



Table des matières

1) Téléchargement et installation de RaspiOS sur la carte microSD :.....	1
2) Paramétrage du système «à chaud» :.....	1
3) Paramétrage du système «à froid» avec un «chroot» :.....	1
4) Émulation du système avec QEMU :.....	1

1) Téléchargement et installation de RaspiOS sur la carte microSD :

a) Sur ma machine j'insère le lecteur de carte micro sd et j'y insère ma carte micro sd

Je fais la commande suivante pour déterminer le périphérique bloc correspondant à ma micro sd :

```
test@202-15:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0       7:0      0   55,3M  1 loop /snap/core18/1885
loop1       7:1      0   69,2M  1 loop /snap/lxd/18137
loop2       7:2      0  219,1M  1 loop /snap/multipass/2772
loop3       7:3      0  219,2M  1 loop /snap/multipass/2886
loop4       7:4      0   97,8M  1 loop /snap/core/10185
loop5       7:5      0   55,4M  1 loop /snap/core18/1932
loop6       7:6      0   69,4M  1 loop /snap/lxd/18324
loop7       7:7      0   33,3M  1 loop /snap/multipass-gui/16
loop8       7:8      0   97,7M  1 loop /snap/core/10126
sda         8:0      0  232,9G  0 disk
├─sda1      8:1      0    512M  0 part /boot/efi
├─sda2      8:2      0  224,5G  0 part /
└─sda3      8:3      0    7,9G  0 part [SWAP]
sdb         8:16     1    7,2G  0 disk
└─sdb1      8:17     1    7,2G  0 part /media/test/COOKIES
sdc         8:32     1   14,7G  0 disk
├─sdc1      8:33     1    256M  0 part /media/test/boot
└─sdc2      8:34     1   14,5G  0 part /media/test/rootfs
sr0        11:0     1  1024M  0 rom
```

b) Je télécharge maintenant la dernière version de raspbian :

```
test@202-15:~$ wget --content-disposition \
https://downloads.raspberrypi.org/raspbian_lite_armhf_latest
```

c) Je vérifie le sha256sum de mon fichier sur le site de raspberry :

```
test@202-15:~$ sha256sum *raspbian*.zip
4522df4a29f9aac4b0166fbfee9f599dab55a997c855702bfe35329c13334668
2020-08-20-raspbian-buster-armhf-lite.zip
```

Sur le site :

Raspberry Pi OS Lite

Release date: August 20th 2020

Kernel version: 5.4

Size: 432MB

[Show SHA256 file integrity hash:](#)

```
4522df4a29f9aac4b0166fbfee9f599
dab55a997c855702bfe35329c133346
68
```

Le sha256sum est bien identique

d) Je me met en tant que root et je copie directement l'image compressée sur le périphérique :

```
root@202-15:/home/test# unzip -p *raspbian*.zip | dd of=/dev/sdc
bs=4M conv=fsync
0+26276 enregistrements lus
0+26276 enregistrements écrits
1845493760 octets (1,8 GB, 1,7 GiB) copiés, 68,4164 s, 27,0 MB/s
```

Pour voir le temps qu'il faut à la copie on peut mettre time juste avant la commande unzip

e) J'enlève la carte micro sd de la clé puis je l'insère dans le raspberry Pi

Je branche les périphériques nécessaires à mon raspberry (clavier, souris, écran)

Je fini en alimentant le raspberry

Les identifiants sont pi et raspberry

f) Je regarde le nombres de paquets qui sont installés de base sur la raspi :

```
sudo dpkg -l |wc -l  
480
```

g) Je regarde les services activés de base sur le raspi :

```
sudo service --status-all  
alsa-utils  
avahi-daemon  
bluetooth  
console-setup.sh  
cron  
dbus  
dhcpcd  
dphys-swapfile  
fake-hwclock  
keyboard-setup.sh  
kmod  
networking  
nfs-common  
paxctld  
procps  
raspi-config  
rng-tools  
rpcbind  
rsync  
rsyslog  
ssh  
sudo  
triggerhappy  
udev
```

h) Pour donner la configuration actuelle des interfaces réseau je fais la commande "ip a"

Je vois que j'ai 3 interfaces :

- loopback
- eth0
- wlan0

2) Paramétrage du système «à chaud» :

a) Pour changer le nom par défaut du raspi je vais dans le fichier /etc/hostname et je modifie la ligne suivante :

```
pi202-14
```

Ensuite je vais dans le fichier /etc/hosts et je rajoute la ligne suivante :

```
127.0.1.1    pi202-14
127.0.1.1    raspberrypi
```

Maintenant je reboot le raspi pour prendre en compte les changements :

```
reboot
```

b) Les utilisateurs existants sur le Rpi sont pi et root (voir dans le fichier /etc/passwd les utilisateurs qui sont dans /bin/bash)

root appartient au group root et pi appartient à plusieurs groupes :

```
cat /etc/group |grep pi
adm
dialout
cdrom
sudo
audio
video
plugdev
games
users
input
netdev
spi
i2c
gpio
```

c) Pour paramétrer le système et mettre la clavier en français je fais la commande suivante :

```
sudo raspi-config
```

On va dans l'onglet numéro 4 et on change la langue du clavier

d) Pour changer l'utilisateur de pi et le passer à ido je commence par créer un mot de passe à l'utilisateur root :

```
sudo passwd root
```

Maintenant je ferme la connexion avec l'utilisateur pi (CTRL+D) et je me connecte sur root

Ensuite je peux changer le nom d'utilisateur de pi :

```
usermod -l ido pi
```

Maintenant je peux retourner sur mon utilisateur ido et changer son mot de passe

e) Je branche mon raspi au réseau de l'iut avec un câble ethernet

f) Son adresse IP attribuée est 10.202.0.165. On peut la retrouver en faisant une capture de trames au moment où l'on branche le câble ethernet car il y a une demande d'IP sur le serveur dhcp

g) Pour activer une adresse IP fixe sur le Pi on va éditer le fichier /etc/dhcpd.conf :

h) Pour activer le service ssh je fais la commande suivante :

```
sudo service ssh start
```

Pour activer le service ssh au démarrage du Rpi je fais la commande suivante :

```
sudo systemctl enable ssh
```

Maintenant je dois pouvoir ssh mon raspi :

```
test@202-15:~$ ssh idol@10.202.0.165
idol@10.202.0.165's password:
Linux pi202-14 5.4.51-v7+ #1333 SMP Mon Aug 10 16:45:19 BST 2020
armv7l

The programs included with the Debian GNU/Linux system are free
software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Aug 20 14:31:08 2020

Wi-Fi is currently blocked by rfkill.
Use raspi-config to set the country before use.
```

3) Paramétrage du système «à froid» avec un «chroot» :

a) Je commence par installer les paquets nécessaires au chroot :

```
test@202-15:~$ sudo apt install qemu qemu-user-static binfmt-support
```

b) Depuis mon poste Linux j'affiche le contenu du fichier image de base :

```
root@202-15:/home/test# unzip *raspios*.zip
Archive: 2020-08-20-raspios-buster-armhf-lite.zip
  inflating: 2020-08-20-raspios-buster-armhf-lite.img
root@202-15:/home/test# fdisk -l *raspios*.img
Disque 2020-08-20-raspios-buster-armhf-lite.img : 1,7 GiB,
1845493760 octets, 3604480 secteurs
Unités : secteur de 1 × 512 = 512 octets
Taille de secteur (logique / physique) : 512 octets / 512 octets
taille d'E/S (minimale / optimale) : 512 octets / 512 octets
Type d'étiquette de disque : dos
Identifiant de disque : 0x907af7d0
```

Périphérique	Amorçage	Début	Fin
Secteurs Taille Id Type			
2020-08-20-raspios-buster-armhf-lite.img1		8192	532479
524288 256M c W95			
2020-08-20-raspios-buster-armhf-lite.img2		532480	3604479
3072000 1,5G 83 Linux			

c) Je renomme le fichier pour indiquer qu'il s'agit d'une copie :

```
root@202-15:/home/test# mv *raspios*.img rpi-15.img
root@202-15:/home/test# ls
rpi-15.img
```

d) Je charge l'image sur un périphérique de bouclage :

Je vérifie avant lesquels sont libres :

```
root@202-15:/home/test# losetup -f
/dev/loop9
root@202-15:/home/test# losetup -P /dev/loop9 rpi-15.img
```

Je vérifie maintenant qu'il possède bien 2 partitions :

```
root@202-15:/home/test# fdisk -l /dev/loop9
Disque /dev/loop9 : 1,7 GiB, 1845493760 octets, 3604480 secteurs
Unités : secteur de 1 × 512 = 512 octets
Taille de secteur (logique / physique) : 512 octets / 512 octets
taille d'E/S (minimale / optimale) : 512 octets / 512 octets
Type d'étiquette de disque : dos
Identifiant de disque : 0x907af7d0

Périphérique Amorçage Début Fin Secteurs Taille Id Type
/dev/loop9p1 8192 532479 524288 256M c W95 FAT32
(LBA)
/dev/loop9p2 532480 3604479 3072000 1,5G 83 Linux
```

e) Je monte en premier la partition 2 (celle du système de fichier) puis la partition 1 (celle du boot) dans le répertoire boot :

```
root@202-15:/home/test# mkdir /mnt/rpi
root@202-15:/home/test# mount /dev/loop9p2 /mnt/rpi
root@202-15:/home/test# mount /dev/loop9p1 /mnt/rpi/boot/
```

f) Je modifie le fichier boot/cmdline.txt et je le met de la façon suivante pour mettre une ip fixe (tout le fichier est sur une seule ligne) :

```
ip=10.202.15.3:10.255.255.200:10.202.255.254:255.255.0.0
```

Pour activer le ssh il faut simplement créer un fichier ssh dans /mnt/rpi/boot/

g) Je modifie les fichiers dans /etc pour fixer le nom du système :

Fichier hostname :

```
sam
```

Fichier hosts :

```
127.0.0.1 localhost
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

127.0.1.1 sam
```

J'ajoute aussi le support du clavier en français (fichier /etc/default/keyboard) :

```
# KEYBOARD CONFIGURATION FILE

# Consult the keyboard(5) manual page.

XKBMODEL="pc105"
XKBLAYOUT="fr"
XKBVARIANT=""
XKBOPTIONS=""

BACKSPACE="guess"
```

Je fini en sécurisant les identifiants utilisateurs :

Pour sécuriser les identifiants il faut mettre le mot de passe dans le fichier /etc/shadow :

```
pi:
$6$TcStb3ADs0Vmh11P$H0z0afl84EFN0Ws4IFhEtgW6iFt3VbmI2u4Q6WsAy9IPeX
iJRJ0v0hBPzGRNE0rR.gdI6jWe7Nmy8Ub1ZSRHY/:18494:0:99999:7:::
```

Mon mot de passe est bien sécurisé (hashé) et dans passwd le mot de passe n'est pas visible en clair et si on veut changer le mot de passe il faut connaître le hash correspondant au mot de passe que l'on veut mettre

h) Je monte les dossiers spéciaux du système :

```
root@202-15:/mnt/rpi/etc# mount --bind /proc /mnt/rpi/proc/
root@202-15:/mnt/rpi/etc# mount --bind /sys /mnt/rpi/sys/
root@202-15:/mnt/rpi/etc# mount --bind /dev /mnt/rpi/dev/
root@202-15:/mnt/rpi/etc# mount --bind /dev/pts /mnt/rpi/dev/pts/
```

i) Je copie le programme d'émulation qemu dans l'image :

```
root@202-15:/mnt/rpi/etc# cp /usr/bin/qemu-arm-static
/mnt/rpi/usr/bin/
```

j) Je commente les lignes dans le fichier /mnt/rpi/etc/ls.so.preload pour éviter les messages d'erreurs dans la suite :

```
#/usr/lib/arm-linux-gnueabi/hf/libarmmem-${PLATFORM}.so
```

k) Je lance mon cmd avec une nouvelle racine :

```
root@202-15:/mnt/rpi/etc# chroot /mnt/rpi /bin/bash
root@202-15:/#
```


l) Maintenant toutes les commandes que je passerais seront faites comme si je me trouvais dans le raspberry Pi

Je test en mettant à jour tous les paquets du système :

```
root@202-15:/# apt update
Get:1 http://raspbian.raspberrypi.org/raspbian buster InRelease
[15.0 kB]
Get:2 http://archive.raspberrypi.org/debian buster InRelease [32.6
kB]
Get:3 http://raspbian.raspberrypi.org/raspbian buster/main armhf
Packages [13.0 MB]
Get:4 http://raspbian.raspberrypi.org/raspbian buster/contrib
armhf Packages [58.7 kB]
Get:5 http://archive.raspberrypi.org/debian buster/main armhf
Packages [336 kB]
Fetched 13.4 MB in 9s (1567 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
31 packages can be upgraded. Run 'apt list --upgradable' to see
them.
```

Je n'ai plus qu'à mettre à jour les paquets avec apt upgrade

m) Je commence par libérer un peu d'espace disque :

```
root@202-15:/# rm /var/cache/apt/srcpkgcache.bin
```

Je peux installer quelques paquets (ici je n'installe que hostapd car j'ai très peu d'espace disque) :

```
root@202-15:/# apt install hostapd
```

n) Pour désactiver un service afin qu'il ne soit pas lancé au démarrage je fais la commande suivante :

```
root@202-15:/# systemctl disable dhcpcd
```

o) Je peux maintenant sortir du chroot (exit) et sortir du répertoire /mnt et je démonte tout puis je ferme le périphérique de bouclage :

```
root@202-15:/mnt/rpi/etc# cd
root@202-15:~# umount /mnt/rpi/{dev/pts,dev,sys,proc,boot,}
umount: /mnt/rpi/dev/pts: non monté.
umount: /mnt/rpi/dev: non monté.
umount: /mnt/rpi/sys: non monté.
```

```
umount: /mnt/rpi/proc: non monté.  
umount: /mnt/rpi/boot: non monté.  
  
root@202-15:~# losetup -d /dev/loop9  
root@202-15:~# losetup -a  
/dev/loop1: [2050]:9330755 (/var/lib/snapd/snaps/lxd_18137.snap)  
/dev/loop8: [2050]:9328897 (/var/lib/snapd/snaps/core_10126.snap)  
/dev/loop6: [2050]:9330758 (/var/lib/snapd/snaps/lxd_18324.snap)  
/dev/loop4: [2050]:9330747 (/var/lib/snapd/snaps/core_10185.snap)  
/dev/loop2: [2050]:9326050  
(/var/lib/snapd/snaps/multipass_2772.snap)  
/dev/loop0: [2050]:9328500 (/var/lib/snapd/snaps/core18_1885.snap)  
/dev/loop7: [2050]:9330509 (/var/lib/snapd/snaps/multipass-  
gui_16.snap)  
/dev/loop5: [2050]:9330751 (/var/lib/snapd/snaps/core18_1932.snap)  
/dev/loop3: [2050]:9330507  
(/var/lib/snapd/snaps/multipass_2886.snap)
```

Je vois bien que mon périphérique de bouclage est fermé

p) Je copie la nouvelle version sur la carte micro sd et je la teste sur mon raspi sans écran et avec un câble ethernet :

Avant de copier je dois démonter les partitions dans sdc (qui contient le boot et le rootfs de ma carte micro sd) :

```
test@202-15:~$ lsblk  
test@202-15:~$ sudo umount /dev/sdc1  
test@202-15:~$ sudo umount /dev/sdc2
```

Je peux maintenant copier mon fichier image dans la carte micro sd :

```
root@202-15:/home/test# cat rpi-15.img |dd of=/dev/sdc bs=4M  
conv=fsync
```

Je remet la micro sd dans le raspi et je l'alimente. Une fois alimenté je dois pouvoir ssh le raspi avec l'adresse ip mise dans le fichier précédemment

```
test@202-15:~$ ssh pi@10.202.15.3  
pi@10.202.15.3's password:  
pi@sam:~ $
```

4) Émulation du système avec QEMU :

a) J'installe le paquet suivant :

```
root@202-15:/home/test# apt install qemu-system-arm
```

b) Je charge l'image sur un périphérique de bouclage et je monte la partition 2 :

```
root@202-15:/home/test# losetup -P /dev/loop9 rpi-15.img
root@202-15:/home/test# mkdir /mnt/rpi2
root@202-15:/home/test# mount /dev/loop9p2 /mnt/rpi2/
```

c) Je crée le fichier 90-qemu.rules avec les commandes suivantes :

```
MNT=/mnt/rpi2
cat << 'EOF' >> $MNT/etc/udev/rules.d/90-qemu.rules
KERNEL=="sda", SYMLINK+="mmcblk0"
KERNEL=="sda?", SYMLINK+="mmcblk0p%n"
KERNEL=="sda2", SYMLINK+="root"
EOF
```

Si je vais voir dans le fichier je dois retrouver les 3 KERNEL :

```
root@202-15:/home/test# cat $MNT/etc/udev/rules.d/90-qemu.rules
KERNEL=="sda", SYMLINK+="mmcblk0"
KERNEL=="sda?", SYMLINK+="mmcblk0p%n"
KERNEL=="sda2", SYMLINK+="root"
```

d) Je démonte la partition et je ferme le périphérique de bouclage :

```
root@202-15:/home/test# umount /mnt/rpi2
root@202-15:/home/test# losetup -d /dev/loop9
```

e) La version de mon kernel je peux le voir au démarrage de mon image (ici elle est 5.4.51)

Je la télécharge donc :

```
root@202-15:/home/test# git clone
https://github.com/dhruvvyas90/qemu-rpi-kernel
```

f) Je n'ai plus qu'à lancer l'émulation de mon raspiOS :

```
root@202-15:/home/test/qemu-rpi-kernel# qemu-system-arm -M
versatilepb -dtb versatile-pb-buster.dtb -cpu arm1176 -m 256 -
drive "file=/home/test/rpi-15.img,index=0,media=disk,format=raw" -
kernel kernel-qemu-4.19.50-buster -serial stdio -append
```

```
'root=/dev/sda2 panic=1 rootfstype=ext4 rw' -net nic -net  
user,hostfwd=tcp::2222-:22 -no-reboot
```