

Première partie

Tutoriel pour l'utilisation de Qemu

Table des matières

I	Tutoriel pour l'utilisation de Qemu	1
	Table des matières	2
1	Présentation	3
2	Installation.	4
2.1	Téléchargement des sources.	4
2.2	Compilation des sources pour le noyau Linux 2.6.x.	4
2.3	Vérification des droits.	5
3	Qemu par la pratique	7
3.1	Utiliser un LiveCD	7
3.2	Installer un OS dans un fichier virtuel	7

Chapitre 1

Présentation

Ce tutoriel explique l'installation de Qemu 0.8.1 avec le module Kqemu 1.3pre7 et son utilisation sommaire sous Linux.

QEMU permet de faire tourner un ou plusieurs systèmes d'exploitation (et leurs applications) de manière isolée sur une même machine physique. QEMU fonctionne sur les plateformes x86, PPC, Sparc, ARM et bientôt x86-64. QEMU fonctionne sous les systèmes d'exploitation Linux, Mac OS X, Unix et Windows.

QEMU est un outil de virtualisation libre fiable et performant. Les OS invités partagent ainsi les ressources de la machine physique.

QEMU est un « émulateur de système » ou une « machine virtuelle ». Les OS invités n'ont pas « conscience » du QEMU sous-jacent, ils n'ont pas besoin d'être « portés » (adaptés) pour fonctionner sur QEMU.

Le module noyau Linux KQEMU (pour Kernel QEMU) permet d'accélérer l'émulation sur les systèmes d'exploitation Linux.

Source de la définition : Wikipédia [<http://fr.wikipedia.org/wiki/QEMU>]

Ce tutoriel est écrit sous Slackware Current, bien que je tente d'être le plus neutre possible il se peut que vous deviez faire certaines adaptations selon votre distribution pour les chemins des répertoires de kqemu entre autre, surtout pour les distributions fortement modifiées (Mandriva, Fedora Core, Debian, ...).

Licences

- QEMU is a trademark of Fabrice Bellard.
The QEMU Accelerator is free to use, but it is a closed source proprietary product.
<http://fabrice.bellard.free.fr/qemu/license.html>
- Éditeur $\LaTeX$ Utilisé pour réaliser ce document : Vim - <http://www.vim.org/>
- Écrit sous Slackware Linux
Slackware is a registered trademark of Patrick Volkerding and Slackware Linux, Inc.

Document sous licence Creative Commons : Attribution-NonCommercial-ShareAlike 2.0

<http://www.creativecommons.org>

Marc Poiroud <marciun@free.fr> 2006 ©

Chapitre 2

Installation.

2.1 Téléchargement des sources.

Les archives **QEMU** et **QEMU Accelerator Module (kqemu)** sont téléchargeables à l'adresse :
<http://fabrice.bellard.free.fr/qemu/download.html>

2.2 Compilation des sources pour le noyau Linux 2.6.x.

Décompresser l'archive Qemu :

```
$ tar xzvf qemu-0.8.1.tar.gz
```

```
$ cd qemu-0.8.1/
```

Installer Qemu

```
$ ./configure
```

L'exécution du configure devrait ressembler à ceci :

```
./configure
```

```
Usage: sdl-config [--prefix[=DIR]] [--exec-prefix[=DIR]] [--version] [--cflags] [--libs]
      [--static-libs]
```

```
Usage: sdl-config [--prefix[=DIR]] [--exec-prefix[=DIR]] [--version] [--cflags] [--libs]
      [--static-libs]
```

```
Install prefix      /usr/local
```

```
BIOS directory      /usr/local/share/qemu
```

```
binary directory    /usr/local/bin
```

```
Manual directory    /usr/local/share/man
```

```
ELF interp prefix   /usr/gnemul/qemu-%M
```

```
Source path         /home/marc/compile/XAP/qemu/qemu-0.8.1
```

```
C compiler           gcc
```

```
Host C compiler      gcc
```

```
make                 make
```

```
install              install
```

```
host CPU              i386
```

```
host big endian      no
```

```
target list           i386-user arm-user armeb-user sparc-user ppc-user mips-user mipsel-user
                      i386-softmmu ppc-softmmu sparc-softmmu x86_64-softmmu mips-softmmu mipsel-softmmu
                      arm-softmmu
```

```
gprof enabled        no
```

```
profiler              no
```

```
static build         no
```

```
SDL support           yes
```

```
SDL static link       no
```

```
mingw32 support       no
```

```
Adlib support         no
```

```
CoreAudio support     no
```

```
ALSA support          no
```

```
DSound support        no
```

```
FMOD support          no
```

```
kqemu support         yes
```

```
Documentation         yes
```

Le programme sera installé dans `/usr/local`, si vous souhaitez l'installer dans le répertoire `/usr` vous devez le préciser lors du configure : `./configure prefix=/usr`.

Plusieurs options sont possibles, voir `./configure --help`

On y retrouve les répertoires d'installation `/usr/local`, le compilateur (gcc), le processeur cible (i386), les processeurs de l'émulateur et surtout le support de `kqemu`, si la ligne `kqemu support` retourne `no`, alors le module `kqemu` n'est pas pris en compte.

```
$ make
$ su
password
# make install
```

Maintenant, on va compiler le module d'accélération `kqemu` :

Dans un premier temps, il faut vérifier que vous possédez les sources de votre noyau.

```
$ ls /var/log/packages/kernel*
- /var/log/packages/kernel-source-2.6.15.4-i486-1
```

Dans le cas où les sources ne sont pas installés vous savez ce qui vous reste à faire.

Installer KQemu

```
$ tar xzvf kqemu-1.3.0pre7.tar.gz
$ cd kqemu-1.3.0pre7
$ ./configure
L'exécution du configure devrait ressembler à ceci :
Source path      /home/$USER/chemin/vers/kqemu-1.3.0pre7
C compiler       gcc
Host C compiler  gcc
make             make
host CPU         i386
```

```
kernel sources   /lib/modules/2.6.15.4/build
kbuild type      2.6
```

Si vous avez donné l'option `-prefix=/usr` à Qemu, vous devez évidemment en faire de même pour le module KQemu.

```
$ make
$ su
password
# make install
```

2.3 Vérification des droits.

Arrivé à ce stade, vous devez avoir Qemu et le module Kqemu installés sur votre système, cependant pour que le module `kqemu` soit pris en compte à l'utilisation, il nous reste des ajouts et vérifications mineures.

Kqemu est installé dans `/lib/modules/kernel_version/misc`. Le fichier de périphérique `'/dev/kqemu'` est créé avec les droits de lecture et d'écriture pour tout le monde, commençons par le vérifier :

```
$ ls -l /dev | grep kq
crw-rw----  1 root root    10,  63 2005-07-31 10:00 kqemu
```

On peut voir que les droits de lecture et d'écriture ne sont pas attribués aux utilisateurs. Voici la démarche pour corriger ce problème :

```
$ su -
password
# cd /dev
# chmod 666 kqemu
# ls -l kqemu
crw-rw-rw-  1 root root 10, 63 2005-07-31 10 :00 kqemu
```

Pour une raison encore inconnue, les droits ne restent pas fixés chez moi. Prenez le temps de vérifier les droits lors de l'utilisation de Qemu avec le module `kqemu`.

Pour que le module `kqemu` soit créé automatiquement au démarrage, vous devez ajouter au fichier `/etc/rc.d/rc.local` les lignes suivantes :

```
#!/bin/sh
#
# /etc/rc.d/rc.local: Local system initialization script.
#
# Put any local setup commands in here:
#
# Charger le module KQEMU
/sbin/modprobe kqemu major=0
```

Pour les distributions utilisant *udev*, l'option `major=0` est nécessaire. Dans le cas où votre distribution n'utilise pas *udev*, vous pouvez donner la valeur 250 ou N.

Autres méthodes

Il existe sûrement un binaire pour votre distribution, il vous faudra chercher sur les sites de contribution officiel de votre distribution si vous ne voulez pas le compiler par vous même. Vous devrez tout de même faire attention à installer un paquetage correspondant à votre noyau.

- Slackware - <http://www.slackware.com/alien/slackbuilds/qemu/pkg/>
- Red Hat/Fedora - <http://dag.wieers.com/packages/qemu/>
- RH/Fedora, Mandriva, SuSe - <http://rpm.pbone.net/index.php3?stat=3&search=qemu&srodzaj=3>

Qemu supporte l'instruction `DESTDIR=...`, donc si vous souhaitez construire un paquetage pour votre distribution préférée, il vous faudra le faire lors du `make install DESTDIR=/tmp/qemu`.

La commande `su` peut être remplacée par `fakeroot`, qui présente l'avantage de ne pas passer en super-utilisateur. Le paquet `fakeroot` est disponible à cette adresse : <http://packages.debian.org/stable/utils/fakeroot>

Chapitre 3

Qemu par la pratique

Avec Qemu vous pouvez :

- utiliser un LiveCD sans redémarrer,
- installer un OS dans un fichier virtuel.

3.1 Utiliser un LiveCD

Je vais décrire la méthode pour utiliser un LiveCD avec Qemu en une ligne de commande. Il est tout à fait possible d'émuler l'image ISO d'un liveCD de la même façon.

Par défaut, Qemu émule la mémoire vive avec 128 Mo pour ajouter plus de mémoire, vous devez spécifier l'option `-m 256` ou `-m 512 ...`

Mettre le CD dans le lecteur :

```
$ eject  
<mettre le CD dans le lecteur>  
$ eject -t
```

Il n'est pas nécessaire de monter de LiveCD ...

Lancer le LiveCD avec Qemu :

```
$ qemu -cdrom /dev/cdrom  
$ qemu -cdrom mon_image.iso
```

À partir de cet instant, une fenêtre devrait apparaître.

Pour réutiliser la souris du système hôte pendant l'émulation, vous devez faire : **Ctrl+Alt**

Pour clore le LiveCD, vous pouvez fermer comme vous le feriez avec n'importe quelle distribution, ou bien de manière plus « sale » en fermant simplement la fenêtre.

3.2 Installer un OS dans un fichier virtuel

C'est essentiellement pour cela que Qemu est destiné en effet il vous permet d'installer un OS sans toucher à vos disques durs simplement en créant un fichier virtuel. Pour cette exemple, je vais utiliser l'image ISO de **Zenwalk** nommé *zenwalk-2.2.iso*.

Préparation

Créer une image ISO

À partir d'un CD récupéré dans un magazine ou prêté par un copain¹, vous devez simplement monter le CD, puis créer l'image à l'endroit voulu.

```
$ mkisofs -r -o /home/user/zenwalk-2.2.iso /dev/cdrom
```

les options correspondent à :

-r : Créer l'image ISO9660

-o fichier : Nom du dossier auquel l'image du système de fichier *iso9660* devrait être écrite.

/dev/cdrom : répertoire ou périphérique de lecture, pour notre cas on lit de CD.

¹Je parle bien sûr d'un OS libre sinon vous devez posséder la licence de l'OS à installer

Création de l'image disque

Image disque servant de support à l'installation²

```
$ qemu-img create ZW-2.2.img 5G
```

Installation

Lancer l'installation de Zenwalk par l'émulateur Qemu³

```
$ qemu -hda ZW-2.2.img -cdrom zenwalk-2.2.iso -m 256 -boot d
```

-hda : correspond à votre disque, enfin au fichier que vous avez créé pour qu'il soit votre disque.⁴

arch_0.7_fr.img : est l'image disque créée ci dessus. à remplacer par tout fichier *xxx.img*

-cdrom arch-0.7-FR.iso : l'image ISO téléchargée est lue comme le cdrom.

-m 256 : taille virtuelle de la RAM. Par défaut la RAM est de 128 MB.

-boot a|c|d : Pour démarrer sur la disquette (**a**), le disque dur (**c**) ou le CDROM (**d**). Le démarrage sur le disque dur est par défaut.

À partir de là l'installation démarre et vous pouvez installer votre distribution sur votre disque virtuel.

En fin d'installation, le système devrait vouloir redémarrer mais cette manoeuvre échouera, le meilleur moyen est de fermer la fenêtre une fois arrivé à la ligne reboot, puis relancer normalement votre distribution (virtuelle) toute fraîche ↯

Relancer l'émulateur avec la distribution :

```
$ qemu -hda ZW-2.2.img -cdrom zenwalk-2.2.iso -m 256
```

En faire plus ...

Il existe une quantité d'option à donner à Qemu, pour ajouter ou enlever des services au démarrage de votre distribution virtuel. Je ne vais pas les détailler ici, car la lecture de la page man vous apportera toutes les réponses souhaitées.

²pensez à voir : `$ man qemu-img`

³pensez aussi et surtout à lire la page man : `$ man qemu`

⁴voir la ligne : Création de l'image disque ...