

BOCHS

(<http://bochs.sourceforge.net/>)

➤ Téléchargement

Depuis le svn : `svn co http://svn.code.sf.net/p/bochs/code/trunk/bochs bochs`

➤ Configuration

```
./configure --enable-x86-64 --enable-vmx=2 --enable-cpu-level=6 --enable-es1370
--with-all-libs --enable-e1000 --enable-ltdl-install --enable-plugins --enable-ne2000
--with-sdl --with-svga --with-term --enable-smp --enable-all-optimizations
--enable-avx --enable-evex --enable-long-phy-address
```

➤ Build & intallation

`make && sudo make install`

➤ Launch

`sudo bochs (-q : quiet)`

Avant de pouvoir lancer, il faut modifier le fichier de configuration pour renseigner certains éléments essentiels tels que le disque image de la machine à émuler, les interfaces réseau, le cpu, la mémoire, etc.

➤ Fichier de config bochsrc

- Disques : chercher les configs `ata[0-3]`. Vous pouvez utiliser un disque `*ata0-master*` pour le bootloader (image de la VM) et un autre `*ata1-master*` comme disque externe à monter depuis la machine hôte
 - Depuis l'hôte:
 - ✧ `dd if=/dev/zero of=disque_à_monter.img bs=1G count=25`
 - ✧ `mkfs.ext4 disque_à_monter.img`
 - ✧ `mkdir disque_monté_depuis_lhote && sudo mount -t ext4 -o loop disque_à_monter.img disque_monté_dans_lhote/ ==> maintenant chaque fois que l'on copiera quelque chose dans ce dossier il sera accessible depuis le guest bochs`
 - Depuis le guest (à l'intérieur de la machine bochs une fois qu'on l'a démarrée):
 - ✧ `sudo fdisk -l` (pour trouver le chemin vers notre `ata1-master`, ce sera quelque chose du genre `/dev/sdX`)
 - ✧ `mkdir disque_monté && sudo mount /dev/sdX disque_monté/`
- Réseau : chercher la config `e1000`. Tout d'abord il faut être connecté en ethernet, et depuis ifconfig récupérer l'adresse ethernet de l'hôte (disons

ethX), son masque ainsi que la gateway du réseau (*route -n*). S'il y a plusieurs interfaces ethernet, retrouver celle active en faisant *nmcli dev status*

- Sur la ligne *e1000* : mettre dans le champ *ethdev*, l'adresse de *ethX* précédemment notée
- Dans le guest bochs:
 - ✧ Retrouver le nom de l'interface ethernet (chez moi *enp0s2* et *enp0s3* en fonction de l'adaptateur utilisé, *ne2k* ou *e1000* ==> attention vous n'aurez pas forcément des interfaces en *ethx*)
 - ✧ Configurer cette interface : *ifconfig enp0sX addr_hote.xxx/masque* (une adresse dans la plage du réseau dans lequel se trouve votre hôte), *add default gw gw_de_lhote*, *echo "nameserver 8.8.8.8" > /etc/resolv.conf*
 - ✧ Ensuite vérifier que vous parvenez bien à faire un *ping* vers la *gw* puis vers *8.8.8.8* et ensuite faire un *apt update*

Astuce : Du fait non seulement de l'émulation qui peut être extrêmement lente, mais également du fait que n'aurez très probablement pas une connexion par ethernet toujours à portée de main, il sera préférable pour vous d'effectuer les installations et mises à jour (y compris la création même de l'image de disque avec installation d'Ubuntu), de le faire avec XEN par exemple et de juste monter par la suite l'image de disque pour la démarrer avec bochs. Donc concrètement, vous créez une VM XEN dans laquelle vous installez ubuntu et faites toutes les mises à jour et installations nécessaires, et ensuite vous renseignez cette image dans *ata0-master* pour l'utiliser avec bochs.

- CPU (voir la ligne CPU du fichier) :
 - Pour avoir l'aide sur les versions de processeurs supportés : *bochs --help cpu (/features /etc.)*. Il faut modifier le type de CPU qui est présent dans le fichier de config par défaut en fonction des fonctionnalités que vous souhaitez exploiter.
 - Pour avoir PML et SPP il faut choisir le modèle : *corei7_icelake_u* (à partir de *skylake* on a PML), il faudra ensuite configurer convenablement le nombre de cpus en fonction du modèle (par exemple, de façon empirique je me suis rendu compte que pour le modèle *corei7_icelake_u* je ne pouvais pas utiliser plus de 2 CPUs (à voir de votre côté)).

Installation de XEN à l'intérieur de la machine émulée avec bochs

- Pour l'hyperviseur :
 - Copier simplement un noyau de XEN déjà compilé depuis votre machine physique hôte (*xen-xx.gz*) ainsi que le fichier de config y associé *xen-xx.config* dans le */boot* du guest bochs, car la compilation en émulation peut prendre 2 jours et 2 nuits!

- Pour l'y copier, vous pouvez par exemple penser à utiliser un disque monté. Faire ensuite un *update-grub* puis un *reboot* (bien sûr là on est dans bochs). Vous pourrez dès lors démarrer sur XEN dans bochs.
- Pour les tools :
 - Si vous avez compilé tout le dossier XEN y compris les tools depuis votre machine physique, vous devrez tout de même refaire l'installation des tools dans bochs. Pour cela, il faudra monter le dossier XEN dans bochs : attention à ce que les chemins soient exactement les mêmes pour la compilation. i.e. dans votre machine il faudra avoir fait l'installation par exemple depuis le dossier */mnt/chemin/...* et dans bochs il faudra également monter ce dossier dans */mnt/chemin/...* pour pouvoir faire le *make install*. Donc après avoir copié le noyau de xen, vous devrez monter le dossier de xen (dans bochs bien sûr) dans */mnt/* et vous déplacerez dans */mnt/xen/tools* pour faire un *sudo make install* ==> cette opération peut prendre des heures soyez patients et maintenez votre machine branchée tout le temps!
 - Faites enfin un *ldconfig* et *xencommons start* : vous pouvez à présent créer des machines virtuelles XEN depuis bochs. De même pour la création des VMs n'oubliez pas l'astuce : créez les depuis votre machine physique et réutilisez juste l'image pour démarrer la VM dans bochs!

Bonne bochs à tous!