

FreeBSD sur un Raspberry Pi

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 3 janvier 2016

<https://www.bortzmeyer.org/raspberry-pi-freebsd.html>

Je viens d'installer FreeBSD sur un Raspberry Pi. Rien d'extraordinaire, je voulais juste expliquer pourquoi je ne garderai pas ce système en production.

Il y a désormais des images officielles pour le Raspberry Pi chez FreeBSD : <<ftp://ftp.freebsd.org/pub/FreeBSD/releases/ISO-IMAGES/10.2/>> (« ...arm-armv6-RPI-B... »). On télécharge, on décompresse, on copie sur un carte SD avec dd et on démarre le Pi dessus :

```
wget ftp://ftp.freebsd.org/pub/FreeBSD/releases/arm/armv6/ISO-IMAGES/10.2/FreeBSD-10.2-RELEASE-arm-armv6-RPI-B.i
unxz FreeBSD-10.2-RELEASE-arm-armv6-RPI-B.img.xz
sudo dd if=FreeBSD-10.2-RELEASE-arm-armv6-RPI-B.img of=/dev/sdb bs=1M conv=sync
[Démarrage du Pi ; il faut être patient la première fois, cela prend
plusieurs minutes pour générer les clés SSH.]
```

Ensuite, on peut se connecter au Pi en SSH (login/mot de passe freebsd/freebsd). On crée des comptes, on change les mot de passe, tout va bien :

```
amour% uname -a
FreeBSD amour 10.2-RELEASE FreeBSD 10.2-RELEASE #0 r286666: Thu Aug 13 03:16:22 UTC 2015      root@releng1.nyi.fr
amour% uptime
1:59PM up 39 mins, 2 users, load averages: 1.67, 1.83, 1.66
```

(Les messages de démarrage du noyau sont disponibles ici (en ligne sur <https://www.bortzmeyer.org/files/dmesg-FreeBSD-10.2-RPI.txt>.)

C'est ensuite que les ennuis commencent. Le programme `bsdinstall` censé permettre de configurer la machine facilement ne marche pas, la liste des miroirs n'étant pas adapté au processeur ARM : la plupart des miroirs n'ont pas ARM et on n'a aucun moyen de le savoir à l'avance. `bsdinstall` quitte donc rapidement. Il faut donc faire les réglages à la main. Par exemple, pour la date et le fuseau horaire :

```
ln -s /usr/share/zoneinfo/Europe/Paris /etc/localtime
ntpdate ntp.nic.fr
```

Et pour le clavier :

```
% kbdmap
kbdcontrol: getting keymap: Inappropriate ioctl for device
You are not on a virtual console - expect certain strange side-effects
lang_default = en
dialect = en
lang_abk = en
keymap="fr.kbd"
```

Mais le plus gênant concerne les paquetages. Dans la terminologie FreeBSD <https://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/ports.html>, le **port** est le source et le **paquetage** le binaire (pré-compilé, donc). Quand on a démarré sur l'image officielle, on n'a que le système de base. Pour des raisons historiques (enfin, plutôt, pré-historiques), FreeBSD maintient une division des anciens Unix, entre un système de base et les applications. La distinction est assez arbitraire (le système « de base » contient un compilateur, par exemple, alors que l'écrasante majorité des machines n'en ont pas besoin). Pour installer des applications utiles, il existe plusieurs mécanismes (et qui changent tout le temps, donc les vieux articles récupérés sur le Web ne vous aideront pas). Un peu au hasard, j'ai testé avec `pkg` :

```
# pkg update
The package management tool is not yet installed on your system.
Do you want to fetch and install it now? [y/N]: y
Bootstrapping pkg from pkg+http://pkg.FreeBSD.org/FreeBSD:10:armv6/quarterly, please wait...
pkg: Error fetching http://pkg.FreeBSD.org/FreeBSD:10:armv6/quarterly/Latest/pkg.txz: No address record
A pre-built version of pkg could not be found for your system.
Consider changing PACKAGESITE or installing it from ports: 'ports-mgmt/pkg'.
```

Le message « *"No address record"* » est erroné. Le site existe bien, mais n'a rien pour les processeurs ARM <<https://forums.freebsd.org/threads/raspberry-pi-and-running-pkg-issues.46564/>>. En fait, le fond du problème est qu'il n'y a pas de paquetages (donc de binaires) pour le Raspberry Pi; il paraît qu'ils apparaîtront dans la prochaine version de FreeBSD (je rappelle que les mises à jour du système sont très complexes avec FreeBSD, rien à voir avec ce que permet un système comme Debian, donc je n'envisage pas de passer dans les systèmes ultérieurs).

Il faut donc tout compiler soi-même ce qui, sur le Raspberry Pi, prend des siècles (et use la carte SD). J'ai utilisé `portsnap` pour extraire les ports (les sources, ou plutôt les références aux sources) comme suggéré par Johnson D <<https://fosskb.wordpress.com/2014/11/29/getting-started-with-freebsd/>> :

```
portsnap fetch
portsnap extract
```

puis, `cd /usr/ports/CATEGORY/SOFTWARE` et `make install`. C'est horriblement lent et je ne rêve même pas d'arriver à installer `emacs` ainsi. Donc, pas utilisable « en vrai ».

Autre problème, j'ai eu du mal à activer IPv6 (qui ne l'est pas par défaut, ce qui est incroyable en 2016). Il existe d'innombrables variables IPv6 pour le `/etc/rc.conf` et les explications sur leur interaction sont confuses. Mon réseau utilise SLAAC pour la configuration IPv6. Un peu au hasard, j'ai mis :

<https://www.bortzmeyer.org/raspberry-pi-freebsd.html>

```
ipv6_enable="YES"  
ipv6_activate_all_interfaces="YES"  
ipv6_privacy="YES"  
ifconfig_ue0_ipv6="inet6 accept_rtadv"
```

Où ue0 est le nom de l'interface Ethernet du Pi. Avec un peu de patience ensuite, cela marche.

Quelques documentations et articles :

— Le Wiki officiel <<https://wiki.freebsd.org/FreeBSD/arm/Raspberry%20Pi>> (très peu de contenu)

— Un autre récit d'expérience <<https://adminblog.foucry.net/FreeBSD-raspberrypi/>>, en français

Des alternatives possibles :

— NetBSD : annonce « officielle » <<https://www.raspberrypi.org/blog/netbsd-is-here/>> et un récit <<http://paulscomputernotes.blogspot.fr/2013/11/netbsd-on-raspberry-pi.html>>.

— NanoBSD : un récit <<http://www.freebsdnews.net/2014/05/10/nanobsd-raspberrypi/>>.