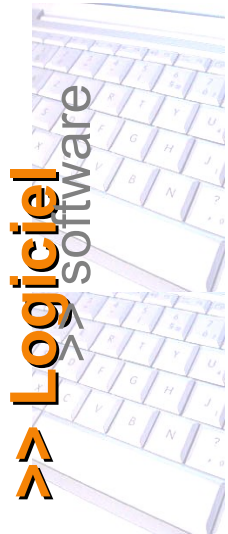
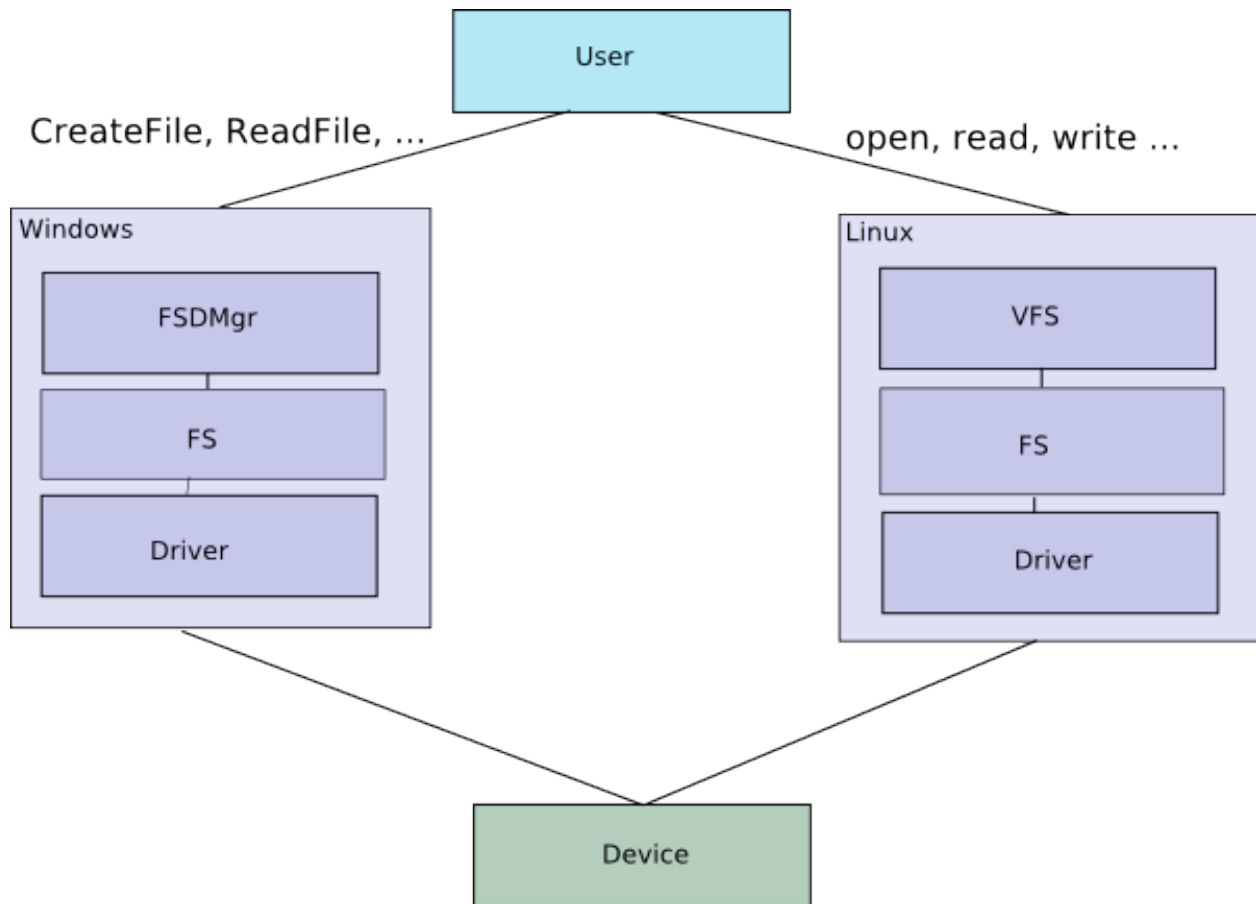


« Système de fichier pour NAND Flash »



- Système de fichier ?
- Mémoire Flash
  - NOR ou NAND ?
  - Besoins
- Solutions existantes
  - Fat + FTL
  - YAFFS2
  - JFFS2
  - LogFS
- NandFS
  - Fonctionnement
  - État du projet
  - Benchmarks

# Système de fichier



>> **Logiciel** >> software

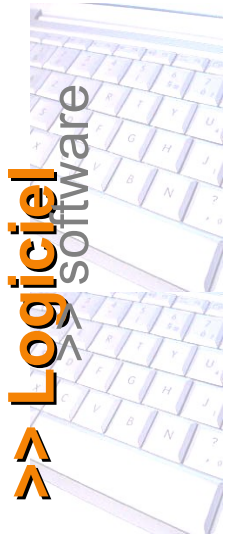
# NOR ou NAND ?



	<b>NOR</b>	<b>NAND</b>
Interface	BUS	I/O
Coût des cellules	Élevé	Faible
Lecture	Rapide	Lente
Écriture un Byte	Rapide	Lente
Écriture plusieurs Bytes	Lente	Rapide
Effacement	Lent	Rapide
Consommation	Élevé	Faible
Exécution de code	Oui	Non
Réécritures (sans effacement)	Quasi-illimité	1-3 fois
Stockage garanti	Oui	Non, besoin d'ECC



- Durée de vie d'un bloc 100 000 cycle  
écriture/effacement (avec ECC)



- Lire une page : 200 $\mu$ s
- Ecrire une page : 300 $\mu$ s
- Lire un spare : 30 $\mu$ s
- Ecrire un spare : 100 $\mu$ s
- Effacer un bloc : 2ms

Pour 256 MB

- Scan Page et Spare : 30s
- Scan Spare : 4s



- Mise a jour « déplacée »
- Bonne répartition
- Temps de montage rapide
- Résistant aux coupures brutales
- Correction des données (ECC)
- Portable (WinCE, Linux ..)
- Ralentir la détérioration de la NAND
- Faible empreinte mémoire





## Flash Translation Layer (FTL)

Permet d'utiliser une NAND comme un périphérique de stockage traditionnel. Utilisé avec un système de fichier pour disque dur, les performances ne sont pas optimales.

## Garbage Collector

A cause des limitations de la NAND, les données sont rendues obsolètes et effacées plus tard, c'est le rôle du garbage collector.





Systèmes de fichiers traditionnel + FTL

- FAT, TFAT
- Ext2

Systèmes de fichiers travaillant directement sur la NAND

- JFFS
- YAFFS
- LogFS





- « Power Safe » (en théorie uniquement)
- Problèmes dus au FTL
  - Performances
  - Durée de vie
- WinCE uniquement



Plus :

- Compression à la volée
- Support XATTR, ACL, ..

Moins :

- Temps de montage long
- Empreinte mémoire
- Performances
- Pas de portage vers WinCE

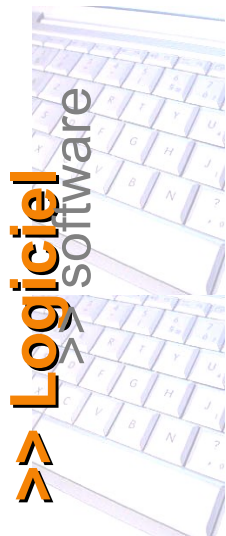


Plus :

- Performances
- Temps de montage

Moins :

- Portage vers WinCE payant (double licence)
- NAND Large Page uniquement (YAFFS1 pour les Small)
- GC non threadé



Plus :

- Empreinte mémoire
- Temps de montage
- Compression

Moins :

- Développement en cours
- Performances ?
- Linux uniquement



- Principe
- Représentation en RAM, sur la NAND
- Cohérence des données
- « Wear Leveling »
- Garbage Collector
- Benchmarks
  - Montage
  - Écriture
  - Manipulation de fichiers
- État du projet



## Logique (RAM)

Object :

Fichiers, Répertoires, Sockets, Fifo, Devices

Id unique, version, type

Header :

Métadonnées (nom, uid, gid, atime, mtime, etc ..)

Chunk :

Données des fichiers

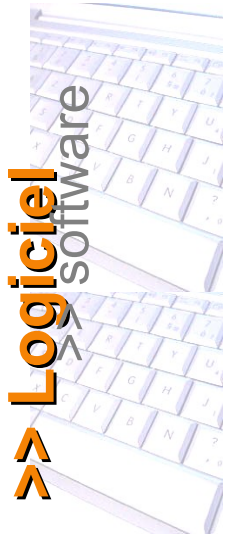
Id du fichier, version, offset/len

## Physique (NAND)

Page :

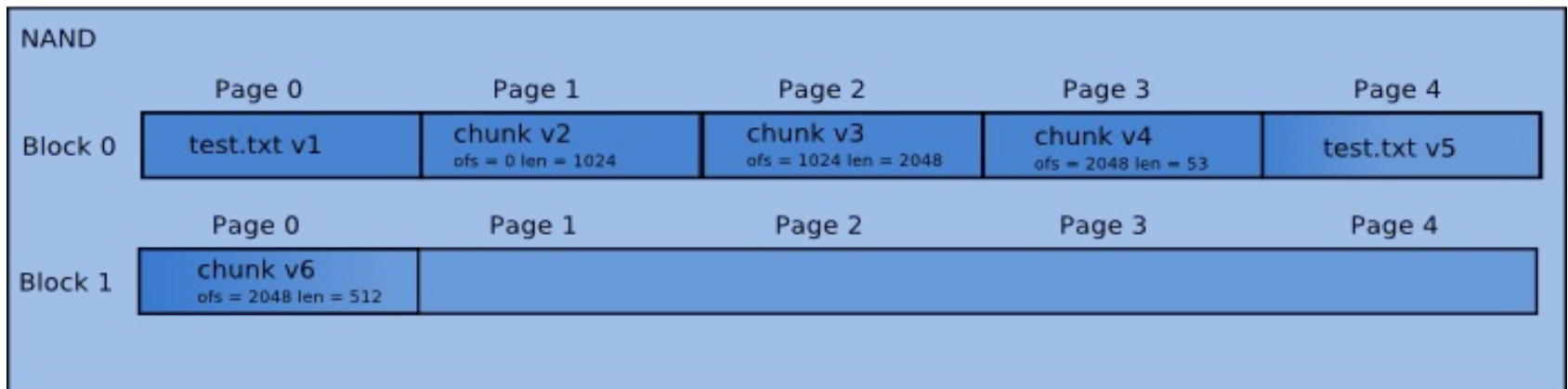
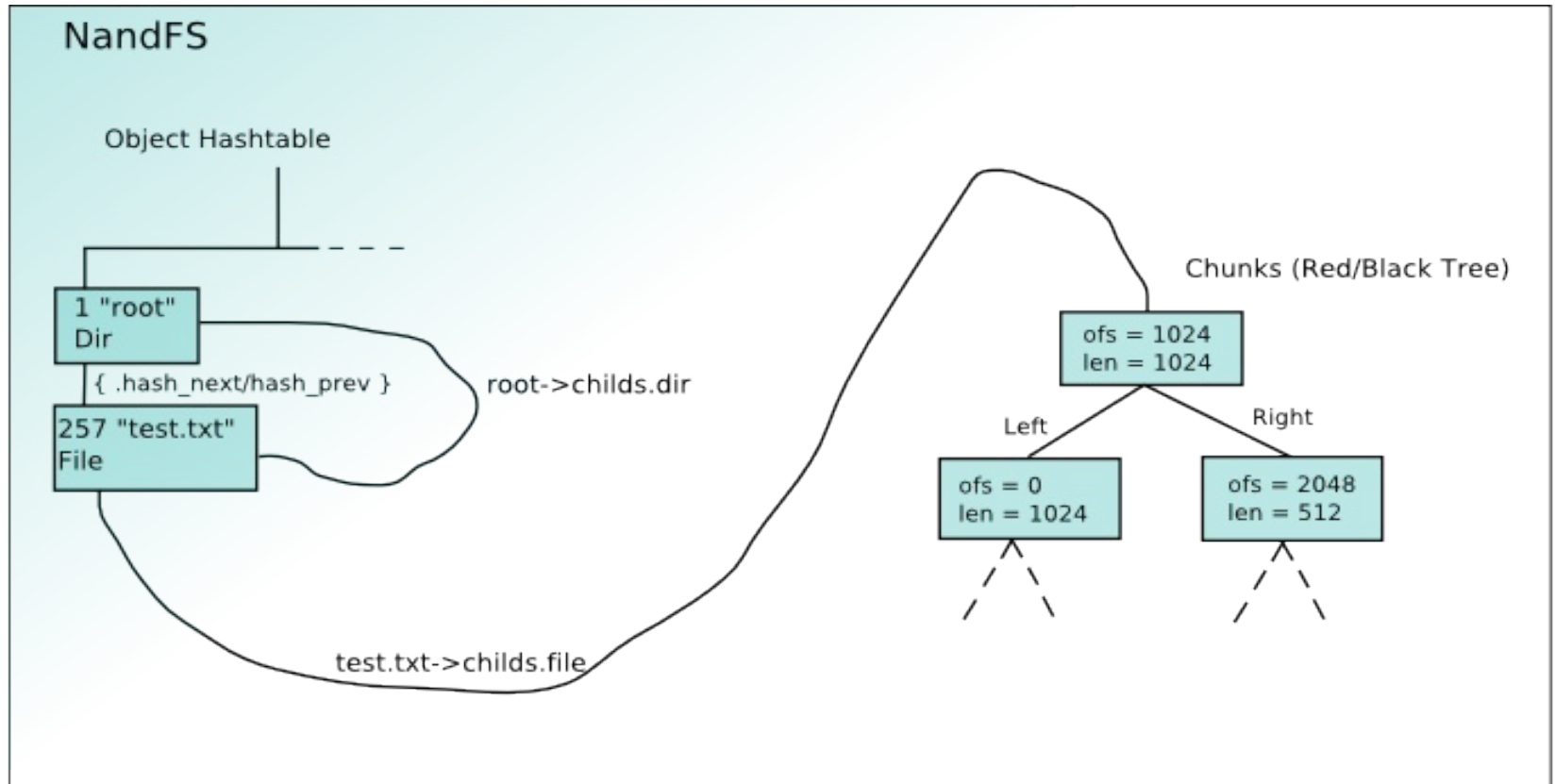
Id, Version, Type, Données spécifiques (objet ou chunk)

Données





# NandFS - Représentation





## Lecture

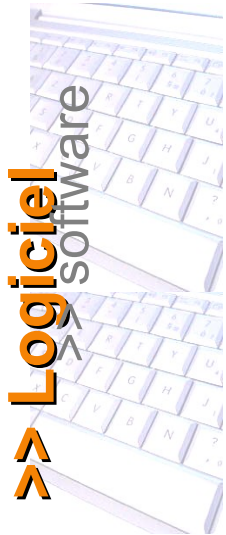
- Correction d'erreurs (ECC)

## Écriture

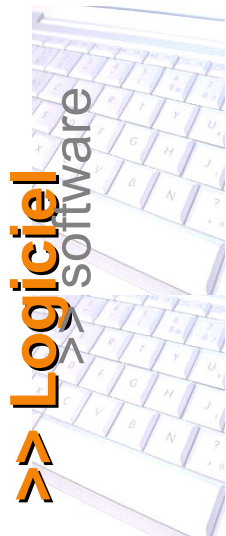
- Vérification des pages après écriture (optionnelle)
- Dernière version toujours conservée tant que la nouvelle n'est pas correctement écrite

## Montage

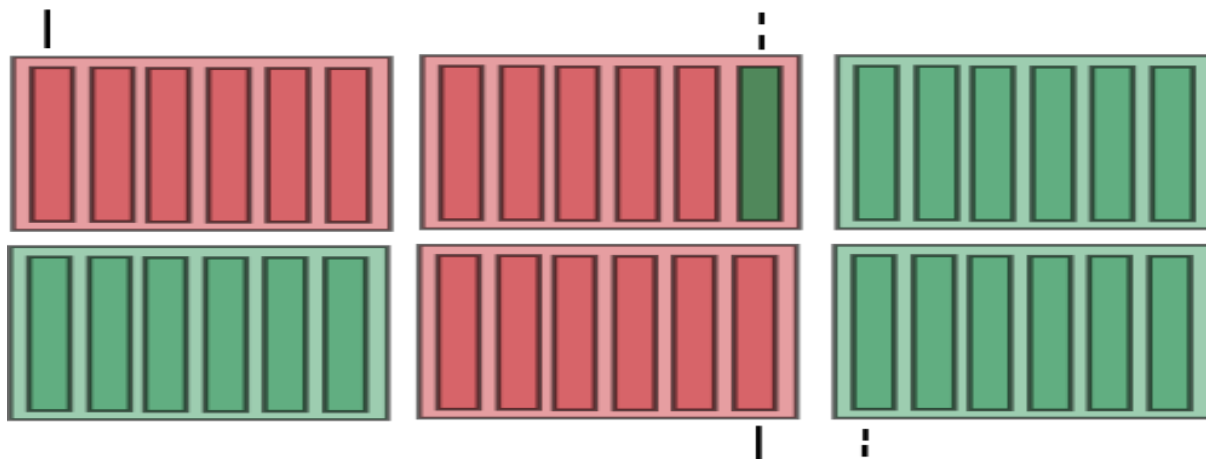
- Conserve uniquement la dernière version de l'object
- Déplacement des fichiers perdus dans le répertoire lostnfound



# NandFS – Wear Leveling

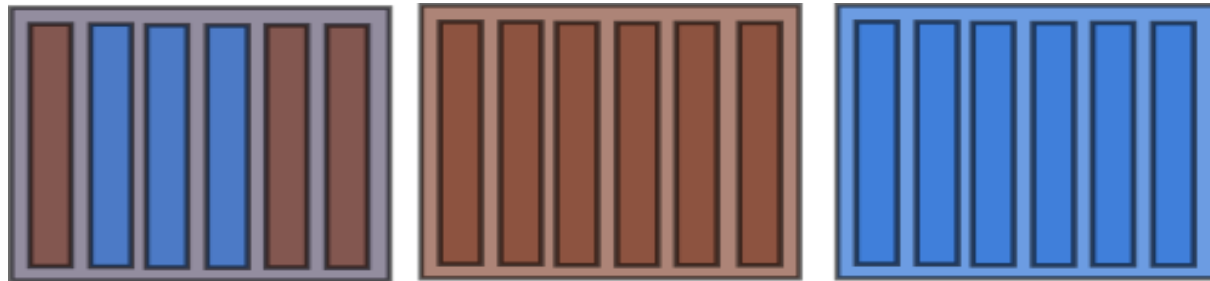


Chaque modification nécessite une nouvelle page.  
Lors de l'attribution de page pour l'écriture on cherche un bloc contenant des pages libres a partir de la dernière page libre attribuée.



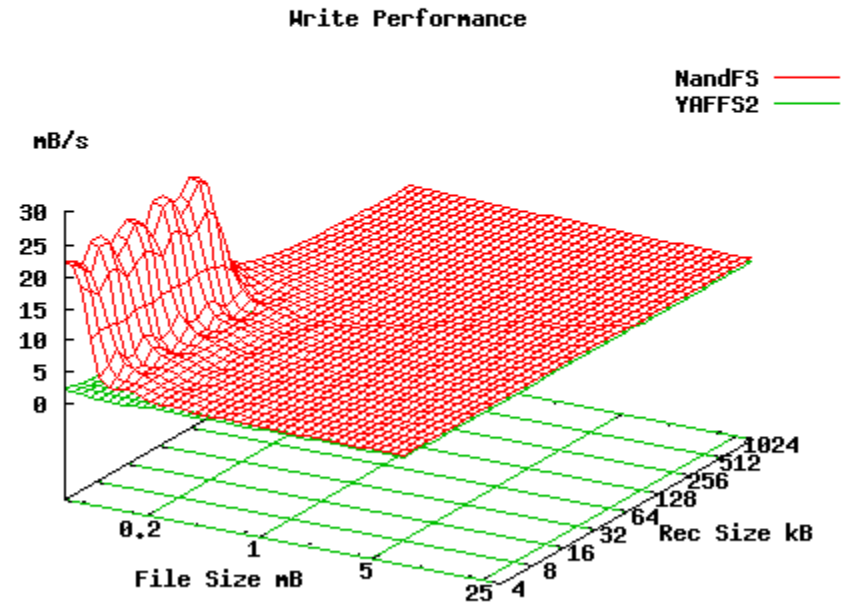
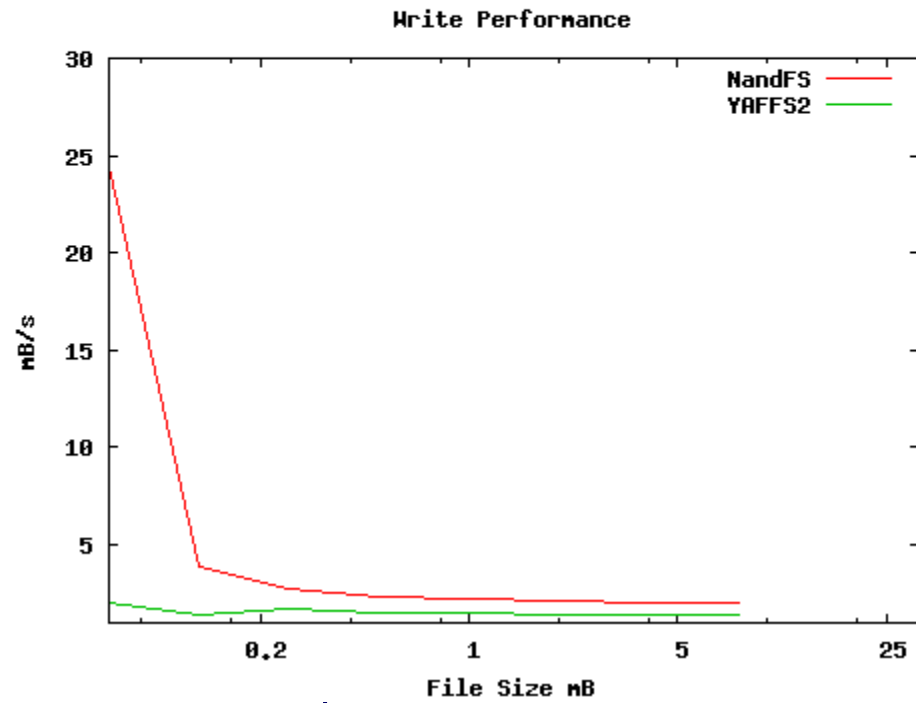


- Nettoyage des blocs « sales »

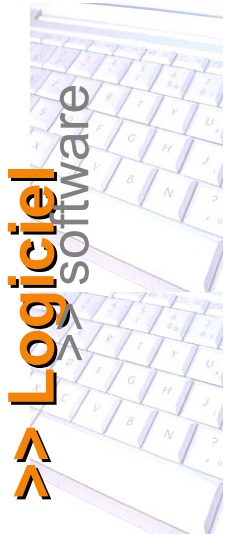


- Traitement des « bad blocks »

# NandFS – Benchmarks - iozone



# NandFS – Benchmarks



	Montage 1	Montage 2	Copie 1	Copie 2	Suppression
JFFS2	1,28 sec	9,95 sec	2 m 39 sec	18,14 sec	12,23 sec
YAFFS2	0,20 sec	0,13 sec	26,50 sec	6,62 sec	5,15 sec
NandFS	0,14 sec	2,66 sec	26,54 sec	6,84 sec	0,17 sec

Montage 1 : NAND vide

Montage 2 : NAND pleine à 30%

Copie 1 : 60MB (500 Fichiers, 40 dossiers)

Copie 2 : 16MB

Suppression : 500 Fichiers, 40 Dossiers

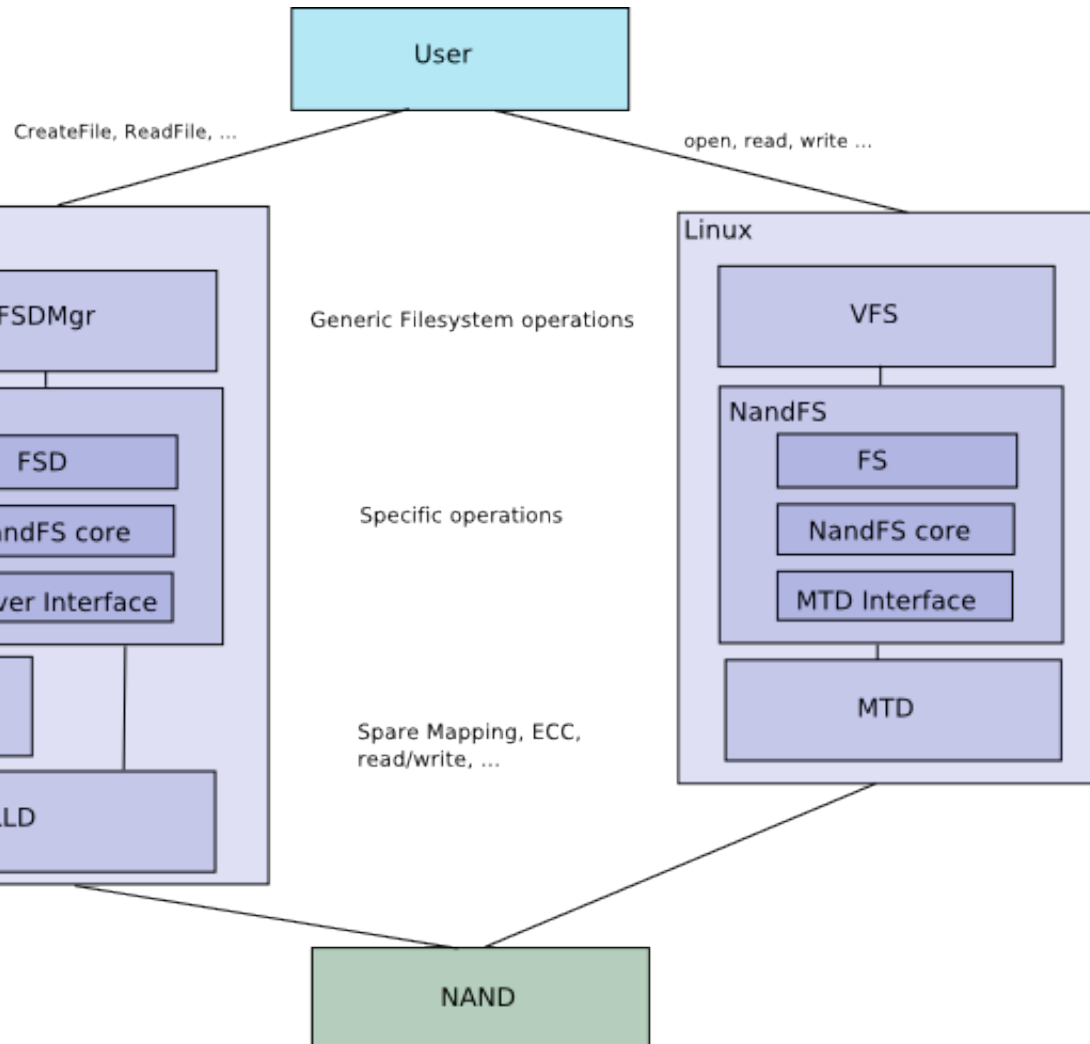
# NandFS – Benchmarks - bonnie++



	Sequential Output			Sequential Input		Random Seek
	Per Char	Block	Rewrite	Per Char	Block	
NandFS	1611 K/sec	2103 K/sec	1674 K/sec	2592 K/sec	4246 K/sec	696 seek/sec
YAFFS	1608 K/sec	1443 K/sec	1299 K/sec	2476 K/sec	3880 k/sec	590 seek/sec

	Sequential Create			Random Create		
	Create	Read	Delete	Create	Read	Delete
NandFS	274 /sec	16662 /sec	972 /sec	283 /sec	+++	440 /sec
YAFFS	83 /sec	5404 /sec	309 /sec	97 /sec	+++	186 /sec





>> Logiciel  
>> software



- Réduction empreinte mémoire
- Checkpoints
- Meilleure intégration Linux
- Portage WinCE