

Informatique Embarquée M2 / 2017

Systèmes Fichiers Embarqués

Références

- http://elinux.org/File_Systems
- <http://www.linux-mtd.infradead.org/faq/general.html>
- <http://free-electrons.com/doc/flash-fileSystems.pdf>

Systèmes de Fichiers

- InitRAMFS
- CRAMFS
 - Compressed RAM File System, Read-Only
 - Compression par page (accès direct facilité)
 - Taille de fichier: < 16 Mb
 - Taille Totale < 256 Mb (272Mb dans certaines conditions)
- SquashFS (*similaire*) (*kernel* $\geq 2.6.29$)
 - Meilleure compression,
 - Moins de limitations (taille...)

Mémoire Flash

- Non Volatile, sans alimentation
- Deux types de mémoire
 - NOR, NAND
- Lecture « rapide » (pas autant que DRAM)
- Écriture
 - A partir d'un bloc effacé (tous les bits à 1)
 - Écriture : met les bits appropriés à 0
- Nombre d'effacements limités
 - (~100 000... 1 000 000)

Mémoire Flash

- NOR:
 - Accès en lecture direct (byte)
 - Permet « Exécution en Place » (XIP)
 - ▶ Code exécuté depuis la Flash
 - ▶ Pas de recopie en RAM
- NAND
 - Accès de type « bloc » comme un disque
 - Mais un disque peut-être lu/écrit
 - Une mémoire flash peut / doit être effacée (erased)

Wear Levelling

- Si on écrit / ré-écrit toujours le même bloc
 - Il s'use « plus rapidement » que le reste de la mémoire flash...
- Répartir l'usure de manière la plus uniforme possible entre les blocs de mémoire.
 - Couche basse du pilote de périphérique
- Mais importance des structures de données du système de fichiers.
 - Systèmes de fichiers spécifiques pour les

Block Devices / MTD

Périphérique de type bloc	Memory Technology Devices (MTD)
Secteurs (en général 512 / 1024 octets)	“Erase Blocks” souvent 128 Kb NOR 8 Kb NAND
Read / Write	Read / Write / Erase
Secteurs corrompus masqués par le matériel (le plus souvent)	Blocs usés doivent être masqués par logiciel
Pas d'usure des secteurs	Usure des blocs

Systèmes usuels

- On peut utiliser un système de fichiers « standard » sur une mémoire Flash
- Utilisation d'une couche FTL (Flash Translation Layer) « émulant » un périphérique de type bloc sur la Flash
- On peut ensuite
 - Partitionner la flash
 - Formater les partitions avec
 - ▶ Ext2 / Ext3
 - ▶ FAT (pratique pour échanges Windows / Linux)
 - ▶ Etc...

Flash File Systems (NAND)

- JFFS2 Journalling Flash File System
 - Basé sur la « journalisation »
 - Doit parcourir la mémoire pour reconstituer état du système de fichiers
 - => Temps de montage (mount) important
 - Temps de montage réduit avec « EBS »
 - ▶ Erase Block Summary
- YAFFS2: Yet Another Flash File System
 - Temps de montage réduit

UBIFS

- Unsorted Block Image
- Fonctionne sur les mémoires sans FTL
 - (cf une des pages précédentes)
- Deux couches
 - UBI
 - UBIFS
- Permet le « wear levelling » global indépendamment du partitionnement.