Guide pratique du MIDI sous Linux

Version française de **The Linux MIDI-HOWTO**

Phil Kerr

<phil CHEZ plus24 POINT com>

Jean-Marc Legrand

<cjm POINT legrand CHEZ wanadoo POINT fr>

Adaptation française

Éric Madesclair

<eric TIRET m CHEZ wanadoo POINT fr>

Relecture de la version française

Jean-Philippe Guérard

<fevrier CHEZ tigreraye POINT org>

Relecture rapide

Jean-Philippe Guérard

<fevrier CHEZ tigreraye POINT org>

Préparation de la publication de la v.f.

Version : 1.2.fr.1.0

2004-04-04

Historique des versions		
Version 1.20.fr.0.9	2004-04-04	JML, ÉM, JPG
Première traduction française.		
Version 1.20	2002-05-24	PK
Version 1.10	2002-04-02	PK
Version 1.00	2002-01-25	PK

Résumé

Ce document présente, et explique comment utiliser le matériel et les logiciels permettant de jouer et de composer des fichier MIDI sous Linux.

Table des matières

Introduction

Ce document traite des points suivants :

- comment configurer votre interface MIDI ;
- comment configurer et utiliser des synthétiseurs logiciels ;
- comment jouer des fichiers MIDI ;
- comment utiliser un séquenceur ;
- comment contrôler du matériel MIDI externe ;
- logiciels MIDI basés sur la synthèse de sons ;
- exemples de code MIDI.

Droits d'utilisation (*Copyright of this document*)

Copyright © 2002 Phil Kerr.

This HOWTO is copyrighted 2002 Phil Kerr.

Copyright © 2002 Frank Barknecht pour la la section intitulée « Guide d'utilisation de séquenceurs MIDI avec des synthétiseurs virtuels ».

The "HOWTO Use MIDI Sequencers With Softsynths" is copyright 2002 Frank Barknecht.

Copyright © 2004 Jean-Marc Legrand, Éric Madesclair, Jean-Philippe Guérard pour la version française.

Ce guide pratique est distribué sous la licence de documentation libre GNU (GFDL). Vous devriez avoir reçu une copie de cette licence avec celui-ci. Si tel n'est pas le cas, vous pouvez vous la procurer en version originale anglophone à cette adresse :

This document is distributed under the terms of the GNU Free Documentation License. You should have received a copy along with it. If not, it is available from:

http://www.fsf.org/licenses/fdl.html

Il en existe une version française non-officielle à cette adresse :

http://cesarx.free.fr/gfdlf.html

Où se procurer ce guide ?

La dernière version française officielle de ce guide est disponible sur : http://www.traduc.org/docs/howto/lecture/MIDI-HOWTO.html.

La dernière version originale de ce guide pratique est publiée sur : http://www.midi-howto.com.

Remerciements

Ce guide est basé sur le petit guide du MIDI-SB (*MIDI-SB mini-HOWTO*) écrit par Hideki Saito. Tous mes remerciements pour sa contribution à la communauté Linux.

Ce guide contient désormais le « Guide d'utilisation des Séquenceurs MIDI avec des Synthétiseurs virtuels » écrit par Frank. Tous mes remerciements, Frank.

Une grande partie des exemples de code contenus dans ce guide provient de la liste de discussions LAD (Développeurs Audio Linux). Merci à eux pour m'avoir permis de les reprendre dans ce guide.

Avertissement (*Disclaimer*)

L'utilisation des informations de ce document est de votre responsabilité.

Use the information in this document at your own risk.

Je décline toute responsabilité quant au contenu de ce document.

I disavow any potential liability for the contents of this document.

L'utilisation des principes, exemples, et autres contenus de ce document sera entièrement de votre responsabilité.

Use of the concepts, examples, and/or other content of this document is entirely at your own risk.

Tous les droits d'auteur et de copyright sont la propriété de leurs détenteurs respectifs, sauf indication contraire.

All copyrights are owned by their owners, unless specifically noted otherwise.

Aucune remarque contenue dans ce document ne pourra être considérée comme un jugement de valeur contre telle ou telle marque.

Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark.

Le fait de citer une marque ou un produit particulier ne saurait être considérée comme une recommandation.

Naming of particular products or brands should not be seen as endorsements.

Il vous est fortement recommandé d'effectuer une sauvegarde de votre système avant toute installation d'importance et de le sauvegarder régulièrement au cours de votre travail.

You are strongly recommended to take a backup of your system before major installation and backups at regular intervals.

Histoire du MIDI

Le MIDI est un ensemble de protocoles matériels et logiciels permettant à des instruments de musique électroniques de communiquer entre eux. Le MIDI est apparu pour la première fois en 1982, et s'est imposé comme le standard de communication pour ce type d'instruments. Ce protocole fixe les paramètres physiques du matériel (connexions et interfaces matérielles), et permet de disposer d'un éventail précis de protocoles de communication pour l'échange de données musicales et rythmiques.

Avant le MIDI, il existait d'autres protocoles de connexion, tout d'abord analogiques, puis numériques. Le plus ancien, au alentours de 1974, permettait de transmettre des informations musicales sous forme de tension électrique via les câbles de liaison entre les différents claviers. Plus tard, en 1980-1981, Roland créa un protocole numérique de référence, le DCB. L'association des fabricants d'instruments MIDI (*MIDI Manufacturers Association* - MMA) a défini, à partir de leurs propres équipements, un protocole et un câblage standards pour des matériels compatibles, qui permettait à tous les matériels issus de fabricants adhérent à ce standard de se connecter et de communiquer entre eux.

À partir de 1985, les interfaces MIDI commencèrent à apparaître sur les PC personnels et peu après apparurent les premiers séquenceurs virtuels.

Configurer ses périphériques MIDI

Les périphériques MIDI sont soit intégrés à la carte son, soit inclus dans un matériel séparé. Les interfaces MIDI externes sont reliées soit au port série, soit à un port USB.

La première chose à faire — et la plus importante — est de vérifier la compatibilité de votre carte son !

Les adresses suivantes peuvent vous y aider :

http://www.alsa-project.org/alsa-doc/

http://www.4front-tech.com/osshw.html

La façon de configurer ses périphériques MIDI varie d'une distribution Linux à l'autre. Normalement, une carte son compatible sera configurée correctement à l'installation du système d'exploitation.

Le noyau Linux contient les pilotes OSS, et, depuis les versions 2.5, les pilotes ALSA. La plupart des distributions offrent un outil de configuration (essentiellement dédié aux cartes son). Toutefois, si vous utilisez le port MIDI d'une carte son, il devra être configuré manuellement. Sous RedHat, il faut utiliser sndconfig, sous SuSE : yast, et sous Mandrake : DrakConf.

Si aucun de ces outils n'est utilisable, ou si vous rencontrez des problèmes, suivez les étapes suivantes :

Est-ce que la commande **lsmod** montre des modules spécifiques au MIDI ? Voici ce que l'on peut obtenir sur un système basé sur OSS :

```
[root@beatbox]# lsmod
Module                  Size  Used by
lockd                  32208   1  (autoclean)
sunrpc                 54640   1  (autoclean) [lockd]
autofs                  9456   2  (autoclean)
usb-ohci               12624   0  (unused)
usbcore                43632   1  [usb-ohci]
hisax                 470096   0  (autoclean) (unused)
isdn                  104208   0  (autoclean) [hisax]
slhc                    4544   0  (autoclean) [isdn]
eepro100               16144   1  (autoclean)

#---- modules de cartes son
    opl3                   11376   2
    mad16                   7968   1
    ad1848                 16848   1  [mad16]
    sb                     34752   1  [mad16]
    uart401                 6384   1  [mad16 sb]
    sound                  58368   0  [opl3 mad16 ad1848 sb uart401]

soundlow                 464   0  [sound]
soundcore               2800   6  [sb sound]
nls_cp437               3952   2  (autoclean)
vfat                    9408   1  (autoclean)
fat                    30432   1  (autoclean) [vfat]
ide-scsi                7664   0
```

Recherchez les modules mpu401, olp3, uart401 et oss.

Si vous utilisez une interface USB, n'oubliez pas de vérifier que les modules USB correspondants apparaissent bien.

Pour vérifier la configuration, faites un **cat** du fichier `sndstat` :

```
[root@beatbox]# cat /dev/sndstat
OSS/Free:3.8s2++-971130
Load type: Driver loaded as a module
Kernel: Linux mega 2.2.17-21mdk #1 Thu Oct 5 13:16:08 CEST 2000 i686
Config options: 0

Installed drivers:

Card config:

Audio devices:
```

```
0: MAD16 WSS (82C930) (DUPLEX)

Synth devices:
0: Yamaha OPL3

Midi devices:
0: Mad16/Mozart

Timers:
0: System clock

Mixers:
0: MAD16 WSS (82C930)
```

Nous voyons ici que le périphérique MIDI est mad16, ce qui correspond au lsmod effectué plus haut.

Si rien n'apparaît concernant le MIDI, vérifiez le contenu de votre fichier /etc/modules.conf.

```
[root@beatbox]# cat /etc/modules.conf
alias net-pf-4 ipx
pre-install pcmcia_core /etc/rc.d/init.d/pcmcia start
alias usb-interface usb-ohci
alias parport_lowlevel parport_pc
alias block-major-11 scsi_hostadapter
pre-install plip modprobe parport_pc ; echo 7 > /proc/parport/0/irq
alias scsi_hostadapter ide-scsi
alias eth0 eepro100
alias eth1 hisax

#---- Carte Son
    alias sound-slot-0 mad16
    options sound dmabuf=1
    alias midi opl3
    options opl3 io=0x388
    options sb support=1
    options mad16 io=0x530 irq=5 dma=0 dma16=1 mpu_io=0x300 mpu_irq=7 joystick=1
```

Voici les données de sortie à vérifier dans le module /proc/modules pour voir si les modules MIDI sont bien chargés dans le noyau.

```
[root@mega /proc]# cat modules
0000-001f : dma1
0020-003f : pic1
0040-005f : timer
0060-006f : keyboard
0070-007f : rtc
0080-008f : dma page reg
00a0-00bf : pic2
00c0-00df : dma2
00f0-00ff : fpu
0170-0177 : ide1
01f0-01f7 : ide0
02f8-02ff : serial(auto)

#---- Périphérique MIDI
    0300-0303 : MPU-401 UART

0376-0376 : ide1
0388-038b : Yamaha OPL3
03c0-03df : vga+
03f6-03f6 : ide0
03f8-03ff : serial(auto)
0530-0533 : MAD16 WSS config
0534-0537 : MAD16 WSS
de00-de1f : Intel Speedo3 Ethernet
f000-f007 : ide0
f008-f00f : ide1
```

Vous devriez trouver quelque chose ressemblant à cela. Si tel n'est pas le cas, vous devrez installer manuellement les pilotes MIDI.

Si vous êtes amenés à utiliser les pilotes ALSA 0.5x, — ce que je vous déconseille — je vous suggère de lire auparavant le Petit guide du son avec ALSA (*Alsa-sound-mini-HOWTO*) de Valentijn Sessink que vous trouverez au lien suivant :

http://www.traduc.org/docs/howto/lecture/Alsa-sound.html

Je vous recommande vivement d'utiliser des versions d'ALSA supérieures à la version 0.9. Pour des pilotes ALSA plus récents que les versions 0.9x, je vous conseille de lire le Guide ALSA (*Alsa HOWTO*) de Madhu Maddy.

http://www.alsa-project.org/alsa-doc/alsa-howto/

Installation rapide de ALSA 0.9

Vous trouverez ci-dessous une méthode d'installation rapide des pilotes et des bibliothèques ALSA 9.0, ainsi que les configurations les plus répandues dans les applications MIDI.

```
[root@beatbox] # tar jxvf alsa-driver....tar.bz2
[root@beatbox] # cd alsa-driver.....
```

```
[root@beatbox] # ./configure
messages - no errors
[root@beatbox] # make
messages - no errors
[root@beatbox] # make install
messages - no errors
[root@beatbox] # ./snddevices
```

À ce stade, vous devez éditer le fichier /etc/modules.conf, ou, pour certaines distributions, le fichier ALSA de votre répertoire de modules. Il se pourrait que vous rencontriez des indications pour d'autres matériels, non MIDI : soyez prudent lors de l'édition du fichier.

Si il existe déjà une ancienne configuration ALSA ou OSS dans ce fichier, vous devrez l'effacer ou, mieux encore, la commenter.

Vous trouverez ci-dessous un exemple typique de fichier `modules.conf`, décrivant une configuration ALSA sous OSS.

```
alias char-major-116 snd
alias char-major-14 soundcore

alias snd-card-0 (carte son/MIDI)
alias sound-slot-0 snd-card-0

alias sound-service-0-0 snd-mixer-oss
alias sound-service-0-1 snd-seq-oss
alias sound-service-0-3 snd-pcm-oss
alias sound-service-0-12 snd-pcm-oss
```

Remplacez l'entrée `(carte son/MIDI)` par la référence correcte à votre carte son. Vous trouverez normalement les informations nécessaires sur le site web ALSA.

Une fois les pilotes ALSA installés, vous devrez installer les fichiers de bibliothèques d'en-têtes nécessaires au fonctionnement des programmes basés sur ALSA. C'est ce que contient le paquet alsa-libs.

Assurez-vous tout d'abord que votre paquet alsa-libs correspond bien à vos pilotes ALSA !

```
[root@beatbox] # tar jxvf alsa-libs....tar.bz2
[root@beatbox] # cd alsa-libs.....
[root@beatbox] # ./configure

messages - no errors
[root@beatbox] # make
messages - no errors
[root@beatbox] # make install
```

Votre système devrait alors être configuré ! :o)

Vous pouvez le vérifier à l'aide d'un simple programme en C : si vous réussissez à le compiler et l'exécuter, c'est que votre système fonctionne bien.

```
// Compilez ce programme test ainsi : gcc alsatest.c -o alsatest -lasound

#include <stdio.h>
#include <alsa/asoundlib.h>

int main (int argc, char *argv[])
{
  snd_seq_t *seq_handle;

  if (snd_seq_open(&seq_handle, "hw", SND_SEQ_OPEN_DUPLEX, 0) < 0) {
    fprintf(stderr, "Error opening ALSA sequencer.\n");
    exit(1);
  }

printf("The ALSA libraries are installed.\n");
return 0;
}
```

Les problèmes de latence

Le MIDI est un protocole temps-réel et les réglages de latence peuvent poser de sérieux problèmes.

Désormais, plusieurs développeurs travaillent sur l'amélioration des temps de latence, et les progrès réalisés sur le noyau font aujourd'hui de Linux une plateforme tout à fait efficace pour le MIDI.

Même si les distributions Linux fonctionnent correctement sur ce point, les utilisateurs avancés devront installer des correctifs noyau pour diminuer les temps de latence. Vous trouverez plus d'informations sur les sites suivants :

http://www.gardena.net/benno/linux/audio/

http://www.linuxdj.com/audio/lad/resourceslatency.php3

Vous trouverez en outre le Petit guide de réduction des temps de latence (*Low Latency Mini Howto* — en anglais) à l'adresse suivante :

http://web.archive.org/web/20021211025416/http://www.boosthardware.com/LAU/guide/Low_latency-Mini-HOWTO.ht

Les logiciels

L'intérêt pour des solutions MIDI basées sur Linux étant croissant, la liste suivante ne sera probablement pas exhaustive, mais vous donnera un choix représentatif d'applications MIDI.

Si vous êtes développeur ou utilisateur d'autres applications MIDI non listées, merci de m'en avertir par courrier électronique.

Les pilotes

Les pilotes ALSA

ALSA (l'Architecture Sonore Linux Avancée — *Advanced Linux Sound Architecture*) est constituée de plusieurs éléments. Le premier est un pilote son totalement modulaire, compatible avec le chargement automatique des modules, l'utilisation de devfs, l'autoconfiguration isapnp, et permettant un accès complet aux composants audio analogique, audio numérique, contrôles, mixage, synthétiseur, DSP, MIDI et à la temporisation du matériel MIDI. Il comprend aussi un séquenceur noyau complet, une couche de compatibilité complète avec les applications libres basées sur OSS, une bibliothèque C orientée objet permettant de traiter et améliorer les fonctionnalités des pilotes ALSA pour certaines applications (client/serveur, modules d'extension, le partage et le mixage du son PCM, mesure du son PCM, et cætera), une interface de configuration du pilote, ainsi que quelques autres outils simples pour la configuration et la maintenance.

http://www.alsa-project.org/

OSS

OSS, le Système audio ouvert, fournit des pilotes de cartes son pour Linux et FreeBSD. Ces pilotes sont compatibles avec le son numérique, le MIDI, les synthétiseurs virtuels et les modules de mixage intégrés aux cartes son. Ces pilotes sonores sont conformes aux spécifications de la bibliothèque de programmation Système audio ouvert (OSS — *Open Sound System*). Le système OSS est muni d'une interface graphique très conviviale, qui rend très simple l'installation des pilotes ainsi que la configuration de la carte son. Ces pilotes sont compatibles avec plus de 200 cartes son de marque et permettent une détection automatique de la carte son, l'autoconfiguration (*Plug-n-Play*), l'utilisation des cartes son PCI et la capacité de fonctionnement audio bidirectionnel.

http://www.opensound.com/

Notemidi

Notemidi est un pilote de périphérique pour la sortie MIDI via le port série RS-232 sur les PC portables. Notemidi peut être utilisé avec l'interface MIDIator MS-124W, les modules de son Roland Sound Canvas, ou les modules de son de la série Yamaha MU-x.

http://www.michaelminn.com/linux/notemidi

Les lecteurs MIDI

KMid

KMid est un lecteur MIDI basé sur KDE (et donc X11) pour Linux et FreeBSD. Il permet le défilement des textes de fichiers karaoke, et change la couleur du texte au fur et à mesure qu'il est joué, ce qui permet de le suivre facilement. KMid utilise `/dev/sequencer` comme périphérique de sortie compatible avec les synthétiseurs externes, ainsi qu'avec les cartes AWE, FM et GUS.

http://perso.wanadoo.es/antlarr/kmid.html

Pmidi

Pmidi est un programme en lignes de commande qui permet de jouer des fichiers MIDI sur le séquenceur ALSA.

http://www.parabola.demon.co.uk/alsa/pmidi.html

TiMidity++

TiMidity est un convertisseur MIDI vers WAVE qui utilise des bibliothèques de timbres dans un format compatible Gravis Ultrasound(*) afin de produire des données audionumériques à partir de fichiers General MIDI. Les données audio peuvent alors être jouées par n'importe quel périphérique son ou stockées sur disque. Sur une machine puissante, les données peuvent être jouées en temps réel.

http://timidity.sourceforge.net/

Les séquenceurs

Brahms

Brahms est à la fois un séquenceur et un éditeur de partitions, muni de plusieurs outils d'édition comme Score-, Pianoroll-, Drum-, et Mastertrack Editor. Pour les développeurs C++, il est aisé de créer d'autres outils d'édition à partir d'un module d'édition générique. Ce programme possède aussi une fonction d'import et export MIDI. Combiné avec aRts-0.3.4, il est possible de jouer des fichiers WAVE et d'envoyer simultanément au synthétiseur virtuel des évènements MIDI.

Ce logiciel était autrefois connu sous l'appellation Koobase.

http://brahms.sourceforge.net/

Anthem

Anthem est un séquenceur MIDI libre évolué. Anthem vous permet d'enregistrer, d'éditer et de jouer de la musique, en utilisant une technologie musicale orientée objet très sophistiquée et très appréciée.

http://anthem.sourceforge.net/

Jazz++

JAZZ++ est un séquenceur audio et MIDI complet, sous Linux et Windows.

http://www.jazzware.com/cgi-bin/Zope.cgi/jazzware/

UltiMusE-LX

UltiMusE-LX (*the Ultimate Music Editor*) est un logiciel de composition musicale. Non, non : il ne compose pas à votre place ! C'est un logiciel musical graphique. Vous composez votre partition à l'écran à l'aide de la souris et/ou du clavier, jusqu'à 16 parties ou voix, avec jusqu'à 7 portées. Il permet d'utiliser quasiment toutes les notations standards, ainsi que les outils MIDI de changement d'instrument, d'événement, et de gestion en temps réel.

http://hometown.aol.com/knudsenmj/myhomepage/umuselx.htm

Melys

Melys est une application séquenceur MIDI basé sur ALSA. Melys utilise le séquenceur ALSA et le combine avec les bibliothèques GNOME pour produire un séquenceur puissant et facile à utiliser.

http://www.parabola.demon.co.uk/melys/

MidiMountain Sequencer

MidiMountain est un séquenceur qui permet d'éditer des fichiers MIDI standard. La facilité d'utilisation de son interface devrait aider les débutants à éditer et créer des séquences MIDI. MidiMountain est conçu pour permettre de manipuler toutes les définitions connues des fichiers MIDI standards et du protocole de transfert MIDI, du simple rouleau de piano mécanique à la manipulation des messages exclusifs systèmes binaires.

http://www.midimountain.com/

MusE

MusE est un séquenceur MIDI pour Linux, basé sur Qt 2.1, avec fonctions d'édition et d'enregistrement. Pendant que le séquenceur joue, vous pouvez éditer des évènements MIDI en temps réel à l'aide du rouleau de piano mécanique ou de l'éditeur de partition. Les évènements MIDI enregistrés peuvent alors être traités à part avec le menu arrangeur.

http://muse.seh.de/

Rosegarden

Rosegarden est un séquenceur MIDI intégré et un éditeur de partition.

http://www.rosegardenmusic.com/

Consoles de mixage MIDI

tektracker

ttrk (tektracker) est un séquenceur MIDI muni d'une interface de type console de mixage. Il est fait pour être utilisé en temps réel, notamment grâce à ses boutons de volume par piste et création de boucles. ttrk permet d'envoyer et de se synchroniser avec les pulsations d'horloge MIDI.

http://vektor.ca/audio/ttrk/

ShakeTracker

ShakeTracker est un séquenceur MIDI muni d'une interface de type console de mixage. Il fonctionne correctement et permet d'utiliser quasiment tous les effets de mixage habituels. Ceux d'entre vous qui auront utilisé Impulse Tracker auparavant se sentiront chez eux, et pour les nouveaux utilisateurs, ils bénéficieront d'un système d'aide en ligne très complet et très simple d'utilisation. La plupart des commandes et raccourcis ressemblent à leur équivalent sur une table de mixage.

http://reduz.com.ar/shaketracker/

Éditeurs de rythmes

DrumPatterns

DrumPatterns est un générateur de rythmes, libre et orienté internet, dont l'objectif est d'apprendre à créer des motifs rythmiques. Il est capable de vous apprendre les bases rythmiques, aussi bien que les rythmes les plus compliqués. Il peut éditer des graphismes, des partitions, des évènements MIDI, et inclut des heures d'échantillons d'exemples.

http://www.linux-france.org/prj/drumpatterns/index-fr.html

Éditeurs de timbres

JSynthLib

JSynthLib est un éditeur et conservateur libre de timbres, écrit en langage Java. Ce projet a pour but d'offrir une compatibilité avec tous les synthétiseurs existants, en fournissant aux utilisateurs des méthodes et des documentations qui leur permette de développer des pilotes et des éditeurs pour les synthétiseurs non compatibles, et de les diffuser via le projet.

Synthétiseurs virtuels

Spiral Synth

Spiral Synth est un synthétiseur polyphonique analogique virtuel. Il est capable de réutiliser des sons fabriqués par des synthétiseurs analogiques matériels, comme les bruits utilisés dans la musique électronique. Vous pouvez aussi l'utiliser pour créer les sons les plus bizarres. Il est compatible MIDI et il utilise la sortie sonore standard du système OSS libre (`/dev/dsp`).

http://www.pawfal.org/SpiralSynth/

UltraMaster Juno-6

UltraMaster Juno-6 est une reproduction virtuelle fidèle du synthétiseur polyphonique Roland Juno-6. Il intègre la synthèse d'ondes ULTRANAMOG en temps-réel et en 64 bit, permet de produire des accords et des chœurs dans le style des années 80. Tous les paramètres peuvent être contrôlés en temps réel, soit via une interface graphique, soit via des actions sur des contrôleurs MIDI externes.

http://www.ultramaster.com/juno6/index.html

Pure-Data

Pure-Data est un logiciel temps réel pour l'exécution en direct de pièces musicales et multimédia. Il est développé par Miller Puckette , sans doute en collaboration avec d'autres développeurs. Ce logiciel n'est pas encore abouti, mais peut tout à fait être utilisé dans le cadre de projets complexes. Il a été adapté sur Linux, IRIX, et Windows.

http://www.pure-data.org/

Csound

Csound est un synthétiseur virtuel qui ne souffre pas des limitations des autres synthétiseurs matériels ou logiciels de ce type. Le nombre d'oscillateurs ou de filtres utilisables est illimité. Csound est aussi complètement modulable, de façon à ce que chaque fonction soit utilisable de plusieurs façons différentes.

http://www.csound.org/.

Émulateur de Synthétiseur Bristol

Bristol est une collection d'émulateurs de synthétiseur. Il comprend Moog Mini, Moog Voyager, Hammond B3, Prophet 5, Juno 6, DX 7, ainsi que d'autres logiciels.

http://www.slabexchange.org/index.cgi?DOWNLOAD

Les modules d'extension (*plugins*)

xmms-midi

Il ajoute à x11amp la compatibilité avec les fichiers MIDI (via timidity). Une interface de mixage simplifiée est disponible par le menu de configuration.

http://ban.joh.cam.ac.uk/~cr212/xmms-midi/

Partitions musicales

Mup

Mup prend des données d'entrée de type texte pour les transformer en données Postcript imprimables. Les notations standards classiques et la notation en tablature sont toutes deux utilisables. MUP peut aussi produire des données de sortie de type MIDI.

http://www.arkkra.com/

Lilypond

LilyPond est un logiciel de composition musicale. Il permet de réaliser de très belles partitions, à partir de données d'entrée très détaillées. LilyPond fait partie du projet GNU.

http://www.lilypond.org/

Développement

sfront

Sfront convertit des flux de données audio MP4-SA (MPEG 4 Structured Audio) en programmes C très efficaces qui produisent de l'audio à l'exécution. MP4-SA est un format standard pour des algorithmes de son, combinant un langage de création de signaux audio (SAOL) avec un langage de notation musicale (SASL, et le traditionnel format de fichier MIDI). Sous Linux, sfront exploite le temps réel, les entrées-sorties audio à faible temps de latence, les entrées MIDI à partir de cartes son, ainsi que les entrées MIDI via internet utilisant le RTP et le SIP. Un serveur SIP hébergé sur le campus de Berkeley organise des sessions de travail. Le site comprend aussi une aide en ligne à propos du MP4-SA.

http://www.cs.berkeley.edu/~lazzaro/sa/index.html

jMax

jMax est un système qui permet de dessiner des circuits de flux de données. On peut y entrer des données de type entier, symbole, liste, et cætera. C'est un système piloté par les événements. JMax est utilisé pour la création MIDI. Une autre partie de ce système, DSP, permet de faire circuler un signal en continu dans le circuit, ce qui est le plus pratique pour les sons PCM (i. e. : micros, fichiers son, et cætera). Ce système est extensible avec des bibliothèques partagées, permettant l'utilisation de différents types de gestion des données, de périphériques, d'interfaces, de matériels, et cætera. Les modules de traitement de données peuvent être traités comme des circuits et utilisés en tant que tel.

http://sourceforge.net/projects/jmax/

TSE3

TSE3 est un puissant séquenceur en mode texte, dont les sources sont publiques, écrit en C++. C'est un séquenceur en mode texte parce qu'il présente toutes les fonctionnalités d'un séquenceur mais sans interface graphique. Différentes applications séquenceur ou multimédia utilisent TSE3 derrière leur interface graphique.

http://TSE3.sourceforge.net/

KeyKit

KeyKit est un langage de programmation multi-tâches (inspiré de awk), exclusivement réservé aux opération MIDI algorithmiques et temps réel. Son interface graphique propose plusieurs douzaines d'outils pour expérimenter différents algorithmes musicaux, y compris un séquenceur multipistes et un éditeur de motifs rythmiques. L'interface et les outils sont complètement écrits dans le langage de KeyKit. Cela permet d'ajouter de nouveaux outils et nouvelles fonctionnalités à l'existant, tout en utilisant le système.

http://nosuch.com/keykit/

Le développement MIDI

Ceux qui souhaitent développer des applications MIDI ont souvent besoin de bons exemples pour démarrer.

Les exemples suivants proviennent de la liste de discussions LAD.

Exemple 1

Vous trouverez ci-dessous une fonction de séquencement écrite par Dr. Matthias Nagorni. D'autres exemples sont disponibles sur son site (cf la section Liens).

Vous le compilez de la façon suivante :

```
[phil@beatbox] $ gcc seqdemo.c -o seqdemo -lasound
```

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <alsa/asoundlib.h>

snd_seq_t *open_seq();
void midi_action(snd_seq_t *seq_handle);

snd_seq_t *open_seq() {

  snd_seq_t *seq_handle;
  int portid;

  if (snd_seq_open(&seq_handle, "hw", SND_SEQ_OPEN_DUPLEX, 0) < 0) {
    fprintf(stderr, "Error opening ALSA sequencer.\n");
    exit(1);
  }
  snd_seq_set_client_name(seq_handle, "ALSA Sequencer Demo");
  if ((portid = snd_seq_create_simple_port(seq_handle, "ALSA Sequencer Demo",
            SND_SEQ_PORT_CAP_WRITE|SND_SEQ_PORT_CAP_SUBS_WRITE,
            SND_SEQ_PORT_TYPE_APPLICATION)) < 0) {
    fprintf(stderr, "Error creating sequencer port.\n");
    exit(1);
  }
  return(seq_handle);
}

void midi_action(snd_seq_t *seq_handle) {

  snd_seq_event_t *ev;

  do {
    snd_seq_event_input(seq_handle, &ev);
    switch (ev->type) {
      case SND_SEQ_EVENT_CONTROLLER:
        fprintf(stderr, "Control event on Channel %2d: %5d       \r",
                ev->data.control.channel, ev->data.control.value);
        break;
      case SND_SEQ_EVENT_PITCHBEND:
        fprintf(stderr, "Pitchbender event on Channel %2d: %5d   \r",
                ev->data.control.channel, ev->data.control.value);
        break;
      case SND_SEQ_EVENT_NOTEON:
        fprintf(stderr, "Note On event on Channel %2d: %5d       \r",
                ev->data.control.channel, ev->data.note.note);
        break;
      case SND_SEQ_EVENT_NOTEOFF:
        fprintf(stderr, "Note Off event on Channel %2d: %5d      \r",
                ev->data.control.channel, ev->data.note.note);
        break;
    }
    snd_seq_free_event(ev);
  } while (snd_seq_event_input_pending(seq_handle, 0) > 0);
}

int main(int argc, char *argv[]) {

  snd_seq_t *seq_handle;
  int npfd;
  struct pollfd *pfd;

  seq_handle = open_seq();
  npfd = snd_seq_poll_descriptors_count(seq_handle, POLLIN);
  pfd = (struct pollfd *)alloca(npfd * sizeof(struct pollfd));
  snd_seq_poll_descriptors(seq_handle, pfd, npfd, POLLIN);
  while (1) {
    if (poll(pfd, npfd, 100000) > 0) {
      midi_action(seq_handle);
    }
  }
}
```

Exemple 2

Vous trouverez ci-dessous un « aiguilleur » MIDI basé sur ALSA 0.9 écrit par Nick Dowell.

```
/* Aiguilleur pour séquenceur ALSA.
   Aiguille les données d'entrée vers les sorties définies
   par le canal MIDI (à la demande de Nathaniel Virgo sur
   Linux-Audio-Dev ;o)). Est basé sur l'exemple de fonction
   séquenceur ALSA écrit par Dr. Matthias Nagorni.

   Nick Dowell <nixx CHEZ nixx POINT org POINT uk>    */

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
```

```
#include <alsa/asoundlib.h>

int
main()
{
  snd_seq_t *seq_handle;
  snd_seq_event_t *ev;
  int i;
  int portid;              /* port d'entrée (input) */
  int oportid[16];         /* ports de sortie (outputs) */
  int npfd;
  struct pollfd *pfd;
  char txt[20];

  if (snd_seq_open(&seq_handle, "hw", SND_SEQ_OPEN_DUPLEX, 0) < 0) {
    fprintf(stderr, "Error opening ALSA sequencer.\n");
    exit(1);
  }

  snd_seq_set_client_name(seq_handle, "MIDI Redirect");

  /* open one input port */
  if ((portid = snd_seq_create_simple_port
       (seq_handle, "Input",
        SND_SEQ_PORT_CAP_WRITE | SND_SEQ_PORT_CAP_SUBS_WRITE,
        SND_SEQ_PORT_TYPE_APPLICATION)) < 0) {
    fprintf(stderr, "fatal error: could not open input port.\n");
    exit(1);
  }
  /* open 16 output ports for the MIDI channels */
  for (i=0; i<16; i++){
    sprintf( txt, "MIDI Channel %d", i );
    if ((oportid[i] = snd_seq_create_simple_port
         (seq_handle, txt,
          SND_SEQ_PORT_CAP_READ | SND_SEQ_PORT_CAP_SUBS_READ,
          SND_SEQ_PORT_TYPE_APPLICATION)) < 0) {
      fprintf(stderr, "fatal error: could not open output port.\n");
      exit(1);
    }
  }

  npfd = snd_seq_poll_descriptors_count(seq_handle, POLLIN);
  pfd = (struct pollfd *)alloca(npfd * sizeof(struct pollfd));
  snd_seq_poll_descriptors(seq_handle, pfd, npfd, POLLIN);

  while (1)  /* main loop */
    if (poll(pfd, npfd, 1000000) > 0){
      do {
        snd_seq_event_input(seq_handle, &ev);
        snd_seq_ev_set_source( ev, oportid[ev->data.control.channel] );
        snd_seq_ev_set_subs( ev );
        snd_seq_ev_set_direct( ev );
        snd_seq_event_output_direct( seq_handle, ev );
        snd_seq_free_event(ev);
      } while (snd_seq_event_input_pending(seq_handle, 0) > 0);
    }
  return 0;
}
```

Exemple 3

Vous trouverez ci-dessous un exemple d'écriture de données dans l'interface OSS `/dev/midi` par Craig Stuart Sapp.

Vous pourrez trouver plus d'exemples sur son site (cf la section Liens).

```
//
// Programmeur:            Craig Stuart Sapp
// Adresse électronique :  craig CHEZ ccrma POINT standford POINT edu
// Date de Création :      Lundi 21 décembre 1998 18:00:42
// Dernière modification : Lundi 21 décembre 1998 18:00:42
// Nom de fichier :      ...linuxmidi/output/method1.c
// Langage :             C
// $Smake: gcc -O -o devmidiout devmidiout.c && strip devmidiout
//

#include <linux/soundcard.h>
#include <unistd.h>
#include <fcntl.h>
#include <stdio.h>

int main(void) {
   char* device =  "/dev/midi" ;
   unsigned char data[3] = {0x90, 60, 127};

   // étape 1: ouvre le module OSS pour écriture
   int fd = open(device, O_WRONLY, 0);
   if (fd < 0) {
      printf("Error: cannot open %s\n", device);
      exit(1);
   }

   // étape 2: écrit les données MIDI dans le module OSS
   write(fd, data, sizeof(data));

   // étape 3: (en option) ferme le module OSS
   close(fd);

   return 0;
}
```

Guide d'utilisation de séquenceurs MIDI avec des synthétiseurs virtuels

Version française du guide pratique *HOWTO Use MIDI Sequencers With Softsynths*

Frank BARKNECHT

<barknech CHEZ ph TIRET cip POINT uni TIRET koeln POINT de>

http://linux-sound.org/quick-toots/4-sequencers_and_softsynths/quick-toot-midisynth_howto.html

Ce guide décrit les réglages nécessaires pour contrôler un synthétiseur MIDI virtuel à partir d'un séquenceur MIDI et via une connexion MIDI virtuelle sous ALSA 0.9. Ce document peut être librement traduit et diffusé. Il est soumis à la licence de documentation libre GNU.

Introduction

Les synthétiseurs virtuels, tels que Csound, Pure Data, jMax ou Spiral Synth Modular offrent des possibilités quasi infinies pour expérimenter les sons les plus usuels ou les plus inhabituels. Ils peuvent en outre remplacer les synthétiseurs matériels, relativement chers, ou ceux, souvent de mauvaise qualité, qui équipent les cartes son, si tant est qu'ils soient compatibles avec Linux. En contrepartie, composer sur ces synthétiseurs virtuels peut s'avérer fastidieux, comme sur Csound où l'on doit aligner des pages entières de chiffres pour composer, ce qui n'est pas l'idéal pour faire de la musique.

Les logiciels séquenceurs MIDI virtuels permettent de rendre la tâche plus agréable. Ils sont munis d'une interface permettant d'insérer des notes et des évènements MIDI de façon pratique : soit par l'intermédiaire d'une vraie partition, soit par un rouleau de piano mécanique ou via une liste d'évènements MIDI, suivant vos préférences. D'autres types de séquenceurs virtuels permettent d'entrer les notes en utilisant des morceaux de pistes, comme on le faisait dans le temps en direct. Dernier point, non des moindres, certains séquenceurs MIDI vous permettent d'enregistrer vos propres musiques par l'intermédiaire d'un matériel externe (clavier ou autre), ce qui se révèle, pour certains musiciens, le moyen le plus naturel de faire de la musique.

D'habitude, les séquenceurs MIDI envoient leurs données vers des périphériques MIDI qui à leur tour vont réorienter les données vers l'extérieur, à savoir vers des synthétiseurs et des échantillonneurs matériels. Grâce aux outils MIDI virtuels, on peut garder au sein du PC toutes les données MIDI et les faire contrôler par un logiciel tournant sur la même machine. Ce guide décrit tout ce qui est nécessaire pour en arriver là.

paramétrage du matériel

Pour le paramétrage, nous utiliserons les bibliothèques et les pilotes d'ALSA, ce qui doit être le cas si l'on veut utiliser sérieusement le son et le MIDI sous Linux. À partir de maintenant, nous supposerons que vous utilisez la version 0.9.0 d'ALSA. Toutefois, le module MIDI utilisé est aussi présent dans la version 0.5.x, ce qui rend les explications à venir tout aussi valables. Sous les architectures son OSS/Free (avec les modules de son trouvés dans les noyaux antérieurs à la version 2.5.x) et OSS/Linux, le module v_midi peut être utilisé, mais cela dépasse le cadre de ce guide.

Pour utiliser la carte MIDI virtuelle d'ALSA, le module snd-card-virmidi doit être présent. Avec la plupart des versions récentes d'ALSA (ainsi que dans la version 2.5.x de développement de noyau), le nom de ce module a été renommé « snd-virmidi ». Assurez-vous que vous avez bien pris en compte ce module : il pourrait vous manquer lors de la configuration d'ALSA, si vous n'avez pris en compte que les modules propres à votre carte son.

Vous devez charger le module virmidi pour activer les ports MIDI. Vous pouvez les tester avec :

```
$ modprobe snd-virmidi snd_index=1
```

où snd_index doit être adapté au premier index de carte libre (= 1 si vous disposez uniquement d'une carte qui a déjà l'index 0). Cela dit, il est plus pratique d'adapter la configuration de vos modules pour qu'ils soient disponible chaque fois que vous les utilisez. Pour cela, il faut ajouter à la section ALSA du fichier `/etc/modules.conf` (le chemin dépend de votre distribution) les lignes suivantes :

```
# Configure les supports pour OSS /dev/sequencer et
# /dev/music (aussi appelé /dev/sequencer2)
#
# (Takashi Iwai nous indique qu'il n'est pas nécessaire
# de créer des alias pour des cartes au-delà de la première
```

```
# carte, i.e., la carte 0)
#
alias sound-service-0-1 snd-seq-oss
alias sound-service-0-8 snd-seq-oss

# Configure la carte 1 (deuxième carte) en tant que carte MIDI virtuelle
alias sound-slot-1 snd-card-1
alias snd-card-1 snd-virmidi
```

Désormais, votre carte MIDI virtuelle est configurée en tant que carte d'index 1, ce qui suppose que vous possédez une carte son réelle (ce qui peut s'avérer très utile). Si vous possédez comme moi une seconde carte son réelle, modifiez la configuration ci-dessus comme suit :

```
# Configure la carte d'index 2 (troisième carte) en tant que
# carte MIDI virtuelle
alias sound-slot-2 snd-card-2
alias snd-card-2 snd-virmidi
```

Si vous possédez plus de cartes son que cela, vous saurez sûrement vous adapter.

Il se peut qu'il soit nécessaire de redémarrer le système ALSA, après quoi vous devriez voir indiquée la carte son virtuelle dans le fichier /proc/asound/cards :

```
$ cat /proc/asound/cards
0 [card0          ]: ICE1712 - M Audio Audiophile 24/96
                     M Audio Audiophile 24/96 at 0xb800, irq 5
1 [card1          ]: EMU10K1 - Sound Blaster Live!
                     Sound Blaster Live! at 0xc800, irq 11
2 [card2          ]: VirMIDI - VirMIDI
                     Virtual MIDI Card 1
```

Dans cet exemple tiré de mon propre PC, ma carte VirMIDI est la troisième, d'index 2. Cette configuration se retrouve dans le fichier /proc/asound/devices (seul les paramètres MIDI sont affichés) :

```
$ cat /proc/asound/devices
 8: [0- 0]: raw MIDI
41: [1- 1]: raw MIDI
42: [1- 2]: raw MIDI
75: [2- 3]: raw MIDI
74: [2- 2]: raw MIDI
73: [2- 1]: raw MIDI
72: [2- 0]: raw MIDI
```

Les entrées commençant par « 2- » correspondent à ma carte MIDI virtuelle . Si vous ne possédez qu'une seule carte réelle, ces entrées commenceront par « 1- ».

Vous pouvez obtenir un affichage plus conviviale des paramètres avec l'outil aconnect. Il s'agit d'un outil propre à ALSA, dont vous aurez de toutes façons besoin. Pour obtenir la liste des périphériques MIDI utilisables en sortie, appelez-le avec l'option `-o` (ou `-lo`). Pour ceux utilisables en entrée, appelez-le avec l'option `-i`.

```
$ aconnect -o
[...]
client 80: 'Virtual Raw MIDI 2-0' [type=kernel]
    0 'VirMIDI 2-0     '
client 81: 'Virtual Raw MIDI 2-1' [type=kernel]
    0 'VirMIDI 2-1     '
client 82: 'Virtual Raw MIDI 2-2' [type=kernel]
    0 'VirMIDI 2-2     '
client 83: 'Virtual Raw MIDI 2-3' [type=kernel]
    0 'VirMIDI 2-3     '
$ aconnect -i
[...]
client 80: 'Virtual Raw MIDI 2-0' [type=kernel]
    0 'VirMIDI 2-0     '
client 81: 'Virtual Raw MIDI 2-1' [type=kernel]
    0 'VirMIDI 2-1     '
client 82: 'Virtual Raw MIDI 2-2' [type=kernel]
    0 'VirMIDI 2-2     '
client 83: 'Virtual Raw MIDI 2-3' [type=kernel]
    0 'VirMIDI 2-3     '
```

Les périphériques affichés correspondent aux périphériques MIDI compatibles OSS propres à ALSA tels que définis dans le répertoire /proc/asound/dev. Par exemple, le périphérique /proc/asound/dev/midiC2D0 est le premier de notre carte MIDI virtuelle indexée 2, nommée Virtual Raw MIDI 2-0 par l'outil aconnect. Sous Débian, ces périphériques sont aussi visibles dans le répertoire `/dev/snd/`, et sont aussi liés en interne aux anciens répertoires de périphériques `/dev/midiXX`. Pour m'assurer que je pouvais accéder aux ports MIDI d'ALSA à partir de `/dev/midiXX`, je les ai renommés ainsi :

```
$ ln -s /dev/snd/midiC2D0 /dev/midi20
$ ln -s /dev/snd/midiC2D1 /dev/midi21
[...]
```

mais cela n'est pas nécessaire : ne le faites surtout pas chez vous !

Maintenant que nous avons créé et configuré une carte MIDI virtuelle, nous pouvons l'utiliser dans nos applications comme tout autre périphériques MIDI. Il vous suffit d'indiquer la référence du périphérique MIDI, qu'il s'agisse d'une référence compatible OSS, comme `/dev/midi20` ou d'un port MIDI ALSA comme `80:0`, à l'endroit approprié dans le paramétrage de votre séquenceur ou synthétiseur préféré.

Canalisation des évènements MIDI

aconnect

Sans autre manipulation, il ne vous sera pas possible d'envoyer des évènements MIDI à partir de votre séquenceur vers le synthétiseur virtuel. Pour ce faire, nous devons connecter deux ports avec, vous l'aurez deviné, l'outil aconnect. Cet outil permet de connecter deux ports ou plus. Les fichier obtenus avec -i ou -o plus haut nous ont déjà indiqué les ports disponibles. Maintenant, avec une syntaxe très simple, ces ports peuvent être connectés ainsi :

```
$ aconnect port_d_envoi port_de_reception
$ aconnect 80:0 81:0
```

Cette manipulation permet d'aiguiller toutes les données MIDI envoyées sur le port `80:0` vers le port `80:1`. Dans notre configuration, cela signifie que toute donnée entrant sur le port `/dev/midi20` est réorientée vers le port `/dev/midi21`, sur lequel elle peut être lue (« reçue ») par une autre application.

Si votre carte MIDI virtuelle est configurée en tant que seconde carte (index 1), vous devez obtenir les ports suivants :

```
$ aconnect -lo
client 72: 'Virtual Raw MIDI 1-0' [type=kernel]
    0 'VirMIDI 1-0     '
client 73: 'Virtual Raw MIDI 1-1' [type=kernel]
    0 'VirMIDI 1-1     '
client 74: 'Virtual Raw MIDI 1-2' [type=kernel]
    0 'VirMIDI 1-2     '
client 75: 'Virtual Raw MIDI 1-3' [type=kernel]
    0 'VirMIDI 1-3
```

Ici, vous pouvez par exemple connecter le port `72:0` (`/dev/midi10`) au port `73:0` (`/dev/midi11`) en tapant :

```
$ aconnect 72:0 73:0
```

aconnect peut nous montrer ce qui a été crée avec ses options -lo et -li :

```
$ aconnect -lo
client 72: 'Virtual Raw MIDI 1-0' [type=kernel]
    0 'VirMIDI 1-0     '
        Connecting To: 73:0
client 73: 'Virtual Raw MIDI 1-1' [type=kernel]
    0 'VirMIDI 1-1     '
        Connected From: 72:0
client 74: 'Virtual Raw MIDI 1-2' [type=kernel]
    0 'VirMIDI 1-2     '
client 75: 'Virtual Raw MIDI 1-3' [type=kernel]
    0 'VirMIDI 1-3
```

Vous constatez que le « Virtual Raw MIDI 1-0 » est maintenant connecté au « Virtual Raw MIDI 1-1 ». Dès lors, en fonction de vos applications, vous pourrez lire les données MIDI envoyées à votre port « Virtual Raw MIDI 1-0 » à partir du port « Virtual Raw MIDI 1-1 », ou, en reprenant le vocabulaire OSS : tout ce qui est envoyé sur le port `/dev/midi10` est aiguillé sur le port `/dev/midi11` et peut être lu à partir de là.

Vous pouvez aussi connecter plus d'un port sur un autre. Si vous faites aconnect une deuxième fois comme ceci :

```
$ aconnect 72:0 73:0
$ aconnect 72:0 74:0
```

vous pouvez recevoir les mêmes données envoyées sur `/dev/midi10`, sur les ports `/dev/midi11` et `/dev/midi12`. Et bien entendu, vous risquez vraiment de surcharger votre système si vous créez encore plus de cartes MIDI virtuelles et que vous les connectez entre elles. Mais rien n'empêche de le faire.

Pour déconnecter tous les ports, faites :

```
$ aconnect -x
```

Pour ne déconnecter qu'un seul port, faites :

```
$ aconnect -d 72:0 74:0
```

Tableaux de raccordement MIDI graphiques

Le tableau de raccordement MIDI ALSA (*Alsa Patch Bay*) de Bob Ham est une interface graphique très pratique pour la configuration MIDI d'ALSA sous Linux. Son utilisation est très simple et intuitive : sur la gauche, vous disposez des ports MIDI capables d'envoyer des évènements MIDI, alors que sur la droite, vous avez les ports capables d'en recevoir. Si vous cliquez sur un port de gauche, cela le sélectionne pour être connecté au port de droite sur lequel vous cliquez ensuite. Pour un port connecté, il suffit de cliquer sur le port de droite pour le déconnecter : un outil précis facile à utiliser, qui aurait presque tendance à rendre inutile ce qui a été dit plus haut ! ;o)

aseqview

aseqview est un autre outil très utile pour gérer les flux MIDI. Il a été créé par Takashi Iwai, qui fait partie des développeurs d'ALSA. Vous pouvez le télécharger depuis la page web d'Iwai-sama http://members.tripod.de/iwai/alsa.html, mais il est souvent inclus dans les distributions. Cet outil graphique a été créé pour visualiser et modifier les évènements MIDI pendant qu'ils transitent par votre PC. Il peut aussi orienter les données MIDI vers d'autres ports, tout comme aconnect. Cela peut s'avérer utile lorsque vous devez utiliser le séquenceur OSS, qu'aconnect est parfois incapable d'utiliser. Si vous lancez aseqview sans aucune option, vous obtenez une jolie interface graphique et un nouveau port MIDI. Le port par défaut est nommé `128:0`, et il ressemble à cela :

```
client 128: 'MIDI Viewer' [type=user]
    0 'Viewer Port 0 '
```

Avec ce port, toutes les manipulations effectuées plus haut avec aconnect sont possibles. Si vous souhaitez seulement connecter le port aseqview à un autre port, aseqview peut le faire par lui-même avec l'option -d :

```
$ aseqview -d 73:0 &
```

Cela connecte le port `128:0` au port `73:0` dès le démarrage d'aseqview.

Il existe d'autre outils graphiques de connexion, avec des fonctions très similaires. Maarten de Boer a utilisé la plupart des sources aconnect d'origine pour écrire une interface graphique appelée « aconnectgui » à l'aide de l'outil FLTK. Vous pouvez la trouver à l'adresse http://www.iua.upf.es/~mdeboer/. C'est, à mon avis, l'interface la plus agréable à utiliser.

Personnellement, j'utilise kaconnect, sans doute parce son nom est court et donc plus rapide à taper. kaconnect a été développé par Dr. Matthias Nagorni de SuSE, au sein d'une série d'outils et de synthétiseurs virtuels pour ALSA, les kalsatools. Ne vous méprenez pas sur le « k » : ce logiciel n'a pas besoin de KDE. Il utilise directement les bibliothèques graphiques QT. kaconnect, et d'autres logiciels, sont disponibles sur http://www.suse.de/~mana/kalsatools.html.

Applications

Dans ce dernier chapitre, je veux vous montrer quelques exemples d'utilisation des connections virtuelles MIDI dans diverses applications. Mettons que je dispose d'une carte VirMIDI, troisième sur mon système, utilisant les ports MIDI ALSA `80:0` à `83:0`, ce qui correspond aux périphériques OSS MIDI : `/dev/midi20` à `/dev/midi23`, et aux périphériques ALSA MIDI : `/dev/snd/midiC2D0` à `/dev/snd/midiC2D3`. Le deux premiers d'entre eux ont été connectés de la façon suivante :

```
$ aconnect 80:0 81:0
```

Ce qui signifie que toutes les données envoyées au port `/dev/midi20` (ou port `80:0` ou `/dev/snd/midiC2D0`) peuvent être maintenant lues sur le port `/dev/midi21` (ou port `80:1` ou `/dev/snd/midiC2D1`).

Les séquenceurs

MusE

MusE est un séquenceur virtuel complet écrit par Werner Schweer, disponible à l'adresse : http://muse.seh.de.. Nous devons configurer le port MIDI virtuel comme port de sortie (output) dans le menu « Config->MIDI Ports ». Avec MusE, les ports sont désignés par leur dénomination ALSA, du type « VirMIDI X-X ».

Assurez-vous d'avoir bien sélectionné le bon port comme port de sortie pour le canal sur lequel vous souhaitez que le synthétiseur oriente et joue les données MIDI.

Pour des raisons que j'ignore, je n'ai pas pu utiliser « VirMIDI 2-0 » comme sortie avec MusE 0.4.9. Ce devrait être pourtant la bonne sortie pour recevoir les données sur le port « VirMIDI 2-1 », mais j'ai dû l'utiliser d'une autre façon. Je ne sais pas pourquoi. On peut aussi utiliser les ports `midi02` ou `midi2`.

ttrk

ttrk, développé par Billy Biggs, est un petit séquenceur très simple muni d'une interface de mixage. Il peut sortir toutes ses données MIDI vers n'importe lequel des ports configurés dans le fichier $HOME/.ttrkrc.

Shaketracker

Le Shaketracker de Juan Linietsky reprend une interface de mixage MIDI semblable à celle de ttrk, avec toutefois une transposition complète des effets classiques de mixage sur les données MIDI. Malheureusement, il utilise comme sortie MIDI le séquenceur d'OSS (`/dev/sequencer`), et non les périphériques MIDI natifs (`raw MIDI`). Je n'ai pas pu le faire fonctionner avec aconnect. Toutefois, il possède un environnement qui inclut aseqview. Si l'on démarre aseqview avant Shaketracker, le gestionnaire de piste reconnaîtra et utilisera les ports de aseqview. Il suffit de le sélectionner dans le menu « User Devices » de Shaketracker, où il figure sous son nom ALSA « Viewer Port 0 ».

Il peut être pratique de donner à ce périphérique utilisateur un nom plus explicite que « Null Output ».

Si nous démarrons aseqview sans option, nous aurons besoin de connecter le port aseqview avec le port du synthétiseur virtuel. Mais, comme dit précédemment, nous pourrions aussi démarrer directement aseqview avec un port de destination. N'oubliez pas d'utiliser le menu User Device pour chacune des pistes qui doit aller vers le synthétiseur virtuel. Je fais toujours tourner Shaketracker avec un script qui démarre aseqview, attend la création des ports, et démarre enfin Shaketracker:

```
#!/bin/sh
aseqview -d 81:0 &
# sleep 2 seconds to let aseqview do its work:
sleep 2
shaketracker
```

Synthétiseurs virtuels

Pure Data

Miller Puckette est le génie qui se cache derrière Pure Data, un synthétiseur logiciel libre et un environnement de développement multimédia. Pure Data est issu de MAX et est à la base de l'extension MAX MSP. Pure Data peut aussi utiliser des périphériques MIDI natifs pour lire les évènements MIDI. Il faut, pour cela, indiquer quels périphériques utiliser avec l'option `-midiindev numéro_de_périphérique`. Cependant, la manière d'indiquer quel périphérique utiliser est assez déroutante. Voici la façon de faire : pour utiliser `/dev/midi0`, démarrez Pure Data avec `-midiindev 1`, pour utiliser `/dev/midi1` démarrez-le avec `-midiindev 2` et ainsi de suite. Vous devez donc indiquer le numéro de port + 1. Un autre exemple : pour `/dev/midi21` démarrez Pure Data avec l'option `-midiindev 22`.

Pure Data offre aussi un guide d'aide : « Test audio and MIDI », ce qui peut vraiment aider pour détecter le bon périphérique MIDI.

Csound

Csound est l'ancêtre de quasiment tous les synthétiseurs virtuels actuels, et il a contribué à l'évolution du MIDI. Lancer **`csound --help`** vous indiquera comment configurer le port d'entrée MIDI. Faites `-M nom_du_port` ou `--midieventdev=nom_du_port` et vous pourrez lire les données MIDI en temps réel.

Ainsi, dans notre exemple, nous lancerons Csound de la façon suivante :

```
$ csound -M /dev/midi22 -o devaudio midi.csd
```

Conclusion

Dorénavant, vous devriez pouvoir vous débrouiller avec tous les synthétiseurs virtuels pour orchestrer de la musique créée et jouée par un séquenceur MIDI. Bien sûr, des outils comme aconnect et aseqview ne sont pas nécessaires pour utiliser un séquenceur virtuel. Vous pouvez tout aussi bien rediriger les flux de données MIDI qui arrivent sur votre PC à partir d'un séquenceur externe ou d'un clavier MIDI directement vers le synthétiseur virtuel sans passer par l'étape du séquenceur MIDI. Vous devez simplement « aconnecter » le matériel MIDI externe à votre synthétiseur virtuel ou sur le synthétiseur embarqué sur votre carte son. Ou faire l'inverse : Pure Data, Csound ou des environnements comme KeyKit vous permettent de créer des évènements MIDI de façon algorithmique, ce qui est presque impossible avec des logiciel de type Cubase. Avec aconnect, vous pouvez aiguiller les évènements MIDI vers n'importe quel logiciel capable de les jouer.

Remerciements

L'auteur souhaite remercier Takashi Iwai pour son aide technique lors de l'élaboration de ce document et pour ses précieux conseils en Japonais. Merci, Takashi !

Liens Utiles

Vous trouverez ci-dessous une liste de liens vers des sites traitant du MIDI sous Linux.

http://www.linux-sound.org/ Logiciels audio et MIDI sous Linux. Une mine de renseignements sur le MIDI et l'audio.

http://www.linuxdj.com/audio/lad/ La liste de discussions des développeurs Audio sous Linux. Pour toute discussion sur le développement MIDI.

http://www.linuxdj.com/audio/quality Le guide Linux de la qualité audio. Un bon chapitre sur les cartes MIDI, essentiel pour créer son studio Linux.

http://www.alsa-project.org/ Le projet ALSA — Pilotes audio et MIDI.

http://www.4front-tech.com/ OSS — Pilotes audio et MIDI.

http://www.gardena.net/benno/linux/audio/ Correctif de minimisation des temps de latence.

http://ccrma-www.stanford.edu/~craig/articles/linuxmidi/ Introduction à la programmation MIDI sous Linux. Craig Stuart Sapp décrit les outils de base pour écrire du MIDI en C ou C++ pour des pilotes OSS.

http://www.suse.de/~mana Excellente collection de programmes C pour ALSA 0.9 MIDI et PCM, par Dr. Matthias Nagorni.

Commentaires et corrections

Comme toujours, un guide évolue dans le temps. Je vous encourage à me transmettre (en anglais) vos suggestions ou informations sur les nouvelles applications et les développements intéressants du MIDI sous Linux.

Par leur grande variété, les cartes son sont parfois difficiles à configurer, notamment parce qu'elles ont plusieurs niveaux de pilotes et d'options de configuration. Les meilleures sources de renseignement en cas de dysfonctionnement sont les sites web de ALSA et OSS, mais aussi les forums, ainsi que les listes de discussions consacrées à l'audio sous Linux.

Adaptation française des guides pratiques Linux

L'adaptation française de ce document a été réalisée dans le cadre du Projet de traduction des Guides pratiques Linux.

Vous pourrez lire d'autres guides pratiques traduits et en apprendre plus sur ce projet en allant sur le site de l'association Traduc.org.

Si vous souhaitez apporter votre contribution, n'hésitez pas à nous rejoindre, nous serons heureux de vous accueillir.

N'hésitez pas à faire parvenir vos commentaires et suggestions concernant l'adaptation française de ce document à l'adresse : <commentaires CHEZ traduc POINT org>.