéterminé. Si l'argument 'xx' manque (ESC-`[10]`), la valeur par défaut est 750 Hz.
- ESC-`[11;`xx`]` sélectionne la durée du bip en millisecondes. Au delà de 2 secondes, la valeur par défaut de 125 ms est appliquée. Cette dernière est également utilisée si l'argument 'xx' manque.

Pour avoir par exemple un bip à 50Hz durant une seconde, faites `"echo -e "\\33[10;50]\\33[11;1000]""` avec `bash` (le "-e" signifie 'digère les séquences d'échappement'. Avec `tcsh` la commande devient `"echo "\\033[10;50]\\033[11;1000]""`.

Bien que je ne connaisse aucune version de la commande `setterm` qui supporte une telle configuration, il est possible qu'une version future offre une option permettant de configurer le bip.

Si vous utilisez Linux-1.3.43 ou plus récent, vous pouvez vous contenter des séquences d'échappement et interrompre votre lecture ici. Si votre noyau est plus vieux, ou si vous désirez un bip visuel, vous aller aimer le reste de ce document.

4 Concepts de base au sujet de Termcap et Terminfo

Le fichier `/etc/termcap` est un fichier texte contenant les "`term`inal `cap`abilities" (capacités des terminaux, N.d.T.). Plusieurs applications utilisent les informations de `termcap` pour déplacer le curseur sur l'écran ou d'autres actions en relation avec ce dernier. `tcsh`, `bash`, `vi` et toutes les applications basées sur `curses` utilisent la base de données `termcap`.

La base de données décrit plusieurs types de terminaux. La variable d'environnement `TERM` sélectionne le comportement correct lors de l'exécution en indiquant aux applications quelle entrée de `termcap` utiliser.

Dans la base de données, chaque capacité d'un terminal est représentée par un code de deux lettres et une représentation de la séquence d'échappement permettant d'obtenir l'effet désiré. Les deux-points ":" servent de séparateur entre les différentes capacités. Par exemple, le bip sonore dont le code est "bl" apparaît généralement sous la forme "`bl=^G`", ce qui signifie que le bip est obtenu en "affichant" le caractère control-G, correspondant au code ASCII BEL.

En plus de `bl`, la capacité `vb` est reconnue. Elle est utilisée pour représenter le "bip visible". `vb` est généralement absent de l'entrée `linux` du fichier `termcap`.

La plupart des applications et librairies modernes utilisent la base de données `terminfo` au lieu de `termcap`. Cette base de données utilise un fichier par type de terminal et est située dans `/usr/lib/terminfo`; pour éviter de trop grands répertoires, les descriptions de chaque type de terminal sont stockées dans un répertoire nommé d'après la première lettre : ainsi, l'entrée `linux` se situe dans `/usr/lib/terminfo/l/linux`. Pour construire une entrée `terminfo` il faut "compiler" une description `termcap`, voir le programme `tic` et sa page de manuel.

5 Mise sur pied d'un bip visible

Vous pouvez ajouter une entrée pour la capacité `vb` dans votre fichier `termcap`, si ce n'est pas déjà fait. Dennis Henriksen (duke@diku.dk) a suggéré la ligne suivante dans l'entrée `linux` (appelée `console` dans les anciennes distributions) du fichier `termcap` :

```
:vb=\E7\E[?5h\E[?5l\E[?5h\E[?5l\E[?5h\E[?5l\E[?5h\E[?5l\E8:\
```

Le backslash final sert à cacher le retour à la ligne dans la base. Le code de Dennis fonctionne comme suit (selon ses propres mots) :

- Sauvegarde de la position du curseur (par précaution).
- Bascule du fond d'écran plusieurs fois entre normal et inversé.
- Restauration de la position du curseur.

6 Désactivation du bip sonore dans la console texte

Si vous désirez imposer le bip visuel dans la console, vous pouvez utiliser l'entrée "`bl`" dans la `termcap` et lui donner la même valeur que pour "`vb`" ci-dessus. Cette aproche est pratique si vous ne voulez pas configurer chaque application (cela est de toutes façons décrit ci-dessous). J'utilise cette solution sur toutes les machines sur lesquelles j'utilise Linux sans pouvoir déconnecter le haut-parleur.

7 Empêcher les applications d'émettre des bips

Ceci est une liste incomplète d'applications qui peuvent être configurées pour utiliser l'entrée `vb` pour le type de terminal utilisé (utilisant soit `termcap` soit `terminfo`) :

- Le serveur X : utilisez la commande "`xset b`" pour sélectionner le comportement du bip. Elle prend trois arguments numériques : volume, hauteur et durée. "`xset -b`" désactive complètement le bip. La configuration du serveur affecte toutes les applications tournant dessus.
- `xterm` : xterm peut convertir chaque bip soit en un signal visuel soit en un signal sonore. Si vous utilisez un bip sonore, il obéit à la configuration issue de "`xset`". Par défaut `xterm` utilise un bip sonore, mais l'option "`-vb`" sur la ligne de commande ainsi que la ressource "`xterm*visualBell: true`" le remplace par un flash visuel. Vous pouvez passer de l'un à l'autre en cours d'exécution au moyen du menu obtenu par control-clic-bouton-gauche. Si vous travaillez sous X vous n'avez probablement pas besoin des informations qui suivent.
- `tcsh` (6.04 et ultérieurs) : "`set visiblebell`". Cette instruction peut être placée dans `.cshrc` ou exécutée directement. Pour rétablir le bip sonore, "`unset visiblebell`". Pour n'avoir ni l'un ni l'autre, utilisez "`set nobeep`".

- bash (n'importe quel bash, pour autant que je sache) : Mettez "set bell-style visible" dans votre ~/.bashrc. Les autre valeurs possibles sont "none" et "audible".
- bash (avec readline, ainsi que d'autres applications basées sur readline) : Mettez "set prefer-visible-bell" dans ~/.inputrc.
- nvi et elvis : Mettez "set flash" dans ~/.exrc ou faites ":set flash" directement. (notez les deux points). Pour désactiver le bip visuel, utilisez noflash au lieu de flash.
- emacs : Mettez "(setq visible-bell t)" dans votre ~/.emacs. Pour la désactiver : "(setq visible-bell nil)".
- less : Utilisez l'option "-q" sur la ligne de commande pour avoir un bip visuel, utilisez "-Q" pour n'avoir aucun bip. Pour avoir l'une ou l'autre de ces options par défaut, mettez-les dans la variable d'environnement "LESS".
- screen : Faites "Ctrl-A - Ctrl-G". Cela change le comportement de tous les écrans virtuels. Voyez la page du manuel sous "CUSTOMIZATION" pour fixer le comportement par défaut.

8 La face cachée du problème

La mauvaise nouvelle est que toutes les applications n'utilisent pas termcap ou terminfo. La plupart des petits programmes ont le caractère 'backslash-a' (l'alarme) dans le code source C. Le code "alarme" devient un caractère ASCII BEL dans les chaînes de caractère du code binaire. Les vraies applications ne sont en principe pas de ce type, mais méfiez-vous des débutants en C qui vous donnent leurs programmes. Les pires, je vous assure, sont les étudiants en informatique.

La seule manière de réduire ces applications au silence est de procéder à une "spekearectomy", ou d'utiliser les séquences d'échappement dues à Martin Mares.