



Le Laboratoire Supinfo Des Technologies Linux

Les différents systèmes de fichiers

Auteur : Brice Dékany
Version 0.1 – 30 avril 2004
Nombre de pages : 8

Table des matières

1. INTRODUCTION	3
1.1. QU'EST-CE QU'UN SYSTEME DE FICHIERS.	3
1.2. SPECIALEMENT SOUS LINUX	3
2. LES SYSTEMES DE FICHIERS QUI NE STOCKENT RIEN.	4
2.1. DEVFS.....	4
2.2. PROCFS.....	4
3. SYSTEME DE FICHIERS SOUS LINUX.....	5
3.1. EXT3.....	5
3.1.1. <i>Caractéristiques</i>	5
3.1.2. <i>Formater en ext3</i>	5
3.1.3. <i>Réparer une partition ext3</i>	5
3.2. REISERFS	5
3.2.1. <i>Caractéristiques</i>	5
3.2.2. <i>Formater en ReiserFS</i>	6
3.2.3. <i>Réparer une partition ReiserFS</i>	6
3.3. XFS	6
3.3.1. <i>Caractéristique</i>	6
3.3.2. <i>Formater en XFS</i>	6
3.3.3. <i>Réparer une partition XFS</i>	6
3.4. JFS.....	6
3.4.1. <i>Caractéristique</i>	6
3.4.2. <i>Formater en JFS</i>	6
3.4.3. <i>Réparer une partition JFS</i>	7
4. BENCHMARKS	8

1. Introduction

1.1. Qu'est-ce qu'un système de fichiers.

Les disques durs, aussi petits soient-ils, contiennent des millions de bits, il faut donc organiser les données afin de pouvoir localiser les informations, c'est le but du système de fichiers. Le formatage logique d'un disque permet de créer un système de fichiers sur le disque, qui va permettre à un système d'exploitation (Windows, UNIX, Linux, ...) d'utiliser l'espace disque pour stocker et utiliser des fichiers. Le système de fichiers est basé sur la gestion des clusters (en français « unité d'allocation »), c'est-à-dire la plus petite unité de disque que le système d'exploitation est capable de gérer.

Un cluster est constitué d'un ou plusieurs secteurs, ainsi plus la taille d'un cluster est importante, moins le système d'exploitation aura d'entités à gérer...

En contrepartie, étant donné qu'un système d'exploitation ne sait gérer que des unités d'allocation entière, c'est-à-dire qu'un fichier occupe un nombre entier de cluster, le gaspillage est d'autant plus grand qu'il y a de secteurs par cluster. On comprend alors toute l'importance du choix du système de fichiers.

1.2. Spécialement sous Linux

Les systèmes de fichiers sous Linux sont peu sensibles à la fragmentation contrairement au système comme FAT ou NTFS. C'est-à-dire que les données ne sont pas éparpillées sur tout le disque dur, l'accès aux données est donc plus rapide puisqu'elles sont toutes regroupées dans une même zone. Il n'est donc pas nécessaire de défragmenter son disque dur tout les « x » temps.

Les fichiers et les répertoires sont identifiés par des inodes. Toutes les inodes sont uniques au sein d'une même partition. Une inode contient des informations comme les droits (Read Write eXecute) sur le fichier ou le dossier qu'elle identifie.

2. Les systèmes de fichiers qui ne stockent rien.

Sous linux tout est fichier. Clavier, souris ou disque dur, tout est représenté par un fichier. C'est pour ça qu'il existe des systèmes de fichiers spéciaux qui ne prennent pas de place sur votre disque dur, ils sont placés en mémoire et stockent des liens vers les périphériques ou autres options du noyau.

2.1.Devfs

Le pseudo système de fichiers « devfs » contient des liens vers les périphériques de la machine. Ce système permet d'accéder à un périphérique via des noms simples au lieu d'y accéder par leur adresse réelle du block.

Exemple :

Le disque maitre de la première nappe IDE est identifié par

```
/dev/ide/host0/target0/lun0/disk
```

Grâce à devfs, il est également disponible via :

```
/dev/hda
```

Ce qui est plus rapide et plus facile à retenir.

Ce système de fichiers est totalement optionnel. Il est même devenu obsolète avec l'arrivée du noyau 2.6. Le système devfs est géré par un « daemon » qui s'appelle « devfsd ». Ce programme va se charger de créer le répertoire virtuel contenant tous les liens à chaque démarrage. Par convention, devfs est monté dans « /dev »

2.2.Procfs

Le pseudo système de fichiers « procfs » contient une information directement fournie par le noyau. Par convention, on peut y accéder via le répertoire « /proc ». On y trouve des informations depuis le bus pci jusqu'au taux de débit de la carte réseaux en passant par la fréquence du CPU.

3. Système de fichiers sous Linux

3.1.EXT3

3.1.1. Caractéristiques

C'est le plus connu et le plus utilisé des systèmes de fichiers sous GNU/Linux. L'ext3 est le descendant de l'ext2, il est donc très simple de transformer ses partitions ext2 en ext3. Cette opération se fait très rapidement et sans perdre les données. L'ext3 est un système journalisé, c'est-à-dire que toutes les opérations sont recensées dans un journal. Si la machine s'arrête à cause d'une coupure de courant par exemple, il suffira de lire le journal pour savoir où en était la machine et quelles opérations elle devait faire. Cette opération se fait automatiquement avec l'utilitaire « fsck ».

3.1.2. Formater en ext3

Les trois commandes suivantes sont équivalentes.

```
# mkfs.ext3 /dev/hda1
# mkfs -t ext3 /dev/hda1
# mke2fs -j /dev/hda1
```

Transformer sa partition ext2 en ext3

```
# tune2fs -j /dev/hda1
```

3.1.3. Réparer une partition ext3

```
# fsck.ext3 /dev/hda1
```

3.2.ReiserFS

3.2.1. Caractéristiques

ReiserFS est basé sur un b-arbre. Ce qui augmente ses performances comparé à ext3, surtout pour les petits fichiers (moins de 4ko.) Comme ext3, reiserfs est un système journalisé. C'est un système de fichiers destiné à l'utilisation de tous les jours pour les clients. En effet, reiserfs ne gère pas les quotas et le système NFS n'est pas très bien supporté, il y a beaucoup de latence. ReiserFS est un choix très intéressant surtout qu'il est proposé par défaut par de plus en plus de distributions (Mandrake, Debian, ...) La version 4 de ReiserFS est en préparation pour le noyau 2.6, elle offre d'excellente performance, mais n'est pas encore stable.

Site Officiel : <http://www.namesys.com/>

3.2.2. Formater en ReiserFS

```
# mkreiserfs /dev/hda1
```

3.2.3. Réparer une partition ReiserFS

```
# fsck.reiserfs /dev/hda1
```

3.3.XFS

3.3.1. Caractéristique

XFS est un portage pour Linux du système de fichiers des stations SGI. Ce système est basé sur un arbre b+ qui alloue dynamiquement les inodes. Il est également journalisé. XFS est surtout conseillé pour les disques durs en SCSI et offre de très bonnes performances surtout sur les gros fichiers. Ce système est aussi très performant avec les machines multi processeurs. XFS est définitivement orienté serveur. Seul problème, l'arrêt à chaud entraîne souvent des complications.

Site Officiel : <http://oss.sgi.com/projects/xfs/>

3.3.2. Formater en XFS

```
# mkfs.xfs /dev/hda1
```

3.3.3. Réparer une partition XFS

```
# fsck.xfs /dev/hda1
```

3.4.JFS

3.4.1. Caractéristique

JFS est un système de fichier porté sous Linux par IBM. Il est très récent ce qui fait qu'il n'y a pas encore beaucoup de retour. Il est donc déconseillé d'utiliser ce système pour stocker des données sensibles. JFS est journalisé et il a été conçu pour se réparer très rapidement et en gardant un maximum de données. JFS n'est pas encore beaucoup utilisé sous Linux, seul Gentoo le propose à l'installation, mais on peut espérer qu'il se développera très rapidement. Inconvénient de ce système de fichier, la lenteur due au fait qu'il journalise énormément.

Site Officiel : <http://oss.software.ibm.com/developer/opensource/jfs/>

3.4.2. Formater en JFS

```
# mkfs.jfs /dev/hda1
```

3.4.3. Réparer une partition JFS

```
# fsck.jfs /dev/hda1
```

4. Benchmarks

Les benchmarks suivant ont été réalisé par Scott Kveton de l'université d'état de l'oregon :
<http://oregonstate.edu/~kveton/fs/>

Large File - Kernel 2.6

	Sequential Output						Sequential Input				Random Seeks		Sequential Create						Random Create					
	Per Char		Block		Rewrite		Per Char		Block				Create		Read		Delete		Create		Read		Delete	
	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	/sec	% CPU	/sec	% CPU	/sec	% CPU	/sec	% CPU	/sec	% CPU	/sec	% CPU	/sec	% CPU
ext3	28460	99	47787	27	21086	9	24380	70	42029	10	520.5	1	52	15	71	6	144	1	51	15	23	2	75	1
jfs	30850	98	49510	14	19299	6	23844	68	39115	8	420.9	1	74	14	72	6	2263	9	73	14	22	2	142	0
reiserfs	29123	99	46959	23	19878	8	23742	72	40615	9	525.1	1	67	20	13	1	464	14	67	20	25	2	66	4
xfs	32168	99	46607	13	22575	8	26313	74	45720	10	487.4	1	71	14	69	6	1104	9	72	14	18	1	286	3

Small File - Kernel 2.6

	Sequential Output						Sequential Input				Random Seeks		Sequential Create						Random Create					
	Per Char		Block		Rewrite		Per Char		Block				Create		Read		Delete		Create		Read		Delete	
	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	K/sec	% CPU	/sec	% CPU	/sec	% CPU	/sec	% CPU	/sec	% CPU	/sec	% CPU	/sec	% CPU	/sec	% CPU
ext3	28292	98	48541	28	21206	8	24507	71	42177	10	528.2	1	379	3	69162	99	656	3	384	3	61731	100	92	1
jfs	31114	97	49343	14	21549	7	25012	73	40382	10	481.5	1	1097	18	67222	100	6265	21	1032	16	59649	100	208	1
reiserfs	29723	98	47272	22	19941	8	24035	70	40081	10	537.0	1	467	14	67680	99	521	10	469	14	61866	99	154	5
xfs	32234	99	46727	13	22028	8	27514	76	48469	11	488.0	1	133	2	941	3	1486	10	153	2	681	2	254	3