

# BOCHS

Stella Bitchebe (bitchebe@i3s.unice.fr)

## ➤ Documentation

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

## ➤ Prerequisite

Install dependencies : `sudo apt update && sudo apt install docbook`

## ➤ Download

From 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)`

Before launching, you need to modify the configuration file to fill in some essential informations such as disk image to be emulated, network interfaces, CPU, memory, etc.

## ➤ Config file bochsrc

- Disks: search for the configs `ata[0-3]`. You can use one of the disks `*ata0-master*` as the bootloader (VM image) and another one `*ata1-master*` as an external disk to be mounted from inside the physical host machine.
  - From the physical host side:
    - ✧ `dd if=/dev/zero of=disk_to_be_mounted.img bs=1G count=25`
    - ✧ `mkfs.ext4 disk_to_be_mounted.img`
    - ✧ `mkdir disk_mounted_inside_host && sudo mount -t ext4 -o loop disk_to_be_mounted.img disk_mounted_inside_host/ ==> from now, anything copied inside this disk will be accessible from inside the bochs host.`
  - From the bochs host side (inside the bochs machine, once started):

- ✧ `sudo fdisk -l` (to find out the path to our disk *ata1-master*, it will be something like */dev/sdX*)
  - ✧ `mkdir mounted_disk && sudo mount /dev/sdX mounted_disk/`
- Network: search for the config *e1000*. First of all, you should be connected via ethernet; using `ifconfig` command, get the ethernet interface address of the physical host machine (let's say *ethX*), its mask and the gateway. If there are many ethernet interfaces, retrieve the active one by typing `nmcli dev status`.
  - To line *e1000* : fill in the field *ethdev* with the address of the previously noted *ethX* interface address
  - Inside the bochs host:
    - ✧ Find out the name of the ethernet interface (*enp0s2* and *enp0s3* for me, depending on the adapter, *ne2k* or *e1000* ==> be carefull, you won't necessarily have interfaces named *ethx*)
    - ✧ Configure that interface: `ifconfig enp0sX host_addr.xxx/mask` (an address in the same network bandwidth as the physical host), `add default gw gw_de_lhote, echo "nameserver 8.8.8.8" > /etc/resolv.conf`
    - ✧ Next, verify that you can ping the gw and 8.8.8.8, and then do an `apt update`

**Tip:** Because the emulation can be extremely slow, and also because an ethernet connection will most likely not always be available, it will be better for you to perform every installations and updates (including the creation of the disk image with Ubuntu installation), using XEN for example and then just mount that disk image to launch bochs. Concretely, you should create a XEN's VM in which you install Ubuntu and make all the necessary updates and installations, and then you fill in this image in *ata0-master* to use it with bochs.

- CPU (look at the line *CPU* in the config file) :
  - To obtain the help on processor versions: `bochs --help cpu (/features /etc.)`. After what, the type of CPU that is present in the default config file must be modified according to the functionalities that you wish to exploit.
  - To have PML and SPP supported, choose the model: *corei7\_icelake\_u* (PML is supported from *skylake version*). You will then need to suitably configure the number of cpus depending on the choosen model (for example, in an empirical way I realized that for the *corei7\_icelake\_u* model I could not use more than 2 CPUs (see on your side)).

## **XEN Installation inside the emulated bochs machine**

- Hypervisor installation:

- You simply have to copy an already compiled XEN kernel from your physical host machine (*xen-xx.gz*), and the associated config file (*xen-xx.config*), into the */boot* of the bochs host, because the compilation in emulation can take over 2 days and 2 nights!
- To perform the copy, think of using the mounted disk! Next perform an *update-grub* then a *reboot* (of course here we are in bochs). You will then be able to boot on XEN from bochs.
- Tools installation:
  - You will need to install tools in the bochs host (previously we have only copied the XEN kernel, that has nothing to do with the tools which are compiled for the dom0). For the purpose, you will need to mount the XEN folder in bochs: make sure that the paths are exactly the same for the compilation. i.e. if in your physical host machine the compilation has been done using the path */mnt/path/...* in the bochs host machine you should make sure to mount the XEN folder exactly in the same directory */mnt/path/...* before performing the *make install* (in */mnt/path/.../xen/tools*) ==> this operation may take some time (in terms of hours), be patient and keep your charger always connected!
  - Finally make a *ldconfig* and *xencommons start* : you can now create virtual machines in bochs. As well, for the VM creation don't forget the tip: create it in your physical host and reuse the disk image to start the VM in bochs!

Good bochs to all!