

# Jack HowTo

## Jack : quid ?

Jack (acronyme de *Jack Audio Connexion Kit*) est une application servant de base pour la gestion du son basse latence dans les distributions **GNU/Linux**<sup>1</sup>. Le rôle de cette application est de permettre la communication entre différentes applications de traitement sonore, notamment en leur permettant d'échanger des flux de données audio.

## Un serveur de son

Ce logiciel est ce qu'on appelle un **serveur**, et les applications de traitement sonore auxquelles il sert de base sont appelées les **clients**. Cet emprunt au monde de la restauration n'est pas anodin ; il représente les relations qui existent entre jack et les applications auxquelles il sert de base :

- il peut apporter des flux de données audio à une ou plusieurs applications clientes,
- il peut récupérer des flux de données audio depuis chacune des applications clientes.

Ainsi toute communication entre deux applications audio (par exemple, un synthétiseur, et un enregistreur audio) passeront par jack : chaque application deviendra un client de jack et l'on pourra faire en sorte de **connecter** les flux audio sortant de l'une aux entrées de l'autre, et réciproquement. Il est important de noter que *le fait de passer par le serveur de son jack n'implique aucune latence de traitement supplémentaire*.

Jack est donc l'élément central de la chaîne de traitement audio basse latence, et ceci est d'autant plus vrai que c'est également lui qui la gère la communication avec la (ou les) carte(s) son au moyen de son (leur) pilote (ou driver). En pratique, la carte son apparaîtra comme un client de jack parmi d'autres, susceptibles d'envoyer et de recevoir des flux audio et dont les entrées et les sorties pourront être connectées à travers jack.

A ce stade de la lecture, cette conception du système son peut paraître complexe ou obscure ; l'ensemble d'exemple et d'explications qui suivent devraient permettre d'éclairer tout cela.

***Note à l'attention des utilisateurs habitués aux logiciels audio monolithiques type Cubase, ProTools, Logic Audio, ou encore, dans une certaine mesure, Live :** le concept qui sous-tend jack et toutes les applications qui reposent dessus nécessitent un changement de point d'observation du système. En effet, en lieu et place d'un seul bloc (un **monolithe**) gérant l'intégralité du système et accueillant en son sein toutes les applications de traitement audio (généralement sous la forme de plugin), il faut envisager le système son basé sur jack comme un système modulaire, c'est-à-dire un système constitué d'un ensemble d'applications indépendantes les unes des autres (donc susceptibles de fonctionner y compris en l'absence des autres applications), et mise en relation à travers le serveur de son. Il n'y a donc plus d'hôte, plus d'obligation d'intégrer les logiciels sous la forme de plugins. Ce système possède des avantages (notamment en termes de robustesse) et des inconvénients (notamment en termes de complexité) par rapport à un système monolithique, mais l'objet de ce document n'est pas de traiter de ce débat.*

## Un logiciel démon

Jack est également ce que le monde de l'informatique appelle un logiciel **démon**, c'est-à-dire qu'une fois lancé, il tourne en **tâche de fond**, et n'a **pas d'interface graphique**<sup>2</sup>.

<sup>1</sup> A noter que jack est également développé pour Mac OS et Windows.

<sup>2</sup> On verra dans la suite que, si jack ne possède pas d'interface graphique, des logiciels dédiés peuvent en fournir une.

# Configurer Jack

Comme expliqué dans le chapitre précédent, jack est le logiciel en charge de la gestion de la (ou les) carte(s) son, et à ce titre, il nécessite une configuration, ne serait-ce qu'en termes de fréquence d'échantillonnage, taille des buffers, profondeur d'encodage, etc.

A ce stade, comme à chaque fois qu'on voudra interagir avec jack par la suite, plusieurs solutions s'offrent à nous :

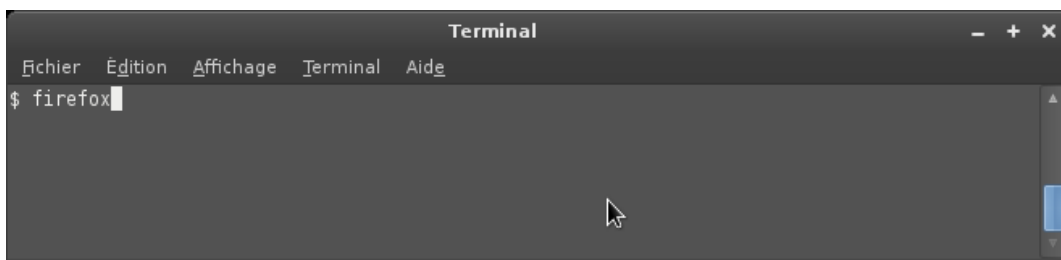
- configurer et lancer le démon jack à l'aide d'une ligne de commande,
- configurer et lancer le démon jack au moyen d'une interface graphique.

## Jack en ligne de commande

La ligne de commande est un outil historique des distributions GNU/Linux. On peut aujourd'hui totalement s'en passer, cependant, cet outil peut parfois représenter un gain de temps énorme ou un moyen de compréhension très intéressant. Aussi, ce document le considérera au même titre que les interfaces graphiques.

### Ligne de commande : quelques éléments d'utilisation

Traditionnellement, une application est lancée au moyen d'une action de la souris (typiquement double-clic) sur une icône représentant l'application. Cette action est en fait reliée à un **l'exécutable** de l'application, c'est-à-dire un fichier contenant le programme informatique qui constitue l'application. La ligne de commande permet de lancer l'application en l'appelant (c'est-à-dire en tapant son nom) depuis un terminal :



Par exemple, l'image ci-dessus lancera l'application Firefox au même titre que le double-clic sur son icône. Le mot « firefox » tapé dans le terminal est ce qu'on appelle une « commande ».<sup>3</sup>

Le symbole \$ est ce qu'on appelle l'invite de commande ; l'apparition de ce symbole signifie que l'on peut écrire une commande.

La commande peut être suivie d'un ensemble d'options (voir ci-après) qui permettent de paramétrer finement la façon dont l'application sera lancée.

### configuration et lancement de jack

La commande pour lancer jack est « jackd » pour Jack Daemon. Pour savoir comment utiliser cette commande, on peut taper « jackd --help » qui nous liste l'ensemble des options disponibles pour la commande jackd :

---

<sup>3</sup> L'une des difficultés de la ligne de commande est la connaissance de la commande (nom du programme) dont on a besoin. La touche **tabulation** permet de s'affranchir de ce problème, elle permet la complétion de la commande en partant des premières lettres de cette dernière : par exemple taper « fir » puis *tabulation* permettra d'écrire « firefox ».

```
Terminal
Fichier  Édition  Affichage  Terminal  Aide
$ jackd --help
jackd 0.118.0
Copyright 2001-2009 Paul Davis, Stephane Letz, Jack O'Quinn, Torben Hohn and others.
jackd comes with ABSOLUTELY NO WARRANTY
This is free software, and you are welcome to redistribute it
under certain conditions; see the file COPYING for details

usage: jackd [ --no-realtime OR -r ]
           [ --realtime OR -R [ --realtime-priority OR -P priority ] ]
  (the two previous arguments are mutually exclusive. The default is --realtime)
           [ --name OR -n server-name ]
           [ --no-mlock OR -m ]
           [ --unlock OR -u ]
           [ --timeout OR -t client-timeout-in-msecs ]
           [ --port-max OR -p maximum-number-of-ports ]
           [ --debug-timer OR -D ]
           [ --no-sanity-checks OR -N ]
           [ --verbose OR -v ]
           [ --clocksource OR -c [ c(ycle) | h(pet) | s(ystem) ] ]
           [ --replace-registry ]
           [ --silent OR -s ]
           [ --version OR -V ]
           [ --nozombies OR -Z ]
  -d backend [ ... backend args ... ]
    Available backends may include: alsa, dummy, freebob, firewire, net, oss, sun, o
r portaudio.

    jackd -d backend --help
    to display options for each backend

$
```

Pour une première approche on retiendra de cette liste les options suivantes :

- `--realtime` ou `-R` : qui permet de démarrer le serveur de son en mode temps-réel (c'est-à-dire que les latences seront garanties),
- `-d backend` : qui permet de préciser le pilote de la carte son qu'on souhaite utiliser :
  - `alsa` : pour les périphériques PCI (cartes internes) et USB
  - `freebob` ou `firewire` : pour les périphériques FireWire.

Prenons le cas d'une carte interne PCI, on va donc utiliser le pilote ALSA, la liste des options pour ce pilote est obtenue grâce à la commande « `jackd -d alsa --help` » :

```
Terminal
Fichier  Édition  Affichage  Terminal  Aide

$ jackd -d alsa --help
jackd 0.118.0
Copyright 2001-2009 Paul Davis, Stephane Letz, Jack O'Quinn, Torben Hohn and others.
jackd comes with ABSOLUTELY NO WARRANTY
This is free software, and you are welcome to redistribute it
under certain conditions; see the file COPYING for details

Memory locking is unlimited - this is dangerous. You should probably alter the line:
    @audio - memlock unlimited
in your /etc/limits.conf to read:
    @audio - memlock 2317911
no message buffer overruns
Parameters for driver 'alsa' (all parameters are optional):
    -C, --capture      Provide capture ports.  Optionally set device (default: none)
    -P, --playback     Provide playback ports.  Optionally set device (default: none)
    -d, --device       ALSA device name (default: hw:0)
    -r, --rate         Sample rate (default: 48000)
    -p, --period       Frames per period (default: 1024)
    -n, --nperiods     Number of periods of playback latency (default: 2)
    -H, --hwmon        Hardware monitoring, if available (default: false)
    -M, --hwmmeter     Hardware metering, if available (default: false)
    -D, --duplex       Provide both capture and playback ports (default: true)
    -s, --softmode     Soft-mode, no xrun handling (default: false)
    -m, --monitor      Provide monitor ports for the output (default: false)
    -z, --dither       Dithering mode (default: n)
    -i, --inchannels   Number of capture channels (defaults to hardware max) (default: 0)
    -o, --outchannels  Number of playback channels (defaults to hardware max) (default: 0)
    -S, --shorts       Try 16-bit samples before 32-bit (default: false)
    -I, --input-latency Extra input latency (frames) (default: 0)
    -O, --output-latency Extra output latency (frames) (default: 0)
    -X, --midi         ALSA MIDI driver (seq|raw) (default: none)

$
```

De cette liste, on retiendra les options génériques communes à la plupart des pilotes :

- --rate ou -r : la fréquence d'échantillonnage,
- --period ou -p : le nombre d'échantillons par période,
- --nperiods ou -n : le nombre de périodes utilisées par le pilote.

Ainsi, pour lancer jack avec une carte PCI dont la fréquence d'échantillonnage est réglée à 44100 Hz, la période à 512 échantillons et le nombre de périodes à 3, on tapera : « jackd -d alsa -r 44100 -p 512 -n 2 » :

```
Terminal
Fichier  Édition  Affichage  Terminal  Aide
$ jackd -d alsa -r 44100 -p 512 -n 2
jackd 0.118.0
Copyright 2001-2009 Paul Davis, Stephane Letz, Jack O'Quinn, Torben Hohn and others.
jackd comes with ABSOLUTELY NO WARRANTY
This is free software, and you are welcome to redistribute it
under certain conditions; see the file COPYING for details

Memory locking is unlimited - this is dangerous. You should probably alter the line:
    @audio - memlock unlimited
in your /etc/limits.conf to read:
    @audio - memlock 2317911
no message buffer overruns
JACK compiled with System V SHM support.
loading driver ..
SSE2 detected
apparent rate = 44100
creating alsa driver ... hw:0|hw:0|512|2|44100|0|0|nomon|swmeter|-.|32bit
control device hw:0
configuring for 44100Hz, period = 512 frames (11.6 ms), buffer = 2 periods
ALSA: final selected sample format for capture: 32bit integer little-endian
ALSA: use 2 periods for capture
ALSA: final selected sample format for playback: 32bit integer little-endian
ALSA: use 2 periods for playback
```

Jack tourne désormais avec les paramètres qu'on lui a fournis, et nous fait état du déroulement des actions qu'il mène.

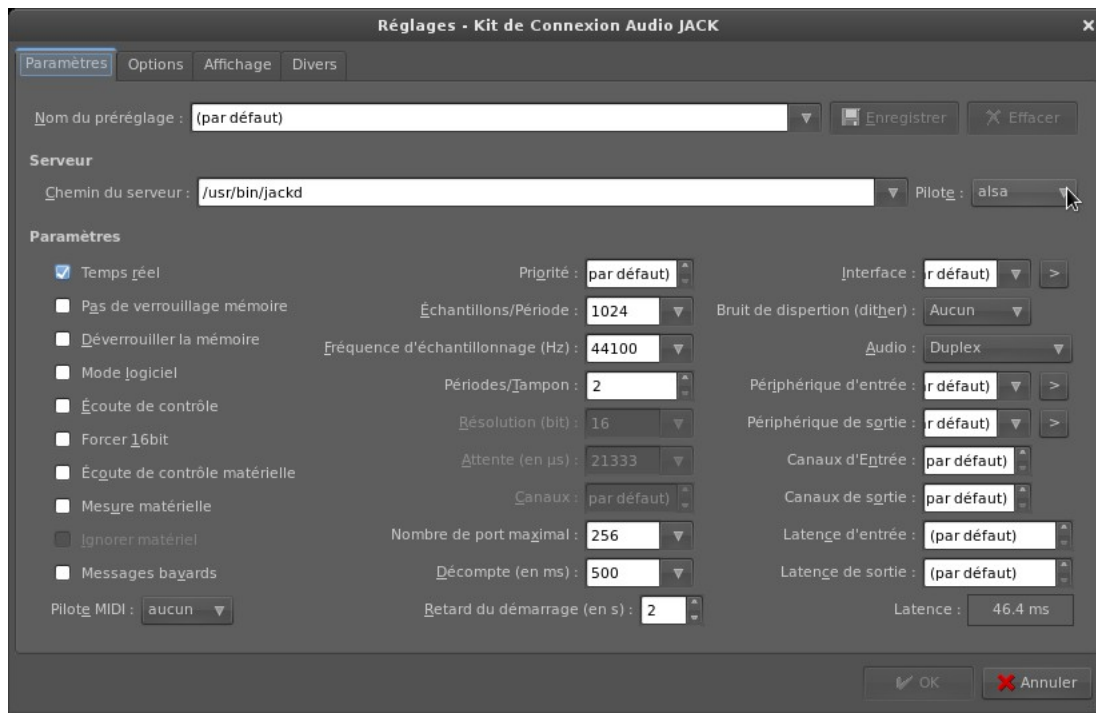
## Jack en mode graphique

Plusieurs interfaces graphiques sont disponibles pour interagir avec jackd. La plus usitée et la plus connue est très certainement qjackctl, également nommée Jack Control. Selon les distributions et les gestionnaire de fenêtres utilisés, la méthode pour lancer Jack Control ne sera pas la même ; néanmoins, on trouvera généralement cette application dans le sous-menu dédié au son.

La configuration du serveur jack se fait à l'aide du bouton « Réglages » de l'interface Jack Control.



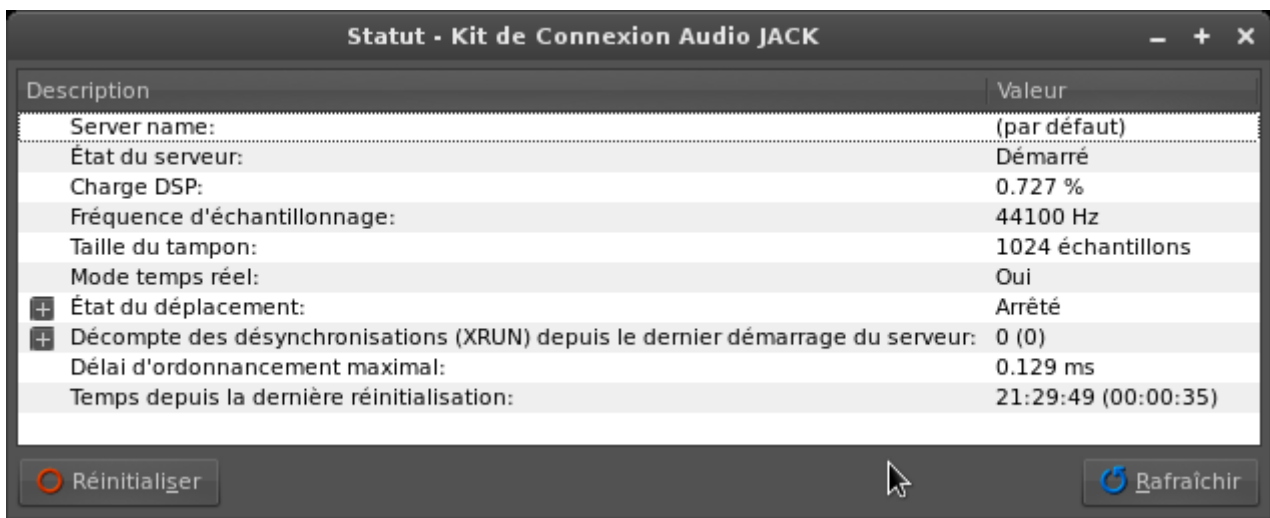
Ce bouton mène sur une nouvelle fenêtre permettant de régler les différents paramètres du serveur (pilote, fréquence d'échantillonnage, buffer, etc.). Pour la compréhension des différents paramètres, se référer au chapitre précédent.



Attention : le lancement de l'interface graphique ne signifie pas le lancement du serveur de son. En effet, celui ci peut être paramétré avant d'être mis en route. La mise en route du serveur de son se fait au moyen du bouton « Démarrer ».

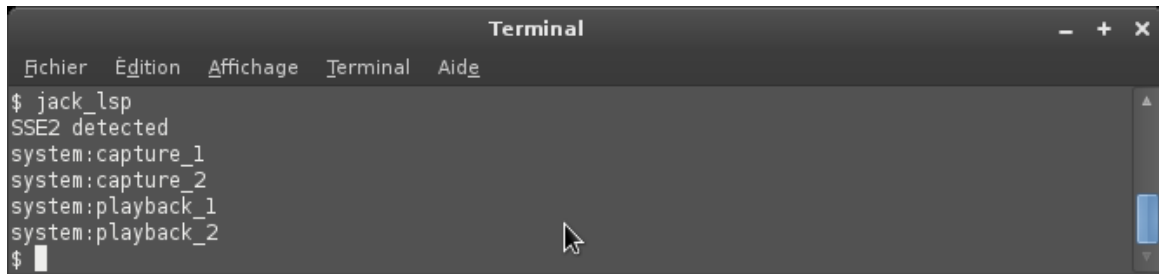


Une fois démarré, certains paramètres et informations apparaissent dans le cadran central (fréquence d'échantillonnage, charge DSP, nombre de désynchronisations, etc.), ces informations peuvent être retrouvées de manière plus détaillée au moyen du bouton « Statut ».

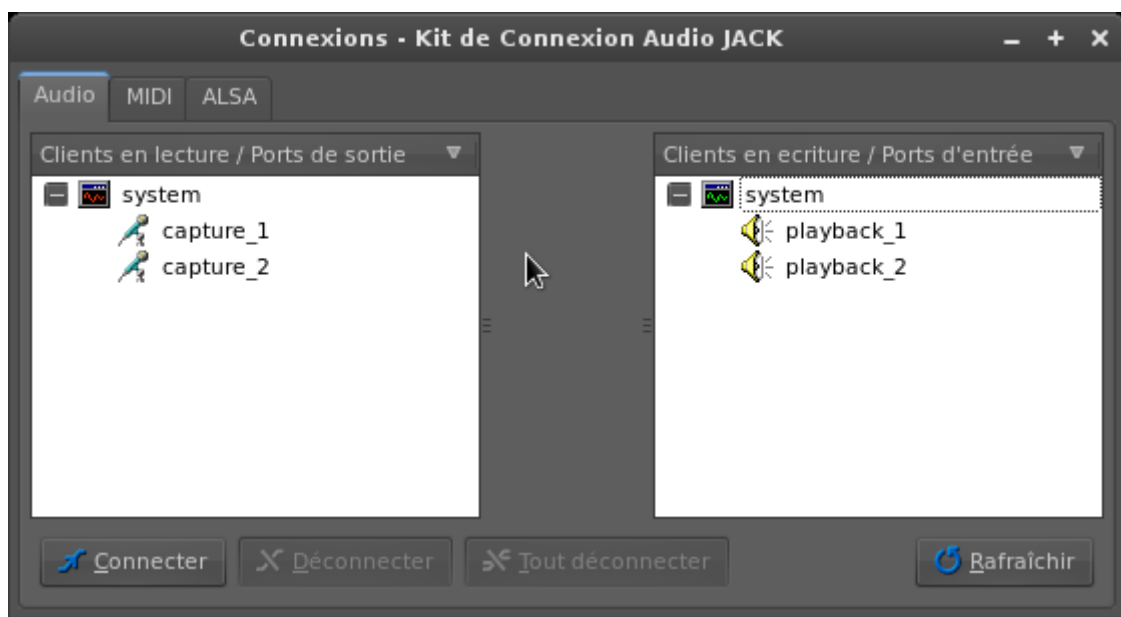


# Les connexions dans jack

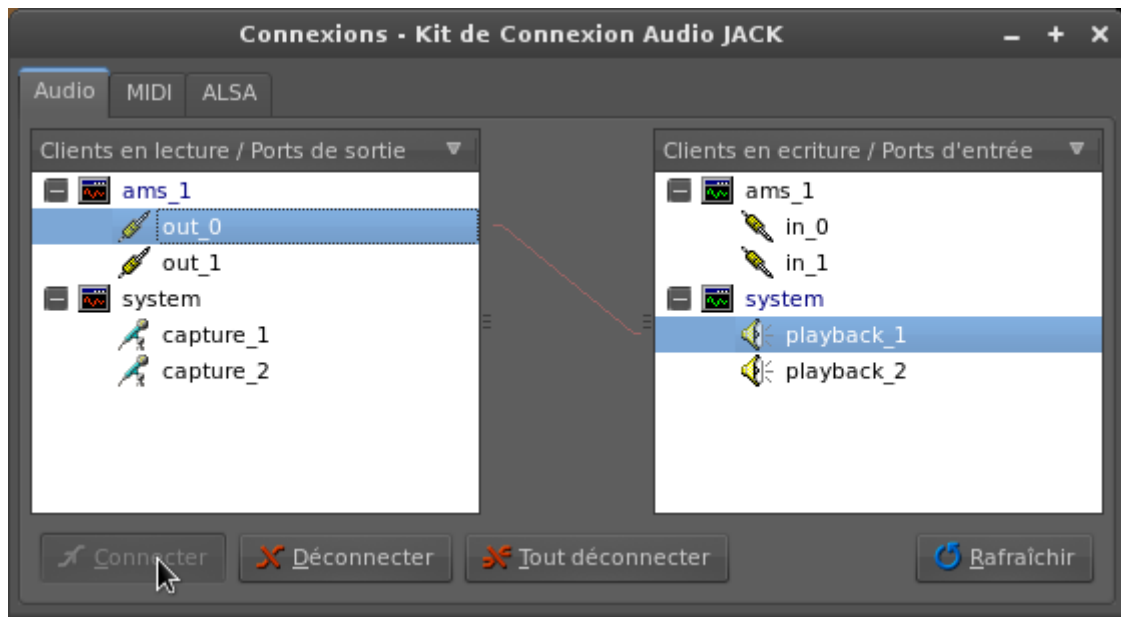
Lorsque jack est lancé, on peut alors commencer à lancer et connecter des applications clientes entre elles. La première application apparaissant est la carte son (qui prend généralement le nom « system »), qui présente spontanément ses entrées et ses sorties, qui constitueront, en langage jack, les ports de cette application cliente. On peut voir les clients disponibles et leurs ports en utilisant la commande `jack_lsp` ou grâce au bouton Connexions de Jack Control.



```
Terminal
Fichier  Édition  Affichage  Terminal  Aide
$ jack_lsp
SSE2 detected
system:capture_1
system:capture_2
system:playback_1
system:playback_2
$
```



Lorsqu'une nouvelle application sera lancée, elle se présentera de la même manière, et on pourra l'observer à travers `jack_lsp` ou Jack Control. Pour connecter la sortie audio d'une application cliente à l'entrée audio d'une autre application, il suffit de sélectionner la sortie audio dans la zone de gauche, l'entrée dans la zone de droite, et de cliquer sur le bouton « Connecter ».



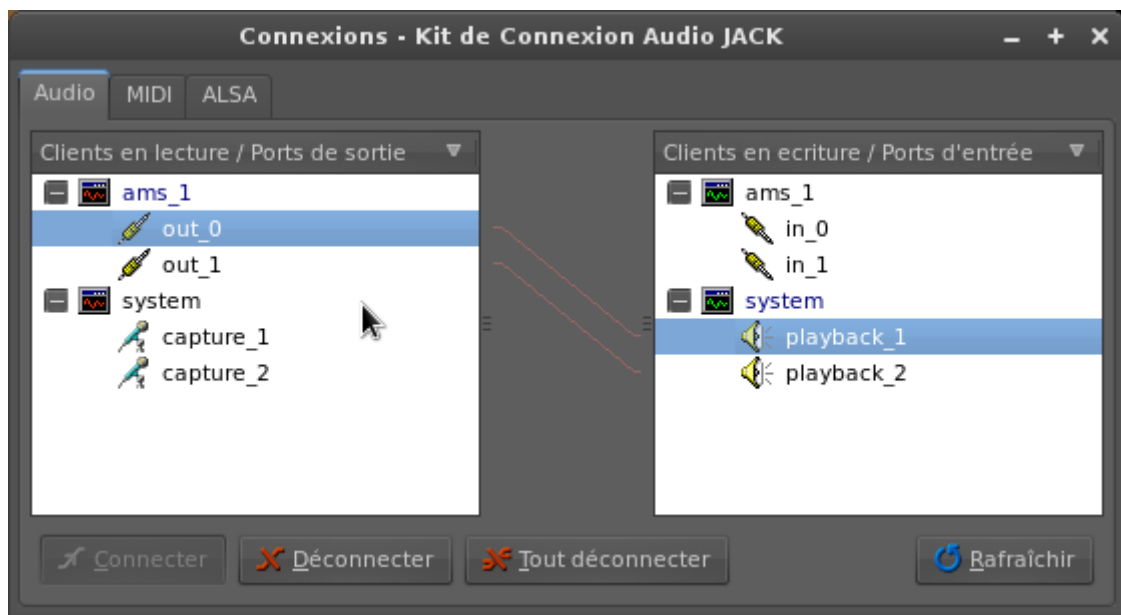
Une autre solution est d'utiliser la commande `jack_connect` :

```

Terminal
Fichier  Édition  Affichage  Terminal  Aide
$ jack_connect ams_1:out_1 system:playback_2
SSE2 detected
$

```

Ces deux options sont strictement équivalentes, et d'ailleurs, on peut observer l'effet de la commande `jack_connect` dans l'interface Jack Control (et réciproquement) :

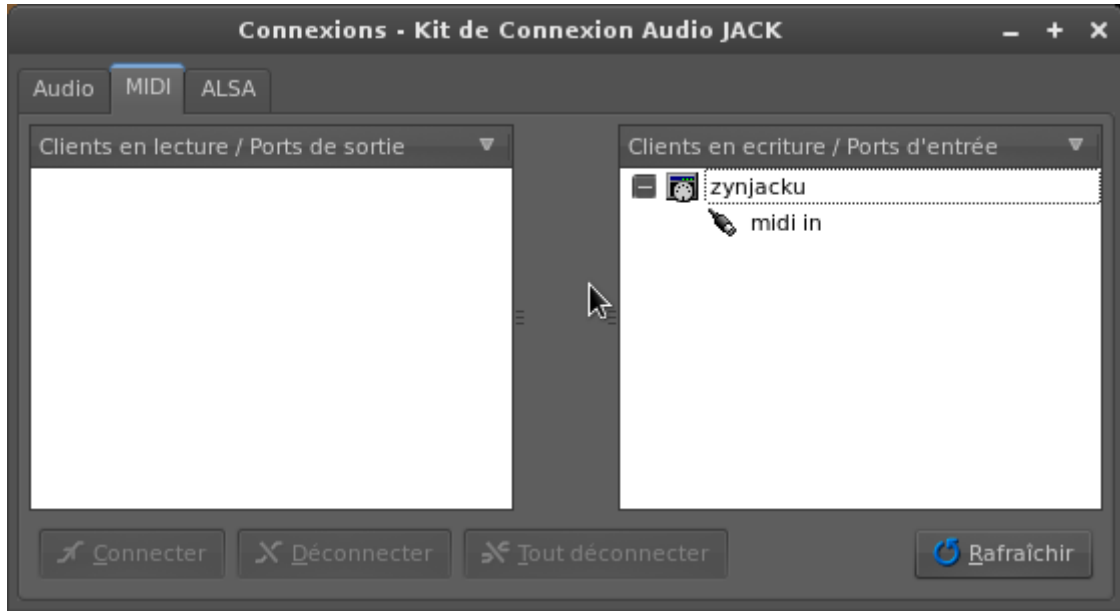


La logique à suivre alors est celle que l'on suivrait si on voulait câbler un patch de scène, c'est-à-dire une logique de câblage pure et simple. Lorsqu'un doute se présente, une analogie avec des tubes d'eau peut parfois permettre de résoudre le problème.



## Et le MIDI ?

Outre l'audio, jack gère également le MIDI. Pour autant, actuellement, il partage cette compétence avec un autre pilote : ALSA Seq. La gestion des connexions MIDI est strictement équivalente celle de l'audio pour la partie gérée par jack. On pourra ainsi utiliser la commande `jack_connect` (voir ci-dessus), ou l'onglet MIDI de la partie « Connexions » de l'interface Jack Control.



Les clients d'ALSA Seq pourront être gérés via la commande `aconnect`, ou via l'onglet ALSA de la fenêtre Connexions de Jack Control.

```
Terminal
Fichier  Édition  Affichage  Terminal  Aide

$ aconnect -i -o -l
client 0 : 'System' [type=noyau]
  0 'Timer'
  1 'Announce'
    Connexion à: 129:0, 132:0
client 14 : 'Midi Through' [type=noyau]
  0 'Midi Through Port-0'
client 128 : 'TiMidity' [type=utilisateur]
  0 'TiMidity port 0'
  1 'TiMidity port 1'
  2 'TiMidity port 2'
  3 'TiMidity port 3'
client 130 : 'ams_1 Midi' [type=utilisateur]
  0 'ams in'
  1 'ams out'
    Connexion à: 131:0
  2 'ams out'
client 131 : 'WhySynth 20100922 DSSI plugin' [type=utilisateur]
  0 'WhySynth 20100922 DSSI plugin'
    Connexion depuis: 130:1
$
```

