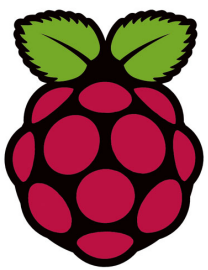
Atelier StarinuX Raspberry Pi

Partie 2

Pascal Bessonneau

12/2016



Démarrage

1. il va chercher un premier fichier bootcode sur la partition FAT32 qui est exécuté par le GPU
2. il va chercher un second fichier start_elf sur la partition FAT32 qui est exécuté par le GPU
3. le GPU va sonner le CPU pour qu'il démarre en passant le noyau avec les arguments qui vont bien
4. Le CPU se lance avec le kernel (noyau) et le démarrage devient celui de Linux :
   - chargement en mémoire du kernel
   - montage en lecture de la partition /
   - ...

Démarrage

Le démarrage se fait à l'aide du GPU et non du CPU.
Les fichiers de démarrage du GPU *bootcode* et *start_elf* sont propriétaires et sont distribués sous la forme de binaire.
Le démarrage ne se fait pas avec les outils GNU/Linux habituel : *grub* ou *lilo* comme sur un PC normal. Le démarrage va dépendre de la présence de cette fameuse partition fat32 au début de la carte.

Démarrage

Le fichier important sur un Raspberry est le fichier “config.txt”. En effet ce fichier est utilisé pour personnaliser le démarrage du Pi en ajoutant la caméra, modifier la mémoire réservé au gpu, ... On y trouve aussi plein de paramètres pour configurer la sortie HDMI si vous lancez le Pi avec une interface graphique.

Gestion de la mémoire

La mémoire d'un Raspberry est commune entre le GPU et le CPU. Selon l'usage que vous faites du Raspberry, par exemple utiliser ou non la caméra ou l'utiliser comme centre multimédia, il peut être intéressant de modifier la quantité de mémoire allouée au GPU. Le minimum pour le GPU est 16Mo. Il est nécessaire d'avoir 64Mo pour utiliser la caméra qui est aussi la valeur par défaut :

```
gpu_mem=16
```

Ici on règle la mémoire à 16Mo

Gestion de la mémoire

Les valeurs maximales pour la mémoire attribuée au GPU sont quant à elles de 448Mo pour 512Mo de mémoire et 944Mo pour 1024Mo de mémoire.
Il y a aussi la possibilité de définir la mémoire en fonction de la taille mémoire installée sur le Pi. La spécification de la mémoire en fonction de la taille mémoire installée écrase le paramètre fixé par "gpu_mem".

```
gpu_mem_1024=64
gpu_mem_512=32
gpu_mem_256=16
```

Gestion de la mémoire

Ici on fixe 64Mo pour une mémoire de 1Go (Raspberry Pi 3), 32Mo pour une Raspberry Pi2+, ... Ces paramètres sont utiles si vous personnalisez une carte pour l'installer sur différents Pi.
Les paramètres de "config.txt" pour la plupart ne sont pas nécessairement à taper à la main dans le fichier. En effet l'utilitaire "raspi-config" permet d'éditer le fichier via un menu plus convivial.

Première connexion

```
$ssh 192.168.0.56 -l pi
$raspberry (c'est le mot de passe par défaut)
The authenticity of host '192.168.0.56 (192.168.0.56)' can
ECDSA key fingerprint is SHA256:jxN8A+IwAD+axlznp4wLME8Tpi3
Are you sure you want to continue connecting (yes/no)?
$yes
Warning: Permanently added '192.168.0.56' (ECDSA) to the li
```

Connexion par clef

Vous pouvez le faire sur votre poste linux, utiliser putty-keygen ou le faire dans la pi.
Pour générer un clef si vous n'en avez pas déjà, il faut utiliser "ssh-keygen"

Connexion par clef

Pour activer la connexion par clef, il suffit de quelques modifications :

```
$mv .ssh/id_rsa.pub .ssh/authorized_keys
$vi .ssh/id_rsa
(copier la clef privée dans un
fichier ‘‘clef_raspberry’’ du répertoire .ssh)
$rm .ssh/id_rsa
```

Connexion par clef

Après il suffit de après de configurer ou de créer le fichier ".ssh/config' :

```
Host monpi
    Hostname 192.169.0.56
    User pi
    Port 22
    IdentityFile clef_raspberry
```

Vous pourrez vous connecter sans mot de passe simplement en tapant "ssh monpi".

Utilitaires

Pour simplifier les choses pour les débutants, les concepteurs de la Raspbian ont inclus deux utilitaires spécifique à la Pi dans la distribution.
Le premier outil est *raspi-config*. C'est un utilitaire clique-bouton (NCURSES en mode console) pour accéder à des fonctions spécifiques de la Pi ou pour lancer des commandes Linux.
L'autre outil est *rpi-update* qui permet de mettre à jour le firmware de la Pi. Il est à lancer régulièrement pour avoir son système à jour. Ce qui est différent des mises à jour de Raspbian.

raspi-config

Quand vous lancez un Rapsberry Pi, il est important de faire quelques réglages :

1. Changer le mot de passe
2. Internationalization Options, changer les locales (fr_FR.UTF-8 UTF-8)
3. Internationalization Options, changer le timezone (Paris)

raspi-config

Toutes ces commandes sont en fait des commandes debian mais elle sont plus simples à utiliser dans “raspi-config”.
Parmi les autres options, il y a l'activation de la caméra et dans les options avancées, il y a également des choses intéressantes :

Hostname Changer l'Hostname pour y accéder par ce nom plutôt que par l'ip

Memory Split la mémoire allouée au GPU

SPI utile pour activer la compatibilité avec des hat comme PiFace

...

rpi-update

Cet outil est lié au Raspberry, c'est pour récupérer et mettre à jour le firmware.
En plus des paquets, il est important de maintenir à jour son Pi y compris son firmware. Il faut éviter absolument de couper l'alimentation ou de couper la connexion SSH pendant la mise à jour du firmware.
La commande est très simple :

```
$rpi-update
```

Changement de mot de passe

La première étape est de changer le mot de passe par défaut. Il suffit de se connecter et de faire :

```
passwd
```

Choisissez un mot de passe le plus sécuritaire possible.

Ajouter un utilisateur

Vous pouvez aussi créer un utilisateur spécifique. Par exemple pour ajouter un utilisateur *pascal* :

```
adduser pascal
....
```

Pare-feu

Il faut veiller si vous bricolez avec votre pi de lui donner les mêmes droits que l'utilisateur pi. Pour cela, il faut éditer le fichier */etc/groups*.
Ce sont surtout le groupe *sudo*, *SPI* qui sont utiles car ils permettront de lancer des commandes en tant que *root* et d'avoir le contrôle sur les interfaces de la pi.

Pare-feu

Vous devez mettre un pare-feu. Surtout si vous utilisez le Pi comme serveur et donc qui est “face” à internet.
Attention, quelque soit le pare-feu, il faut laisser le port 22 ouvert pour vous connecter avec SSH sinon votre Raspberry Pi deviendra une brique... Il faut ouvrir le port et lancer le pare-feu au cours d'une session SSH et se connecter dans une nouvelle connexion pour vérifier que le port est bien ouvert !

Sauvegarde d'un pare-feu vide

Utile si vous jouez avec iptables. Pour sauvegarder un pare-feu vide :

```
$sudo iptables-save ~/iptables.empty
```

Sauvegarde d'un pare-feu vide

Pour l'utiliser il faut faire :

```
sudo iptables-restore ~/iptables.empty
```

Pare-feu à la main

Vous pouvez faire un pare-feu simple “à la main” avec *iptables* :

```
#!/bin/sh

IPTABLES=/sbin/iptables

# Creation des tables
$IPTABLES -N TCP
$IPTABLES -N UDP
...
```

Pare-feu à la main

Attention tous les fichiers du firewall doivent être accessibles seulement par le root.

```
$sudo chmod 700 02-firewall
$sudo chown root:root 02-firewall
```

Pare-feu à la main

Le fichier du firewall est à mettre dans le répertoire “/etc/network/if-pre-up.d”. Il sera exécuter dès que le réseau va devenir accessible. Vous pouvez aussi le placer en fin de lancement de Linux : dans le fichier “/etc/rc.local”.

arno-iptables-firewall

Sinon il y a un paquet très efficace et pratique : *arno-iptables-firewall*. C'est un script BASH qui fait un pare-feu de bonne facture. Il faut indiquer les ports à laisser ouvert lors de l'installation. Il gère aussi des règles plus complexes : NAT, DMZ, ports ouverts sur le LAN et pas sur internet, ...
Pour l'installer, c'est comme pour n'importe quel paquet Debian :

```
$sudo apt-get install arno-iptables-firewall
```

arno-iptables-firewall

Pour stopper le pare-feu, relancer et lancer le pare-feu les commandes sont respectivement :

```
$sudo service arno-iptables-firewall stop
$sudo service arno-iptables-firewall restart
$sudo service arno-iptables-firewall start
```

Mises à jour

Il faut maintenir votre Raspberry Pi à jour... Pour tout ce qui est fourni par les paquets Debian, il suffit de faire de temps en temps un petit :

```
$sudo apt-get update
$sudo apt-get upgrade -y
$sudo apt-get dist-upgrade -y
```

Si la mise à jour contient une mise à jour du noyau, il faudra redémarrer le Pi.

Mises à jour

Comme vu précédemment il y a aussi le firmware à mettre à jour avec la commande spéciale :

```
$sudo rpi-update
```

Mises à jour

Le mieux est de créer un alias dans le fichier *.bashrc* :

```
alias update=sudo rpi-update && sudo apt-get update && \
sudo apt-get upgrade -y && sudo apt-get dist-upgrade -y
```

Mises à jour

Si vous avez installer un serveur web avec WordPress ou Yunohost il vous faudra aussi mettre à jour les composants en plus.

Blocage de SSH par mot de passe

Pour éviter les connexions qui utilisent un mot de passe au lieu de la clef privé/clef publique, il faut changer cette ligne (c'est la dernière ligne) dans "/etc/ssh/sshd_config" :

```
UsePAM no
```

Blocage de SSH par mot de passe

Il faut aussi empêcher les connexions en tant que root (c'est la ligne 28) :

```
PermitRootLogin no
```

Puis il faut redémarrer le serveur :

```
$sudo service ssh restart
```