

# Systèmes d'exploitation pour l'embarqué

## UV 5.2 - Exécution et Concurrency

Paul Blottière

ENSTA Bretagne

2015 / 2016

<https://github.com/pblottiere>

# Amélioration continue

## Contributions



- ▶ Dépôt du cours : <https://github.com/pblottiere/embsys>
- ▶ Souhaits d'amélioration, erreurs, idées de TP, ... :  
ouverture d'Issues (avec le bon label!)
- ▶ Apports de corrections : Pull Request

# Mémoire et système de fichiers

# Plan

1. Généralités
2. RAM
3. ROM
4. Flash
5. Systèmes de fichiers
6. Les outils sous Linux

# Généralités (1)

## Caractéristiques

Il existe deux familles de mémoire :

- ▶ mémoire vive ou RAM (Random Access Memory)
- ▶ mémoire morte ou ROM (Read Only Memory)

# Généralités (1)

## Caractéristiques

Il existe deux familles de mémoire :

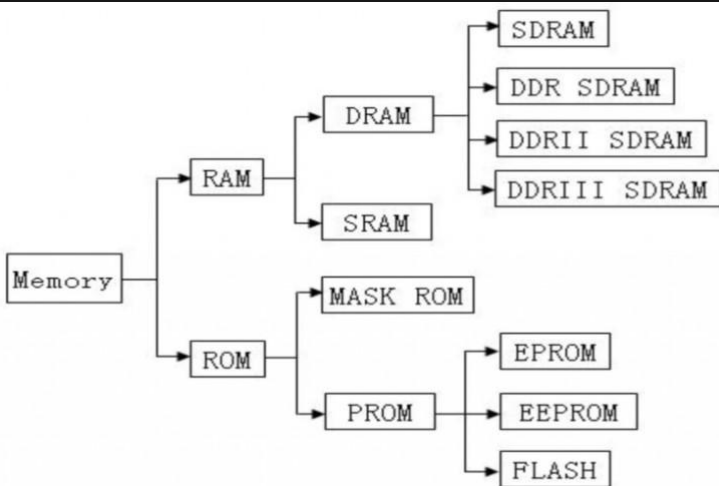
- ▶ mémoire vive ou RAM (Random Access Memory)
- ▶ mémoire morte ou ROM (Read Only Memory)

Chacune de ces familles peut être caractérisée par :

- ▶ la capacité
- ▶ la durée de vie
- ▶ le temps d'accès
- ▶ le coût

# Généralités (2)

## Arbre généalogique



# RAM (2)

## Définition

Mémoire volatile : perte des données en cas de coupure de l'alimentation du périphérique.



# RAM (2)

## Définition

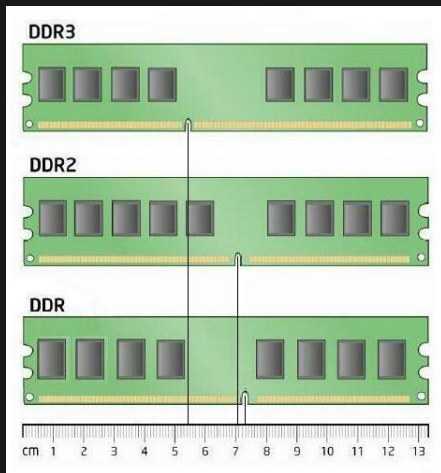
Mémoire volatile : perte des données en cas de coupure de l'alimentation du périphérique.

Plusieurs types :

- ▶ **DRAM** (Dynamic RAM) : les transistors (déchargés ou chargés => bit 0 ou 1) doivent être périodiquement rafraîchit (charge toutes les N ms) pour compenser les fuites.
- ▶ **SRAM** (Static RAM) : comme la DRAM mais pas besoin de rafraîchissement
- ▶ **SDRAM** (Synchronous DRAM) : synchronisation avec l'horloge du CPU
- ▶ **DDR SDRAM** (Double Data Rate SDRAM) : sur tous les PC standards. Grande bande passante.

# RAM (2)

## DDR en images



# ROM (1)

## Caractéristiques

Mémoire véritable : données persistantes même après coupure de l'alimentation.

# ROM (1)

## Caractéristiques

Mémoire véritable : données persistantes même après coupure de l'alimentation.

Plusieurs types :

- ▶ **ROM** (Read Only Memory) : programmée une seule fois lors de sa fabrication.
- ▶ **PROM** (Programmable ROM) : peut être programmée une fois par un utilisateur par un *programmeur*. Obsolète.
- ▶ **EPROM** (Erasable PROM) : effaçable par UV et reprogrammable.
- ▶ **EEPROM** (Electrically EPROM) : effaçable par faible tension et à la volée.

# ROM (2)

En images



# ROM (3)

## CHS

**Cylinder/Head/Sector** : géométrie d'un disque.

**Disque dur** : plusieurs plateaux.

**Plateau** : plusieurs pistes circulaires.

**Piste** : plusieurs secteurs (1 -> S).

**Cylindre** : union des mêmes pistes de tous les plateaux (0 -> N-1).

**Tête d'écriture/lecture** : 2 par plateau (0 -> N-1).

# ROM (3)

## CHS

**Cylinder/Head/Sector** : géométrie d'un disque.

**Disque dur** : plusieurs plateaux.

**Plateau** : plusieurs pistes circulaires.

**Piste** : plusieurs secteurs (1  $\rightarrow$  S).

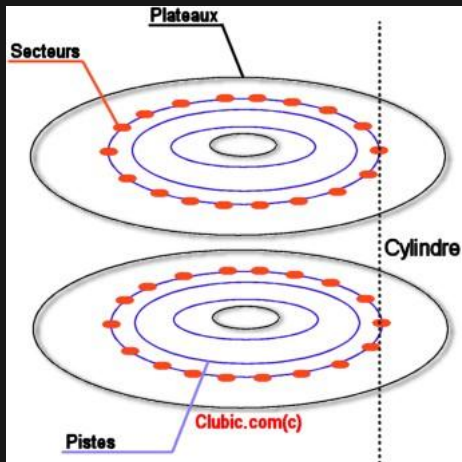
**Cylindre** : union des mêmes pistes de tous les plateaux (0  $\rightarrow$  N-1).

**Tête d'écriture/lecture** : 2 par plateau (0  $\rightarrow$  N-1).



# ROM (4)

Autopsie





# ROM (5)

## MBR vs GPT

**Master Boot Record** : table de partitions + bootloader sur le premier secteur 0/0/1 d'un disque dur. Utilisé par le BIOS.

Le MBR limite le nombre de partitions (4) ainsi que leur taille (2.2 To).

# ROM (5)

## MBR vs GPT

**Master Boot Record** : table de partitions + bootloader sur le premier secteur 0/0/1 d'un disque dur. Utilisé par le BIOS.

Le MBR limite le nombre de partitions (4) ainsi que leur taille (2.2 To).

Une carte mère avec UEFI (remplaçant du BIOS) ne peut démarrer que sur un disque utilisant le système de partitionnement **GPT**. Ce dernier remplace le MBR : 128 partitions, 256 To par partition.

# ROM (6)

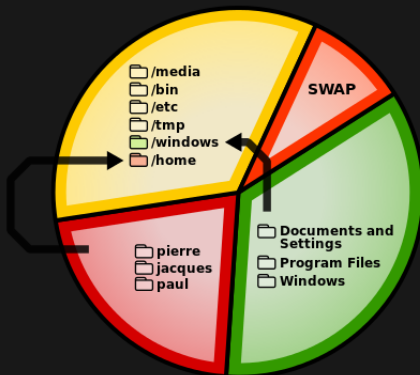
## Partition

Fraction d'un disque dur possédant son propre système de fichiers.

# ROM (6)

## Partition

Fraction d'un disque dur possédant son propre système de fichiers.



# Flash (1)

## Caractéristiques

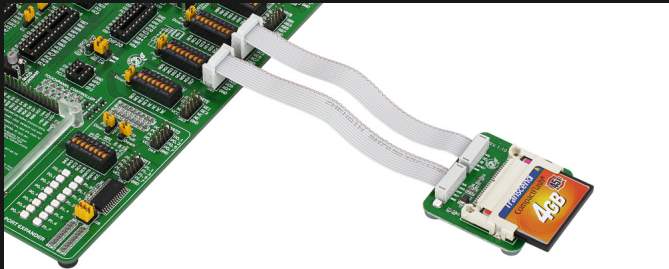
EEPROM améliorée : temps de reprogrammation éclair,  
grosse capacité, pas cher!

# Flash (1)

## Caractéristiques

EEPROM améliorée : temps de reprogrammation éclair, grosse capacité, pas cher!

Lorsque ces mémoires sont intégrées dans des périphériques (comme CompactFlash ou SSD), Linux les gère comme des périphériques blocs classiques (SATA, USB, ...).



# Flash (2)

## Partition virtuelle

Dans les systèmes embarqués, la mémoire flash est souvent intégrée sur les cartes mères.

=> le pilote MTD (Memory Technology Device) est utilisé

# Flash (2)

## Partition virtuelle

Dans les systèmes embarqués, la mémoire flash est souvent intégrée sur les cartes mères.

=> le pilote MTD (Memory Technology Device) est utilisé

Contrairement aux mémoires de masse *classiques*, on travaille directement avec les adresses mémoires.



# Flash (2)

## Partition virtuelle

Dans les systèmes embarqués, la mémoire flash est souvent intégrée sur les cartes mères.

=> le pilote MTD (Memory Technology Device) est utilisé

Contrairement aux mémoires de masse *classiques*, on travaille directement avec les adresses mémoires.

|   |
|---|
| Bootloader (raw, read-only)                             |
| Kernel (raw, read-only)                                 |
| Root filesystem (read-only)                             |
| Data filesystem (read-write)                            |
| Alternate root filesystem<br>(read-write, for upgrades) |

# Flash (3)

/proc/mtd

Pour une carte armadeus APF28 :

```
> cat /proc/mtd
dev:      size      erasesize  name
mtd0: 00300000 00020000 "u-boot"
mtd1: 00080000 00020000 "env"
mtd2: 00080000 00020000 "env2"
mtd3: 00080000 00020000 "dtb"
mtd4: 00080000 00020000 "splash"
mtd5: 00800000 00020000 "kernel"
mtd6: 1c200000 00020000 "rootfs"
mtd7: 03200000 00020000 "user"
```

# Flash (4)

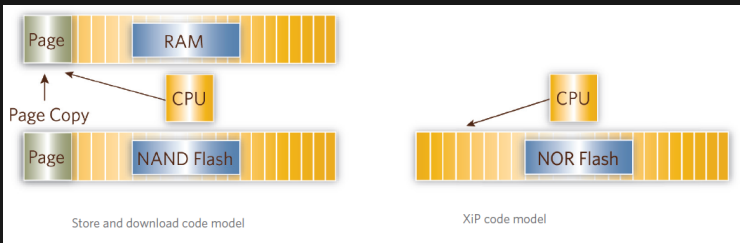
## XIP

**eXecute In Place** : la mémoire flash de type NOR permet d'exécuter du code directement en mémoire sans le charger en RAM.

# Flash (4)

## XIP

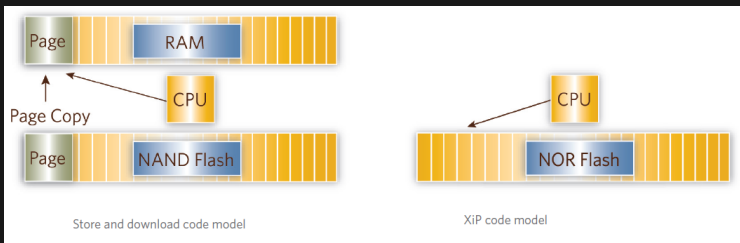
**eXecute In Place** : la mémoire flash de type NOR permet d'exécuter du code directement en mémoire sans le charger en RAM.



# Flash (4)

## XIP

**eXecute In Place** : la mémoire flash de type NOR permet d'exécuter du code directement en mémoire sans le charger en RAM.



De nos jours, la mémoire NAND est la plus utilisée (moins chère)

# Systèmes de fichiers

Une grande famille...

Les différents systèmes de fichiers :

- ▶ **VFAT** : supporté par tous les OS. 4 GB max pour un fichier.
- ▶ **NTFS** : issu du monde Windows mais supporté sous Linux grâce au paquet ntfs-3g.
- ▶ **XFS** : très efficace pour les I/O parallèles.
- ▶ **EXT2/3/4** : issu du monde Linux. Vise à remplacer MINIX fs.
- ▶ **JFFS1/2** : pour les mémoires flash.
- ▶ **UBIFS** : remplaçant du JFFS2 . Utilise la couche UBI du kernel Linux.

# Outils sous Linux (1)

Liste non exhaustive

- ▶ **df** : report file system disk space usage
- ▶ **blkid** : locate/print block device attributes
- ▶ **mount/umount** : mount/unmout a filesystem
- ▶ **fdisk** : manipulate disk partition table
- ▶ **parted/gparted** : a partition manipulation program/GUI
- ▶ **mkfs.fat/mkfs.ext2/...** : build a Linux filesystem
- ▶ **lsblk** : list block devices

# fstab

## Utilité

**fstab** : liste des disques utilisés ainsi que leurs caractéristiques.



# fstab

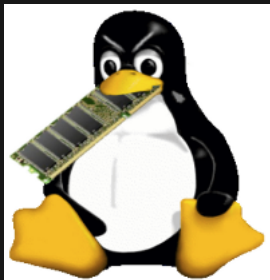
## Utilité

**fstab** : liste des disques utilisés ainsi que leurs caractéristiques.

```
> cat /etc/fstab
# /etc/fstab: static file system information.
#
# <fs> <mount point> <type> <opt> <dump> <pass>
# / was on /dev/sda7 during installation
UUID=XXX-XXX / ext4 errors=remount-ro 0 1
# swap was on /dev/sda5 during installation
UUID=XXX-XXX none swap sw 0 0
/dev/sda2 /media/windows ntfs
/dev/sda6 /media/data ntfs
```

## Conclusion

Dans les systèmes embarqués, la gestion de la mémoire est un point critique!



# Références

- ▶ Linux Embarqué - Pierre Fichoux