



Faire un chroot d'un système ARM sur un ordinateur x86_64

22 septembre 2019

Table des matières

| | | |
|----|--|---|
| 1. | Configuration du système de fichiers | 1 |
| 2. | Installation des paquets requis | 2 |
| 3. | Chroot avec QEMU | 3 |



Chroot un ARM système est un moyen simple d'avoir un accès à un système sans avoir de matériel spécifique, donc nous permettre de construire des paquets pour les appareils ARM. Cela peut être une alternative au "cross-compiling" où nous sommes limités à mettre en relation (linking) uniquement avec les bibliothèques du compilateur.

PS : le guide est aussi disponible sous forme de PDF avec une meilleur coloration syntaxique : [ici](#) ↗

1. Configuration du système de fichiers

1.0.0.1. Utilisation d'une carte SD Dans cette partie je suppose que vous avez déjà installé ArchlinuxARM sur la carte SD. Si ce n'est pas le cas, il faudra télécharger la dernière version du système et utiliser Etcher pour écrire le système sur la carte SD ou créer un disque virtuel (voir ci-dessous).

`/dev/sdbX` représente le chemin d'accès à la carte SD.

```
1 $ mkdir arm-chroot
2 $ mount /dev/sdbX arm-chroot
```

2. Installation des paquets requis

1.0.0.2. Créer et utiliser un disque virtuel Si vous ne disposez pas de carte SD avec ArchlinuxARM vous pouvez créer un disque virtuel pour disposer d'une image d'ArchlinuxARM. J'ai décidé de créer un disque virtuel d'une taille de 4 Gio que je monte dans mon système de fichiers.

Premièrement, il faut créer un disque vierge.

```
1 $ dd of=archlinuxarm.img bs=1 seek=4G count=0
```

Deuxièmement, introduire le système de fichiers `ext4` sur le disque vierge.

```
1 $ mkfs.ext4 -F archlinuxarm.img
```

Troisièmement, monter le disque virtuel dans l'ordinateur.

```
1 $ mkdir arm-chroot
2 $ mount archlinuxarm.img arm-chroot
```

Puis en dernier lieu, installer la dernière version d'ArchlinuxARM (version ARMv7) dans le disque virtuel. *Si vous souhaitez changer la version d'ArchlinuxARM, remplacer ci-dessous "armv7" par la version désirée. Liste des versions d'ArchlinuxARM disponible [ici](#) .*

```
1 $ wget http://archlinuxarm.org/os/ArchLinuxARM-armv7-latest.tar.gz
2 $ tar -xf ArchLinuxArm-armv7-latest.tar.gz -C arm-chroot
```

2. Installation des paquets requis

Pour la suite du guide nous avons besoin des paquets ci-dessous :

- `qemu-user-static` [AUR] : interpréter les instructions ARM dans the chroot
- `arch-install-scripts` : utilisation plus simple de chroot

J'utilise yaourt pour AUR mais vous pouvez aussi utiliser pacaur, pamac, ou autres.

```
1 $ yaourt -S qemu-user-static
2 $ pacman -S arch-install-scripts
```

3. Chroot avec QEMU

3. Chroot avec QEMU

Pour réaliser ce chroot nous devons dans un premier temps copier le fichier binaire `qemu-arm-static` dans le chroot (dossier bin du disque virtuel).

```
1 $ cp /usr/bin/qemu-arm-static arm-chroot/usr/bin
```

Dans un second temps nous devons enregistrer `qemu-arm-static` comme ARM interpréteur dans le kernel. Je conseille l'utilisation de `su` pour la prochaine commande plutôt que `sudo`.

```
1 $ echo ':arm:M::\x7fELF\x01\x01\x01\x00\x00\x00\x00\x00\x00\x00\x00\x00\x02\x00\x00\x00' > /proc/sys/fs/binfmt_misc/register
```

Si vous voulez créer votre propre code, il existe un outil pour vous faciliter la vie, [binfmt-manager](#) ↗

Dernière étape pour chroot ArchlinuxARM : utiliser le script `arch-chroot` qui provient du paquet `arch-install-scripts` qui va monter, bind tous ce qui est nécessaire.

```
1 $ arch-chroot arm-chroot /bin/bash
```

Une fois dans le chroot vous pourrez tester le système avec `uname`.

```
1 [chroot]$ uname -a
2 Linux saturn 3.14.6-1-ARCH #1 SMP PREEMPT Sun Jun 8 10:08:38 CEST
   2014 armv7l GNU/Linux
```

Cela dépend du système de fichiers que vous utilisez mais il peut être nécessaire de définir le `name server` avec l'adresse d'un DNS par exemple ceux de google `8.8.8.8` dans `/etc/resolv.conf` pour avoir la résolution des adresses internet.

Quand vous aurez fini, quitter le chroot et démonter le disque virtuel ou la carte SD.

```
1 [chroot]$ exit
2 $ umount arm-chroot
```

J'espère que cette petite tribune vous aura plus.

3. Chroot avec QEMU

Ce(tte) œuvre est mise à disposition selon les termes de la [Licence Creative Commons Attribution - Pas d'Utilisation Commerciale 4.0 International](#) ↗ .