Emulation x86 facile avec Bochs

sam, 25/03/2006 - 20:13 — grapsus

Logiciels [1]

Bochs (http://bochs.sourceforge.net/ [2]
) est un émulateur x86 libre et facile d'accès, bien que plus lent que les logiciels concurrents il est très complet et pour émuler DOS, Linux (même avec un serveur X) ou encore Windows il peut suffire. En plus Bochs sait émuler du matériel générique hautement compatible avec les systèmes qu'il exécute. Vous pourrez lancer vos anciens jeux DOS avec le son et sans problèmes vidéo par exemple !

I. Installation

Comme presque tout logiciel libre vous pouvez installer Bochs avec votre distribution en paquet deb ou rpm, mais par exemple sous Debian stable le paquet "bochs" est une version datant de deux ans !

Je vous conseille donc de le compiler, surtout que la compilation est classique (***./configure, make, make install***). Les dernières sources sont disponibles ici [3]
Une fois l'archive téléchargée, compilez et installez la :

```
$ tar -zxvf bochs-2.X.X.tar.gz
$ cd bochs-2.X.X.tar.gz
$ ./configure
$ make
# make install (cette dernière commande est à exécuter en tant que root)
```

Voici les options du ***./configure*** les plus courantes, désactivées par défaut :

—enable-cpu-level=LEVEL	le type de processeur à émuler, par défaut Pentium, 386 si LEVEL vaut 3 et 486 si LEVEL vaut 4
—enable-sb16	émuler une carte son SoundBlaster
—enable-gameport	émuler un port de jeux
—enable-ne2000	émuler une carte réseau (prise en charge par Linux, BSD et Windows)
—enable-all-optimizations	peut accélérer Bochs, à utiliser avec précaution
—enable-x86-64	émuler les instructions d'un AMD 64bits
—enable-sse=LEVEL	émuler le jeu d'instructions SSE1 avec LEVEL=1 ou SSE2 avec LEVEL=2
—enable-mmx	émuler le jeu d'instructions MMX

Les autres options sont très bien expliquées ici : http://bochs.sourceforge.net/doc/docbook/user/compiling.html#CONFIG-OPTS [4]. Par défaut aucune option n'est nécessaire, mais si vous voulez émuler un processeur particulier ou un matériel regardez sur cette page ce qu'il vous faut comme options.

Le make install est assez propre, il n'installe que les fichiers suivants :

bin	binary executables (bochs, bxcommit, bximage)

lib/bochs/plugins	plugins (if present)
man/man1	manpages for installed binaries
man/man5	manpage for the config file (bochsrc)
share/bochs	BIOS images, VGABIOS images, keymaps
share/doc/bochs	HTML docs, license, readme, changes, bochsrc sample

Une dernière méthode d'installation pour Debian consiste à télécharger la dernière version de Bochs en rpm (sur le site Bochs) et de le convertir en .deb avec alien si vous êtes allergique aux compilations.

II. Charger une image disque avec bochs

Bochs fonctionne avec des fichiers images .img ou des vrais lecteurs (chargés avec leur fichier bloc dans /dev) pour émuler des disques durs, des disquettes ou des lecteurs CDROM et avec un fichier de configuration bochsrc qui spécifie les paramètres de la machine à émuler, à quel disque associer quelle image etc...
Pour vous entraîner à charger des images toutes faites vous pouvez télécharger des exemples sur http://bochs.sourceforge.net/diskimages.html [5].

Une fois le dossier dezzipé, placez vous dans ce dossier et chargez le fichier bochsrc avec :

```
$ bochs -f bochsrc
```

Une fois Bochs lancé, appuyez sur entrée pour commencer la simulation. Une fenêtre X apparaît alors et votre ordinateur virtuel charge son BIOS puis votre disque dur virtuel ! Les boutons en haut de la fenêtre vous permettent de charger à la volée de nouvelles images dans les lecteurs de disquettes, d'activer ou de désactiver la souris et le clavier, de prendre une capture d'écran ou d'éteindre la machine virtuelle. Une fois la simulation terminée on trouve un fichier bochsout avec des informations diverses et même l'état du processeur à divers moments si vous avez compilé bochs avec l'option déboguage.

III. Créer ses propres images Bochs

Maintenant que vous savez lancer bochs, vous allez pouvoir créer vos propres images disque et les lancer !

Commencez par créer des images vides pour vos disques durs :

```
$ dd if=/dev/zero of=image.img bs=512 count=X
```

Avec X le nombre de blocs de 512 octets qui composeront votre disque dur par exemple 2K pour un disque de 1 mégaoctet.

Une autre solution consiste à utiliser***bximage***, c'est une sorte d'assistant livré avec Bochs, très complet et bien fait.

Une fois l'image faite, vous pouvez y créer un système de fichiers, par exemple :

```
# mkfs.vfat image.img
```

Pour créer une image à partir d'un vrai disque dur ou d'une disquette :

```
# dd if=/dev/floppy of=floppy.img bs=512
```

On peut même monter une image disque (à condition qu'elle possède un système de fichiers et votre noyau prenne un charge un device de loopback) :

```
$ mkdir tmp
# mount -t vfat -o loop image.img ./tmp
```

On peut maintenant y copier des fichiers ! Ou même un secteur de boot, pour ce faire :

```
$ dd if=boot.bin of=image.img bs=512 count=1 conv=notrunc
```

Une fois que toutes les images sont prêtes, on doit créer un fichier bochsrc, placez-vous dans le dossier contenant vos images :

```
$ bochs
```

Sélectionnez l'option ***3 "Edit options"***, les parties les plus importantes à configurer sont ***6 "Memory options"*** qui permet de définir la quantité de ram à émuler et ***8 "Disk options"*** ou l'on choisit les images à charger pour les disques durs. Pour activer un disque dur choisissez cette dernière option, puis ***3 "ATA channel 0"*** pour activer un controlleur IDE, ***4 "First HD/CD on channel 0" [yes] [disk] [imagedisque.img] [flat] [defaut] [defaut] [defaut] [defaut] [auto] [auto]***. Il est important de choisir ***"Boot drive #1" disk***. Revenez au menu principal et sauvegardez votre configuration. Vous pouvez maintenant lancer la simulation !

Les fichiers générés par Bochs contiennent plein d'options inutiles, on peut éditer bochsrc à la main. Voici un fichier de configuration de base qui suffira à une bonne partie des simulations pas trop extravagantes :

```
# ram en Mo
megs: 32
#image BIOS vous pouvez essayer toutes celles de /usr/local/share/bochs/
romimage: file="/usr/local/share/bochs/BIOS-bochs-latest", address=0x00000
vgaromimage: file="/usr/local/share/bochs/VGABIOS-lgpl-latest"
#unité de démarrage, c=premier disque dur, floppy ou cdrom
boot: c
# no floppya
# no floppyb
ata0: enabled=1, ioaddr1=0x1f0, ioaddr2=0x3f0, irq=14
#un disque dur
ata0-master: type=disk, mode=flat, translation=auto, path="image.img", cylinders=0, heads=0,
spt=0, biosdetect=auto, model="Generic 1234"
#un cdrom
ata0-slave: type=cdrom, path=iso.sample, status=inserted
#une disquete
floppya: 1_44="disk.img", status=inserted
#ips est le nombre d'instructions à emuler par seconde,il dépend fortement de la machine sur
laquelle bochs est lancé
cpu: count=1, ips=2000000
mouse: enabled=0
```

Pour plus de détails reportez-vous à la documentation de Bochs !

Licence Creative Commons by-sa 3.

URL source: http://coagul.org/drupal/publication/emulation-x86-facile-bochs

Liens:
[1] http://coagul.org/drupal/rubrique/logiciels
[2] http://bochs.sourceforge.net/
[3] http://bochs.sourceforge.net/cgi-bin/topper.pl?name=See+All+Releases&url=http://sourceforge.net/project/showfiles.phpqmrkgroup_ideq12580
[4] http://bochs.sourceforge.net/doc/docbook/user/compiling.html#CONFIG-OPTS
[5] http://bochs.sourceforge.net/diskimages.html