

Comment convertir la partition racine Raspbian en btrfs

Table of Contents

- [Préparer les initramfs](#)
- [Convertir le rootfs avec btrfs-convert](#)
- [Déplacer le rootfs sur une nouvelle partition](#)

Le noyau Raspbian n'inclut pas le support btrfs par défaut; les étapes de démarrage initiales s'exécutent normalement, mais lorsque le noyau se charge, il ne voit aucun système de fichiers qu'il pourrait monter - et panique. Une solution existe: ajouter btrfs comme module noyau, dans initramfs.

Préparer les initramfs

Installer les packages requis - le module du noyau et les outils pour mettre à jour initramfs avec:

```
sudo apt install btrfs-tools initramfs-tools
```

Dire à initramfs de charger le module btrfs (cela devrait se produire automatiquement mais ce n'est pas toujours le cas) en ajoutant une ligne avec "btrfs" à la liste des modules requis:

```
echo 'btrfs' | sudo tee -a /etc/initramfs-tools/modules
```

Modifier /etc/default/raspberrypi-kernel et décommenter la ligne INITRD=yes pour activer la génération de fichiers /boot/initrd.img-KVER (nécessite initramfs-tools).



Afin d'éviter que la mise à niveau du noyau ne brise le système:

- supprimer /etc/kernel/postinst.d/initramfs-tools
- télécharger **update-rpi-initramfs** pour des mises à jour manuelles plus simples des initramfs.

Utiliser le hook /usr/share/initramfs-tools/hooks/btrfs¹⁾ et le script /usr/share/initramfs-tools/scripts/local-premount/btrfs²⁾ - les valeurs par défaut sont fournies, mais elles ne sont pas utilisées de manière automatique, il faut donc les copier dans /etc:

```
sudo mkdir -p /etc/initramfs-tools/hooks
sudo mkdir -p /etc/initramfs-tools/scripts/local-premount
sudo cp /usr/share/initramfs-tools/hooks/btrfs /etc/initramfs-tools/hooks
sudo cp /usr/share/initramfs-tools/scripts/local-premount/btrfs
/etc/initramfs-tools/scripts/local-premount
```

```
sudo chmod +x /etc/initramfs-tools/hooks/btrfs /etc/initramfs-
tools/scripts/local-premount/btrfs
```

Créer (-c) le nouveau initramfs:

```
update-initramfs -c -k $(uname -r)
```

Modifier `/boot/config.txt` pour utiliser cet initramfs, en ajoutant `initramfs <nom-initrafs> followkernel` Le nom de fichier `<nom-initrafs>` est celui généré à l'étape précédente; "followkernel" contrôle l'emplacement en mémoire:

```
echo "initramfs intrd.img-$(uname -r)" | sudo tee -a /boot/config.txt
```

A ce stade, on a un système qui pourrait utiliser btrfs comme périphérique racine. Tester par redémarrage: le système démarrera toujours à partir de la partition ext4 (ou de tout ce qui se trouve dans `/boot/cmdline.txt`), mais `dmesg | grep -i btrfs` devrait maintenant afficher une ligne contenant "Btrfs chargé".

Si on rencontre des problèmes, un nom de fichier est probablement erroné et on doit toujours pouvoir récupérer. Sinon : il suffit de recommencer à zéro - à ce stade, rien n'est vraiment perdu.

Convertir le rootfs avec btrfs-convert

Insérer la carte SD dans un autre ordinateur et identifier le périphérique (par exemple `/dev/mmcblk0`):

```
$ fdisk -l /dev/mmcblk0
Disk /dev/mmcblk0: 7.48 GiB, 8017412096 bytes, 15659008 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x4301c17b
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/mmcblk0p1		8192	532480	524289	256M	c	W95 FAT32 (LBA)
/dev/mmcblk0p2		540672	15659007	15118336	7.2G	83	Linux

Pour convertir le système de fichiers en btrfs, procéder maintenant comme suit :

- Facultatif : s'assurer que le rootfs est propre (**fsck**)
- Convertir ext4 en btrfs en utilisant **btrfs-convert**
- Monter la nouvelle racine btrfs
- Modifier `/etc/fstab`
- Modifier `/boot/cmdline.txt`

```
fsck.ext4 /dev/mmcblk0p2 # Facultatif mais recommandé
btrfs-convert /dev/mmcblk0p2 # Convertit la racine ext4 en btrfs
mkdir /mnt/sdcard # Ici on va monter le rootfs
mount -t btrfs /dev/mmcblk0p2 /mnt/sdcard # Monter le rootfs nouvellement
créé
```

Maintenant, éditer `/etc/fstab` et changer **ext4** en **btrfs**. Il faut également désactiver la vérification du système de fichiers en définissant les deux derniers chiffres de la ligne btrfs sur **0**.

```
$ vi /mnt/sdcard/etc/fstab
proc                /proc              proc      defaults        0          0
/dev/mmcblk0p1      /boot              vfat      defaults        0          2
/dev/mmcblk0p2      /                  btrfs     defaults,noatime 0          0
```



btrfs-convert ne change pas l'UUID, mais il est préférable de contrôler cela avec `sudo lsblk -f`

Trouver l'UUID de la partition btrfs avec **blkid** - récupérer la partie UUID =(pas UUID_SUB, pas PARTUUID! Cela déclencherait un bogue dans le chargeur de démarrage et le noyau ne démarrerait pas.):

```
sudo blkid

/dev/mmcblk0p2: UUID = "cafebeef-0000-1234-aaaa-12346589" UUID_SUB =
"ababccdd-2345-cafe-beee-587989991110" TYPE = "btrfs" PARTUUID =
"beef0bee-02"
```

Enfin, éditer `/boot/cmdline.txt` à partir de la carte SD (il faut remplacer **rootfstype=ext4** par **rootfstype=btrfs** et définir **fsck.repair=no**):

```
# vi /mnt/sdcard/boot/cmdline.txt
dwc_otg.lpm_enable=0 console=serial0,115200 console=tty1
root=UUID=cafebeef-0000-1234-aaaa-12346589 rootfstype=btrfs
elevator=deadline fsck.repair=no rootwait
```



Fsck cause beaucoup de problèmes. En cas d'erreurs de montage non valides, vérifier si **fsck** a été désactivé dans `/etc/fstab` (le dernier zéro) et dans `/boot/cmdline.txt`

Rebooter le système:

```
sudo reboot
```



Après une mise à jour du noyau, il faut réexécuter **update-initramfs**, il est préférable de ne faire que des mises à jour manuelles du noyau (même en tant que mises à jour de sécurité), sinon Raspi ne redémarrera pas.

Déplacer le rootfs sur une nouvelle partition

Faire une sauvegarde de la /partition (ext4) - en supposant qu'il s'agit de /dev/mmcblk0p2, arrêter le RPi, retirer la carte SD et la monter sur un ordinateur Linux:

```
sudo mount /dev/mmcblk0p2 /mnt
```

Archiver le contenu; (il faut utiliser un outil qui préserve la propriété et les autorisations, par exemple **tar**):

```
cd /mnt  
sudo tar -czvf ~/rpi-rootfs-backup.tgz *
```

Démonter à nouveau la carte SD

Créer une partition btrfs quelque part - par exemple réutiliser la carte SD, en remplaçant la partition ext4 (/dev/mmcblk0p2);

```
mkfs.btrfs /dev/mmcblk0p2
```

Monter la partition btrfs et y restaurer la sauvegarde:

```
sudo mount /dev/mmcblk0p2 /mnt  
cd /mnt  
tar -xzf ~/rpi-rootfs-backup.tgz
```

Editer fstab sur la partition btrfs :

```
sudo vi /mnt/etc/fstab
```

Il devrait y avoir une ligne similaire à celle-ci:

```
/dev/mmcblk0p2 / ext4 foo,bar,baz 0 1
```

Il faut la changer en ceci (le nouveau type FS est btrfs, et il utilise les options par défaut):

```
/dev/mmcblk0p2 / btrfs defaults 0 1
```

Démonter la partition, mais ne pas retirer encore la carte SD!

```
sudo umount /mnt
```

Il faut dire au RPi qu'il va démarrer à partir de btrfs

Trouver l'UUID de la nouvelle partition btrfs avec **blkid** - récupérer la partie UUID =, (pas UUID_SUB, pas PARTUUID! Cela déclencherait un bogue dans le chargeur de démarrage et le noyau ne démarrerait pas.):

```
sudo blkid

/dev/mmcblk0p2: UUID = "cafebeef-0000-1234-aaaa-12346589" UUID_SUB =
"ababccdd-2345-cafe-beee-587989991110" TYPE = "btrfs" PARTUUID =
"beef0bee-02"
```

Monter la partition de démarrage (FAT32):

```
sudo mount /dev/mmcblk0p1 /mnt
```

Enfin, éditer /mnt/cmdline.txt à partir de la carte SD (il faut remplacer **rootfstype=ext4** par **rootfstype=btrfs** et définir **fsck.repair=no**):

```
# vi /mnt/cmdline.txt
dwc_otg.lpm_enable=0 console=serial0,115200 console=tty1
root=UUID=cafebeef-0000-1234-aaaa-12346589 rootfstype=btrfs
elevator=deadline fsck.repair=no rootwait
```

L'UUID est celui récupéré plus tôt, juste sans guillemets.

Démonter la partition de démarrage RPi:

```
sudo umount /mnt
```

Replacer la carte SD dans RPi et démarrer.

1)

Les **hooks** iniramfs sont utilisés lorsqu'une image initramfs est créée, ils peuvent être trouvés à deux endroits : /usr/share/initramfs-tools/hooks et /etc/initramfs-tools/hooks. Ils sont exécutés lors de la génération de l'image initramfs et sont responsable d'inclure tous les composants nécessaires dans l'image elle-même.

2)

Les **Scripts de démarrage** sont inclus dans l'image initramfs et normalement exécutés lors du démarrage du noyau dans l'espace utilisateur précoce avant le montage de la partition racine. ils peuvent être trouvés à deux endroits : /usr/share/initramfs-tools/scripts et /etc/initramfs-tools/scripts

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=howto:rpi-btrfs-rootfs>

Last update: **2025/02/19 10:59**

