

Memoires

Partitionnement et manipulation de mémoires

Mickaël Tansorier

Présentation du fonctionnement Globale de mémoire.

Plan

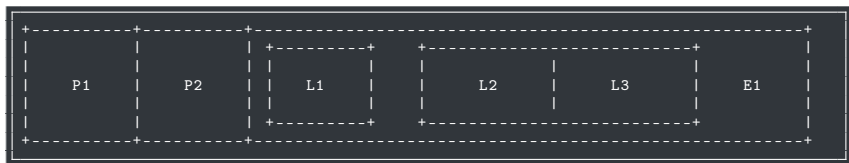
- 1 Introduction
- 2 Le types, les tables et le format de partitions
- 3 Les outils
- 4 U-Boot env

Si l'on veut scripter les créations/modifications de partitions, il faut connaître :

- Comment fonctionne les tables de partition
- Le type de partitionnement
- le Format de partitionnement
- Les outils de partitionnement

Il existe trois types de partitions :

- **primary** — est utiliser comme partition de boot (pour windows)
- **extended** — sert à abriter de multiple partitions logique
- **logical** — sert à abriter les fichier non relatif au système (audio, vidéo, ...)



Le type de table de partition définit l'entête du block device.

On le retrouve sous l'appellation label sous parted.

Il en existe de plusieurs types : aix, amiga, bsd, dvh, gpt, loop, mac, msdos, pc98, ou sun.

En règle générale on se trouve en MBR (Master boot record) (ou msdos) car c'est le plus courant pour des systèmes simplistes.

Donc dans le doute c'est celui-là qu'il faut choisir... :)

En effet GPT est son "successeur" et s'adresse à l'utilisation de boot en UEFI.

Exemple de différences

Taille de partitions :

- Avec `msdos` il est impossible de créer une partition supérieur à 2.2 To (2^{41} octets) sur des secteur de bloc de 512 byte.
- Avec GPT (GUID Partition Table) il est possible de créer une partition allant jusqu'à 9,4 Zo ($9,4 \times 10^{21}$ octets)

Nombre de block :

- Avec GTP il est possible de créer jusqu'à 128 block logique
- Avec `msdos` on peut aller seulement jusqu'à 4 block

Du fait de ses limitations le système de partitions MBR est remplacé la plupart du temps depuis 2013 par le système GPT.

Le type de partition est défini par un numéro d'id.
Ce numéro se trouve en entête de partition.

| | | | | | | | |
|----|-----------------|----|-----------------|----|-----------------|----|-----------------|
| 0 | Empty | 24 | NEC DOS | 81 | Minix / old Lin | bf | Solaris |
| 1 | FAT12 | 27 | Hidden NTFS Win | 82 | Linux swap / So | c1 | DRDOS/sec (FAT- |
| 2 | XENIX root | 39 | Plan 9 | 83 | Linux | c4 | DRDOS/sec (FAT- |
| 3 | XENIX usr | 3c | PartitionMagic | 84 | OS/2 hidden or | c6 | DRDOS/sec (FAT- |
| 4 | FAT16 <32M | 40 | Venix 80286 | 85 | Linux extended | c7 | Syrinx |
| 5 | Extended | 41 | PPC PREP Boot | 86 | NTFS volume set | da | Non-FS data |
| 6 | FAT16 | 42 | SFS | 87 | NTFS volume set | db | CP/M / CTOS / . |
| 7 | HPFS/NTFS/exFAT | 4d | QNX4.x | 88 | Linux plaintext | de | Dell Utility |
| 8 | AIX | 4e | QNX4.x 2nd part | 8e | Linux LVM | df | BootIt |
| 9 | AIX bootable | 4f | QNX4.x 3rd part | 93 | Amoeba | e1 | DOS access |
| a | OS/2 Boot Manag | 50 | OnTrack DM | 94 | Amoeba BBT | e3 | DOS R/O |
| b | W95 FAT32 | 51 | OnTrack DM6 Aux | 9f | BSD/OS | e4 | SpeedStor |
| c | W95 FAT32 (LBA) | 52 | CP/M | a0 | IBM Thinkpad hi | ea | Rufus alignment |
| e | W95 FAT16 (LBA) | 53 | OnTrack DM6 Aux | a5 | FreeBSD | eb | BeOS fs |
| f | W95 Ext'd (LBA) | 54 | OnTrackDM6 | a6 | OpenBSD | ee | GPT |
| 10 | OPUS | 55 | EZ-Drive | a7 | NeXTSTEP | ef | EFI (FAT-12/16/ |
| 11 | Hidden FAT12 | 56 | Golden Bow | a8 | Darwin UFS | f0 | Linux/PA-RISC b |
| 12 | Compaq diagnost | 5c | Priam Edisk | a9 | NetBSD | f1 | SpeedStor |
| 14 | Hidden FAT16 <3 | 61 | SpeedStor | ab | Darwin boot | f4 | SpeedStor |
| 16 | Hidden FAT16 | 63 | GNU HURD or Sys | af | HFS / HFS+ | f2 | DOS secondary |
| 17 | Hidden HPFS/NTF | 64 | Novell Netware | b7 | BSDI fs | fb | VMware VMFS |
| 18 | AST SmartSleep | 65 | Novell Netware | b8 | BSDI swap | fc | VMware VMKCORE |
| 1b | Hidden W95 FAT3 | 70 | DiskSecure Mult | bb | Boot Wizard hid | fd | Linux raid auto |
| 1c | Hidden W95 FAT3 | 75 | PC/IX | bc | Acronis FAT32 L | fe | LANstep |
| 1e | Hidden W95 FAT1 | 80 | Old Minix | be | Solaris boot | ff | BBT {frame} |

Les outils

Les outils classique sous Linux :

- parted
- fdisk
- sfdisk

Objectif

Est de rendre scriptable un repartitionnement.

Donc si vous pensez au confort de `gparted` vous pouvez oublier...

mkpart

Exemple de commande :

```
$ parted -a optimal /dev/usb mkpart primary 0% 4096MB
```

mkpart

Exemple de commande :

```
$ parted -a optimal /dev/usb mkpart primary 0% 4096MB
```

L'option `-a` (ou `--align`) peut prendre plusieurs valeurs :

- `none` — Utilise l'alignement le plus petit autorisé par le disque
- `cylinder` — Aligne les partitions sur les cylindres
- `minimal` — Utilise le le plus petit alignement défini par la topologie du disque.
- `optimal` — Utilise l'alignement le plus optimisé donnée par la topologie du disque.

mkpart

Exemple de commande :

```
$ parted -a optimal /dev/usb mkpart primary 0% 4096MB
```

L'option `mkpart` prend en paramètre le type de partition (`primary`, `extended`, `logical`).

Puis en option le type de file système :

```
mkpart part-type [fs-type] start end
```

Et enfin l'adresse de départ et celle d'arrivé.

Par défaut les valeurs sont en megabit, sinon on peut préciser en pourcent, en secteur `2024s` ou `-1s` pour aller jusqu'au dernier secteur

label-type

```
mklabel label-type
```

L'option prend en paramètre une des valeurs :

aix, amiga, bsd, dvh, gpt, loop, mac, msdos, pc98, ou sun.

L'option print nous renseigne sur des informations intéressante sur le type de mémoire.

```
$ parted /dev/mmcblk0
GNU Parted 3.2
Using /dev/mmcblk0
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) print
Model: MMC W32508 (sd/mmc)
Disk /dev/mmcblk0: 7650MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type    File system  Flags
  1      1049kB  17.8MB  16.8MB  primary fat32        lba
  2      17.8MB  34.6MB  16.8MB  primary fat32        lba
  3      34.6MB  7650MB  7616MB  extended
  5      35.7MB  560MB   524MB   logical ext4
  6      561MB   1085MB  524MB   logical ext4
  7      1086MB  3183MB  2097MB  logical ext4
  8      3185MB  3709MB  524MB   logical ext4
  9      3710MB  4234MB  524MB   logical ext4
 10     4235MB  7650MB  3415MB  logical ext4
```

Alignement

La difficulté est de bien aligner les secteurs si l'on utilise l'option optimal.

On risque de tomber sur ce genre d'avertissement :

```
(parted) mkpart primary 0 100%  
Warning: The resulting partition is not properly  
        aligned for best performance.  
Ignore/Cancel?
```

Si on ignore il va remplacer le premier secteur (à 512B).

Pour anticiper ce genre de problème et pour être sûr d'être sur des secteurs optimiser on peut trouver de l'information dans `/sys/`.

```
$ cat /sys/block/sdb/queue/optimal_io_size
1048576
$ cat /sys/block/sdb/queue/minimum_io_size
262144
$ cat /sys/block/sdb/alignment_offset
0
$ cat /sys/block/sdb/queue/physical_block_size
512
```


Pour connaître le premier block, il faut prendre :
 $(optimal_io_size + alignment_offset) / physical_block_size$

Sauf qu'il nous manque des infos.

Dans l'exemple on avait :

$(1048576 + 0) / 512 = 2048$.

```
(parted) mkpart primary 2048s 100%
```

C'est possible de tester l'alignement de la partition avec

```
(parted) align-check optimal 1 "
```

Attention

Avec `parted`, les effets sont immédiat contrairement à `fdisk` où il faut sauvegarder les changement avant application

sfdisk

sfdisk vs fdisk [man]

- fdisk - Manipuler la table de partitions d'un disque
- sfdisk - Afficher ou manipuler une table de partitions de disque

On peut lire : "sfdisk est un outil orienté script pour le partitionnement de n'importe quel périphérique bloc."

sfdisk prend en entrée de ligne au format :

```
<start> <size> <id> <bootable> <c,h,s> <c,h,s>
```

Exemple :

```
{
echo ,9,0x0C,*
echo ,,-
} | sfdisk -D -H 255 -S 63 -C $CYLINDERS $DRIVE
```

Ou

```
sfdisk /dev/hdc << EOF
    0,407
    ,407
    ;
    ;
    EOF
```

fdisk

fdisk est normalement utilisable sous forme de menu.
On peut scripter les commandes envoyés au menu :

```
(  
  echo "n"      # creer une nouvelle partition  
  echo "p"      # de type primary  
  echo "1"      # de numero 1  
  echo ""       # a partir du premier secteur de libre  
  echo "+16M"   # de taille 16Mo  
  echo "w"      # sauvegarder et quitter  
) | fdisk ${DEVICE}
```

On peut formater la partition (fat/ext) :

```
(  
    echo "t" # change le format  
    echo "1" # partition number  
    echo "c" # id  
    echo "w" #  
) | fdisk ${DEVICE}
```

c = fat (LBA)

83 = linux

```
$ mkfs.vfat ${DEVICE}p1
```

Var d'environnement d'U-Boot

Erreur de partition d'environnement

```
Warning - bad CRC, using default environment
```

Answer : Most probably everything is OK. The message is printed because the flash sector or ERPROM containing the environment variables has never been initialized yet. The message will go away as soon as you save the environment variables using the saveenv command.

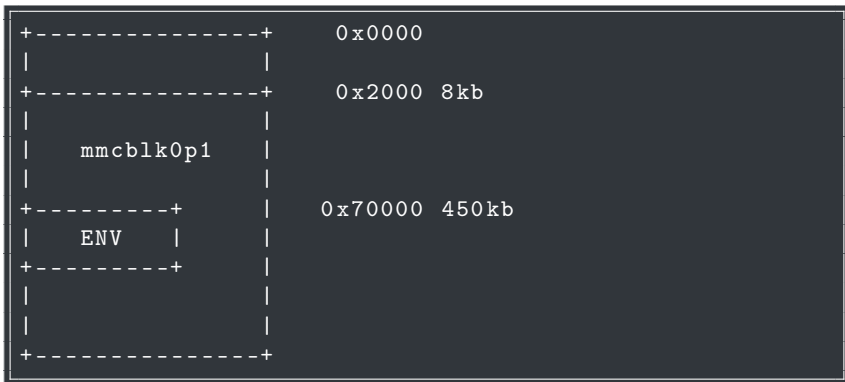
<https://www.denx.de/wiki/DULG/>

WarningBadCRCUsingDefaultEnvironment

Une des possibilités est que côté espace utilisateur vos paramètres ne soient pas correcte pour les outils u-boot.
Exemple d'adressage dans le fichier `/etc/fw_env.config` :

```
# Up to two entries are valid, in this case the redundant
# environment sector is assumed present.
# MTD device name      Device offset  Env. size      Flash sector size
/dev/mmcblk0           0x70000       0x2000         0x2000
```

On obtiendrait un partitionnement comme ceci :



Avec un correctif adapté on pourrait comme par exemple :
 /etc/fw_env.config

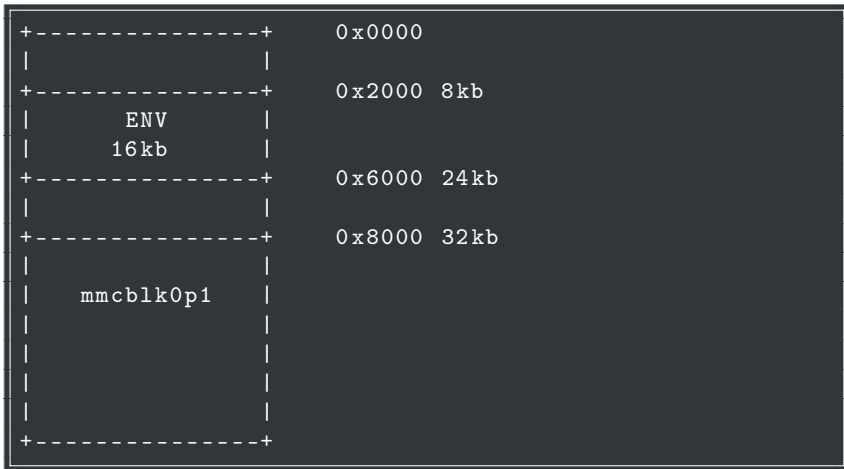
```
----- /etc/fw_env.config -----
index 650a4c3b7..f833324b4 100644
00 -2,4 +2,4 00
# Up to two entries are valid, in this case the redundant
# environment sector is assumed present.
# MTD device name      Device offset  Env. size      Flash sector size
-/dev/mmcblk0          0x70000        0x2000         0x2000
+/dev/mmcblk0          0x2000         0x4000         0x200
```

$0x2000 = 8kb$

$0x4000 = 16kb$

$0x200 = 512bytes$

On obtiendrait donc dans `/dev/mmcblk0` :



Question ?

Enfin je vais essayer de répondre...