Linux Partition Mini-HOWTO

Kristan Koehntopp, kris@koehntopp.de
Adaptation française : Raphaël Gurlie, raphael@ibpc.fr et Guillaume Bertucat, guillaume@ibpc.fr. Partition mini-HOWTO v 2.4, le 03 mars 1998

Ce Mini-HOWTO de Linux décrit comment prévoir et organiser l'espace disque de votre système Linux. Il traite des aspects matériels des disques, des partitions, de la taille et du positionnement des zones de swap, des systèmes de fichiers, des types de systèmes de fichiers ainsi que de thèmes apparentés. L'objectif est de donner quelques notions fondamentales, pas les modes opératoires.

Contents

1 Introduction

1.1 De quoi s'agit-il?

Ceci est un Mini-HOWTO de Linux. Un Mini HOWTO est un court texte qui fait le point sur des questions relatives à l'installation et à la maintenance de Linux. C'est Mini, parce que tant le texte que le thème traité sont trop "petits" pour justifier un vrai HOWTO ou un livre. Un HOWTO ne constitue pas une référence : les pages "man" sont là pour ça.

1.2 De quoi ne s'agit-il pas (et HOWTO apparentés) ?

Ce Mini-HOWTO de Linux explique comment prévoir et organiser l'espace disque de votre système Linux. Il traite des aspects matériels des disques, des partitions, de la taille et du positionnement des zones de swap, des systèmes de fichiers, des types de systèmes de fichiers ainsi que de thèmes apparentés. L'objectif est de donner quelques notions fondamentales, aussi nous parlerons essentiellement de principes et non pas d'outils dans ce texte.

Dans des circonstances idéales, ce document devrait être lu avant votre première installation, mais c'est sans doute peu réaliste dans la plupart des cas. Les débutants ont généralement d'autres problèmes que d'optimiser l'organisation de leur disque. Par conséquent, vous êtes probablement quelqu'un qui vient juste de finir l'installation de Linux, et qui maintenant se demande comment optimiser cette installation, ou comment éviter quelques déplaisantes erreurs de calculs pour la prochaine fois. Bien sûr, j'espère que lorsqu'ils en auront fini avec ce document, certains voudront laisser tomber leur ancienne configuration pour une nouvelle installation. :-)

Ce document se limite pour l'essentiel à la prévision et l'organisation de l'espace disque. Il ne décrit pas l'utilisation de `fdisk`, `LILO`, `mke2fs` ou des programmes de sauvegarde. Il y a d'autres HOWTO qui traitent de ces problèmes. Reportez-vous à la Liste-des-HOWTO de Linux pour obtenir les informations relatives aux différents HOWTOs de Linux. La liste contient également les informations nécessaires pour obtenir les documents eux-mêmes.

Pour apprendre à estimer les besoins en taille et en vitesse pour les différentes parties du système de fichiers, reportez-vous au "Linux Multiple Disks Layout mini-HOWTO", de Gjoen Stein <gjoen@nyx.net>.

Pour obtenir des informations et des instructions concernant les disques de plus de 1024 cylindres, reportez-vous au "Linux Large Disk Mini-HOWTO", de Andries Brouwer <aeb@cwi.nl>.

Pour obtenir des instructions sur la manière de limiter l'espace disque alloué à chaque utilisateur, reportez-vous au "Linux Quota Mini-HOWTO", de Albert M.C. Tam <bertie@scn.org>

Actuellement il n'y a pas de documentation générale sur la sauvegarde des disques, mais il existe un certain nombre de documents qui font le point sur des solutions spécifiques de sauvegarde. Reportez-vous au "Linux ADSM Backup Mini-HOWTO" de Thomas Koenig <Thomas.Koenig@ciw.uni-karlsruhe.de> pour obtenir des renseignements sur la manière d'intégrer Linux dans un environnement de sauvegarde IBM ADSM. Reportez-vous au "Linux Backup with MSDOS Mini-HOWTO" de Christopher Neufeld <neufeld@physics.utoronto.ca> pour obtenir des informations sur les sauvegardes Linux pilotés par MS-DOS.

Pour obtenir des instructions sur la manière d'écrire et de soumettre un HOWTO, reportez-vous à la "Liste-des-HOWTO" de Linux de Éric Dumas <dumas@Linux.EU.Org>.

Butiner dans /usr/src/linux/Documentation peut aussi se révéler très instructif. Les fichiers ide.txt et scsi.txt fournissent quelques informations fondamentales sur les propriétés de vos pilotes disque, et jeter un coup d'oeil à l'arborescence de votre système de fichiers ne peut pas faire de mal.

2 Qu'est-ce qu'une partition ?

Lorsque les disques durs pour PC ont été mis au point, on a rapidement cherché à avoir la possibilité d'installer plusieurs systèmes d'exploitation, même si on ne disposait que d'un seul disque. Par conséquent, il fallait un procédé permettant de diviser un seul disque physique en plusieurs disques logiques. Une partition, c'est justement cela : une section contiguë de blocs sur le disque dur, considérée comme un disque totalement indépendant par la plupart des systèmes d'exploitation.

Il est bien évident que les différentes partitions ne doivent pas se recouvrir : un système d'exploitation n'appréciera certainement pas qu'un autre OS installé sur la même machine écrase des données importantes à cause d'un tel recouvrement. D'autre part, il ne devrait pas non plus y avoir de "trou" entre deux partitions adjacentes. Bien que ce ne soit pas nuisible en soi, vous gâcheriez une place précieuse en laissant vides de tels espaces.

Il n'est pas indispensable que le disque soit entièrement partitionné. Vous pouvez décider de laisser de la place à la fin du disque qui ne soit attribuée à aucun de vos systèmes d'exploitation. Par la suite, lorsque vous saurez quel système vous utilisez le plus souvent, vous pourrez partitionner l'espace restant, et créer dessus un système de fichier approprié.

Les partitions ne peuvent être ni déplacées, ni redimensionnées sans détruire le système de fichiers qui s'y trouve. C'est pourquoi modifier la table de partition implique généralement de sauvegarder puis de restaurer tous les systèmes de fichiers touchés par cette opération. En fait il est assez facile de faire des dégâts irréparables en repartitionnant, et vous devriez faire une sauvegarde intégrale de tous les disques de la machine en question avant même de penser à utiliser un utilitaire comme `fdisk`.

Bon, à vrai dire, certaines partitions contenant certains types de système de fichiers *peuvent* être coupées en deux sans perte de données (si vous avez de la chance). Par exemple, il y a un utilitaire appelé `fips` pour couper en deux les partitions MS-DOS, ce qui permet de créer un espace pour installer Linux sans avoir à réinstaller MS-DOS. Mais vous n'avez pas vraiment l'intention de jouer avec ça sans sauvegarder soigneusement tout ce qui ce trouve sur votre machine ?

2.1 Les sauvegardes sont importantes

Pour les sauvegardes, les lecteurs de bandes sont vos amis. Ils sont rapides fiables et faciles à utiliser, ce qui permet de faire de fréquentes sauvegardes, de préférence automatiquement, et sans s'embêter.

Je tiens particulièrement à insister sur les points suivants : je parle de vrais lecteurs de bandes, pas de cette daube de ftape pilotée par le contrôleur du disque. Envisagez d'investir dans le SCSI : Linux supporte le SCSI de façon native. Vous n'aurez pas besoin de télécharger des pilotes ASPI. Vous ne perdrez pas non plus de précieuses HMA sous Linux dès que vous aurez installé votre contrôleur SCSI, vous n'aurez plus qu'à y ajouter vos disques durs, lecteur de bandes et lecteurs CDROM. Pas d'autres adresses I/O, plus besoin de jongler avec les IRQ, ni de s'inquiéter des compatibilités maître/esclave ou des niveaux PIO. En outre, un contrôleur SCSI approprié vous donne de hautes performances I/O sans augmenter notablement la charge du CPU. Même en cas de grande activité du disque, vous pourrez constater de bons temps de réponse. Si vous envisagez d'utiliser un système Linux comme un centre de distribution de news, ou si vous vous apprêtez à vous lancer dans le domaine des services d'accès à Internet, ne pensez même pas à un système sans SCSI.

2.2 Noms et numéros des périphériques

Le nombre de partitions sur un système à base d'Intel à été limité depuis le commencement : la table de partitions originale faisait partie intégrante du secteur d'amorçage, et la place prévue nous limitait à quatre partitions. Ces partitions sont maintenant appelées partitions primaires. Lorsqu'il est devenu évident que beaucoup avaient besoin de plus de quatre partitions sur leurs systèmes, les partitions logiques ont été créées. Le nombre de partitions logiques n'est pas limité : chaque partition logique contient un pointeur sur la suivante, et par conséquent, vous disposez potentiellement d'une liste non limitée de partitions.

Pour des raisons de compatibilité, l'espace occupé par les partitions logiques doit être comptabilisé. Si vous utilisez les partitions logiques, une des partitions primaires est donc notée "partition étendue" ; son bloc initial et son bloc final délimitent l'espace occupé par les partitions logiques. Ceci signifie que l'espace attribué pour toutes les partitions logiques doit être contigu. Il ne peut y avoir qu'une seule partition étendue : aucun `fdisk` n'acceptera de créer plus d'une partition étendue.

Linux ne peut prendre en charge qu'un nombre limité de partitions par disque. Ainsi avec Linux, vous disposez de 4 partitions primaires (dont 3 utilisables si vous utilisez les partitions logiques) et au mieux 15 partitions en tout sur un disque SCSI (63 en tout sur un disque IDE).

Sous Linux, les partitions sont identifiées par des fichiers périphériques. Un fichier périphérique est un fichier de type c (pour périphérique "caractère", les périphériques qui ne font pas usage de la cache tampon) ou b (pour périphérique "bloc", qui font usage de la cache tampon). Sous Linux, tous les disques sont représentés sous la forme de périphériques blocs uniquement. Contrairement à d'autres Unix, Linux ne propose pas de version strictement caractère des disques et de leurs partitions.

Les seules choses importantes à retenir d'un fichier périphérique sont ses numéros de périphérique, majeur et mineur, affichés à la place de la taille du fichier :

```
$ ls -l /dev/hda
brw-rw----   1 root     disk       3,   0 Jul 18  1994 /dev/hda
                                   ^    ^
                                   |    numéro périphérique mineur
                                   numéro périphérique majeur
```

Lorsqu'on accède au fichier périphérique, le numéro majeur détermine quel pilote périphérique va être appelé pour réaliser l'opération d'entrée/sortie. Cet appel est fait en prenant comme paramètre le numéro mineur, et c'est l'affaire du pilote d'interpréter correctement ce numéro mineur. La documentation du pilote décrit généralement la manière dont il interprète ces numéros mineurs. Pour les disques IDE, cette documentation se trouve dans `/usr/src/linux/Documentation/ide.txt`. Pour les disques SCSI, on s'attendrait à trouver la documentation dans `/usr/src/linux/Documentation/scsi.txt`, mais elle ne s'y trouve pas. Il peut être nécessaire de consulter la source du pilote pour être sûr (`/usr/src/linux/driver/scsi/sd.c:184-196`). Heureusement, il y a la liste des noms et numéros de périphériques de Peter Anvin dans `/usr/src/linux/Documentation/devices.txt`; reportez vous dans cette liste à `block devices`, `major 3`, 22, 33, 34 pour les disques IDE, et `major 8` pour les disques SCSI. Les numéros majeurs et mineurs sont codés chacuns sur un bit, ce qui explique pourquoi le nombre de partition par disque est limité.

Par convention, les fichiers périphériques ont un nom défini, et la plupart des utilitaires système sont compilés en ayant connaissance de ces noms. Ils s'attendent à ce que vos disques IDE s'appellent `/dev/hd*` et vos disques SCSI `/dev/sd*`. Les disques sont numérotés a, b, c et ainsi de suite, donc `/dev/hda` est votre premier disque IDE, et `/dev/sda` votre premier disque SCSI. Chaque périphérique représente un disque à part entière démarrant au bloc un. Écrire sur un de ces périphériques avec les mauvais utilitaires détruira l'enregistrement principal d'initialisation (MBR) et la table de partition, ce qui rendra toutes les données de ce disque inutilisables, et le système ne pourra plus démarrer sur ce disque. Donc soyez sûrs de ce que vous faites, et encore une fois, sauvegardez avant de faire quoi que ce soit.

Les partitions primaires sur le disques sont numérotées 1, 2, 3 et 4. Par conséquent, `/dev/hda1` est la première partition primaire du premier disque IDE, et ainsi de suite. Les partitions logiques se voient attribuer les numéros 5 et suivants; `/dev/sdb5` est donc la première partition logique du second disque SCSI.

Chaque partition se voit attribuer deux adresses pour les blocs initial et final, ainsi qu'un type. Le type est un code numérique (un bit) qui définit une partition pour un système d'exploitation donné. Pour la plus grande joie des experts, il n'existe pas vraiment de code unique définissant les différents types de partition, aussi il y a toujours une possibilité que deux systèmes d'exploitation utilisent le même code pour des partitions de type différent.

Linux réserve les codes 0x82 pour les partitions swap, et 0x83 pour les systèmes de fichier "natif" (c'est à dire ext2 pour la plupart d'entre vous). Autrefois populaire et maintenant périmé, le système de fichiers Linux/Minix utilisait le code 0x81 pour ses partitions. OS/2 marque ses partitions du type 0x07, tout comme les NTFS de Windows NT. MS-DOS attribue plusieurs codes pour les différentes FAT de ses systèmes de fichier : on connaît 0x01, 0x04 et 0x06. DR-DOS utilisait 0x81 pour indiquer une partition FAT protégée, ce qui générait un conflit avec les partitions Linux/Minix, mais ni l'une ni l'autre ne sont très utilisées maintenant. La partition étendue qui sert de container pour les partitions logiques à le code 0x05.

Les partitions sont créées et supprimées avec l'utilitaire `fdisk`. Tout système d'exploitation qui se respecte possède un `fdisk`, qui d'ailleurs est traditionnellement appelé `fdisk` (ou `FDISK.EXE`) dans quasiment tous les OS. Certains `fdisk`, dont celui du DOS, sont quelque peu limités pour gérer les partitions d'autres systèmes d'exploitation. Parmi ces limites, l'impossibilité de prendre en compte tout ce qui est identifié par un code de type étranger, l'impossibilité de prendre en compte plus de 1024 cylindres, et l'impossibilité de créer ou même de reconnaître une partition dont la fin ne coïncide pas avec la borne d'un cylindre. Par exemple, le `fdisk` de MS-DOS ne peut pas supprimer les partitions NTFS, le `fdisk` de OS/2 était réputé pour "corriger" silencieusement les partition crées par le `fdisk` de Linux dont la fin ne coïncidait pas avec une borne de cylindre, et tant le `fdisk` de MS-DOS que celui de OS/2 ont eu des problèmes avec les disques de plus de 1024 cylindres (reportez-vous au "large-disk Mini-HOWTO" pour de plus amples détails sur ces disques).

3 De quelles partitions ai-je besoin ?

3.1 De combien de partitions ai-je besoin ?

Donc, de quelles partitions ai-je besoin ? Pour commencer, certains systèmes d'exploitation ne croient pas au démarrage à partir de partitions logiques pour des raisons qui sont à la portée de tout esprit sain. De ce fait, vous voudrez certainement réserver vos partitions primaires comme partitions d'amorçage pour MS-DOS, OS/2 et Linux ou pour quelque autre système que vous utilisiez. Rappelez-vous toutefois qu'une partition primaire est nécessaire pour créer la partition étendue qui servira de container pour les partitions logiques qui occuperont le reste de votre disque.

L'amorçage des systèmes d'exploitation se passe en mode réel et implique toutes les limitations liées au BIOS, et surtout celle des 1024 cylindres. Vous voudrez donc probablement placer toutes vos partitions de démarrage dans les 1024 premiers cylindres de votre disque dur, afin d'éviter des complications. A nouveau, je vous invite à lire le "large-disk Mini-HOWTO" pour les détails saignants.

Pour installer Linux, vous aurez besoin d'au moins une partition. Si le noyau est chargé depuis cette partition (par exemple grâce à LILO), cette partition doit être lisible du BIOS. Si vous chargez votre noyau par d'autres moyens (par exemple depuis une disquette d'amorçage ou avec LOADLIN.EXE, le lanceur de Linux depuis MS-DOS), cette partition peut être n'importe où. Dans tous les cas, le type de cette partition sera "Linux native", code 0x83.

Votre système aura besoin d'espace swap. A moins de swaper sur des fichiers, il vous faudra une partition swap dédiée. Du fait que ce type de partition n'est accessible que par le noyau de Linux, et que ce noyau n'est pas affecté par les déficiences du BIOS de votre PC, la partition swap peut être installée n'importe où. Je recommande d'utiliser pour cela une partition logique (/dev/?d?5 ou une des suivantes). Les partitions swap dédiées de Linux sont de type "Linux swap", code 0x82.

Ces exigences sont le minimum en terme de partitions. Il peut toutefois se révéler utile de créer plus de partitions pour Linux, comme la suite le montrera.

3.2 Quelle taille attribuer à ma zone swap ?

Si vous avez décidé d'utiliser une partition dédiée à la zone swap, ce qui est une Bonne Idée [tm], considérez les indications suivantes pour estimer sa taille :

- Sous Linux, la taille de la RAM et celle de la zone swap s'additionnent (ce qui n'est pas vrai pour tous les Unix). Par exemple, si vous avez 8 Mo de RAM et 12 Mo de swap, vous disposez d'un total d'environ 20 Mo de mémoire virtuelle.

- En choisissant la taille de votre zone swap, gardez présent à l'esprit que vous devriez disposer d'au moins 16 Mo de mémoire virtuelle. Ainsi pour 4 Mo de RAM envisagez un minimum de 12 Mo de swap ; pour 8 Mo de RAM, envisagez un minimum de 8 Mo de swap.

- Sous Linux, une partition swap ne peut pas excéder 128 Mo. En réalité, sa taille pourrait dépasser 128 Mo, mais l'espace en excès ne serait jamais utilisé. Si vous voulez plus de 128 Mo de swap, vous devez créer plusieurs partitions swap.

- En choisissant la taille de votre zone swap, rappelez vous qu'une zone swap trop grande ne sera pas vraiment utile.

  Tout processus possède un "jeu d'instructions" qui correspond à un ensemble de pages mémoire, et auquel le processeur accédera à nouveau dans un temps très court. Linux essaie de prévoir ces accès mémoire (en partant du principe que les pages récemment utilisées le seront à nouveau dans un futur proche) et conserve ces pages dans la RAM si c'est possible. Si le programme respecte strictement le principe de localité, cette hypothèse sera vérifiée, et l'algorithme de prédiction fonctionnera.

  Conserver en mémoire une zone de travail n'a de signification que s'il y a suffisamment de mémoire. Si trop de processus s'exécutent en même temps sur une même machine, le noyau est alors dans l'obligation de paginer des données auxquelles il devra accéder de nouveau très rapidement (il faudra donc paginer sur disque des données provenant d'une autre zone de travail pour pouvoir les appeler en mémoire). Ceci induit généralement une augmentation critique de l'activité de pagination, et donc une substantielle baisse de performances. On dit d'une machine dans cette situation qu'elle "rame".

  Sur une machine qui rame, les processus tournent essentiellement sur disque, et non dans la RAM. On peut donc s'attendre à une chute de performances de l'ordre de grandeur du rapport entre le temps d'accès mémoire et le temps d'accès disque.

  Mon petit doigt m'a parlé d'une très vieille règle datant de l'époque du PDP et du Vax, et qui est la suivante : la taille du jeu d'instructions d'un programme est égale à environ 25 % de sa taille virtuelle. Ainsi, il est sans doute inutile de prévoir plus de swap que trois fois la taille de votre RAM.

  Mais rappelez-vous que c'est seulement mon petit doigt qui me l'a dit. On peut facilement imaginer des cas ou les programmes ont un très grand, ou au contraire un très petit jeu d'instructions. Par exemple, un programme de simulation avec un très grand jeu de données auxquelles il accède de manière quasi aléatoire ne respectera pas vraiment le principe de localité dans son segment de données, et donc son jeu d'instructions sera relativement important.

  D'un autre côté, `xv` avec de nombreux JPEGs ouverts simultanément, mais tous iconifiés sauf un, aura un très gros segment de données. Mais les opérations ne sont faites que sur une seule image à la fois, et donc la plus grande partie de la mémoire utilisée par xv n'est jamais accédée. C'est également vrai dans le cas d'un éditeur multi-fenêtres où seule une page à la fois est active. Ces programmes - s'ils sont conçus correctement - respectent rigoureusement le principe de localité, et la plus grande partie de la place qu'ils occupent peut rester dans la swap sans qu'on observe de diminution substantielle des performances.

  On peut suspecter que ce chiffre de 25 % datant de l'époque de la ligne de commande n'est plus vrai pour les logiciels modernes dotés d'une IHM graphique et capables d'éditer simultanément plusieurs documents, mais je n'ai connaissance d'aucune donnée récente permettant d'actualiser ces chiffres.

En résumé, si on dispose de 16 Mo de RAM, un configuration minimale peut se passer de swap, et attribuer plus de 48 Mo à la swap est sans doute inutile. L'appoint exact de mémoire requise dépend des applications qui tournent sur la machine (qu'est-ce que vous vous étiez imaginé ?).

3.3 Où positionner ma zone swap ?

- Les mouvements mécaniques sont lents, et les mouvements électroniques rapides.

  Les disques récents on plusieurs têtes de lecture. Permuter entre les têtes qui se trouvent sur la même piste est rapide, puisque c'est purement électronique. Par contre changer de piste est lent, puisque ça implique un mouvement des têtes.

  Par conséquent si vous avez un disque avec plusieurs têtes de lecture et un autre qui en a moins, les autres paramètres étant identiques, le disque qui a le plus de têtes de lectures sera le plus rapide.

  Découper la zone swap en la répartissant sur les disques accélérera aussi la vitesse d'accès.

- Les anciens disques ont le même nombre de secteurs sur toutes les pistes. Avec ce type de disque, la vitesse maximum est généralement obtenue en plaçant la zone swap au milieu du disque, si on part du principe que la tête de lecture devra se déplacer d'une piste quelconque vers l'emplacement physique de la zone swap.

- Les disques plus récents utilisent le ZBR (bit d'enregistrement de zone). Les pistes externes contiennent un plus grand nombre de secteurs. Pour une vitesse de rotation constante, on obtient donc un bien meilleur rendement pour les pistes externes que pour les pistes internes. Placer de préférence votre zone swap sur les pistes les plus rapides.

- Mais bien sûr, la tête de lecture n'est pas animée de mouvement aléatoires. Si le milieu du disque tombe entre une partition `/home` en accès constant et une partition d'archivage presque jamais utilisée, vous feriez mieux de placer votre zone swap au milieu de la partition `/home`, pour limiter l'amplitude de mouvement des têtes de lecture. Le mieux, dans ce cas, serait même de placer votre zone swap sur un autre disque, moins activement utilisé.

En résumé : Placez votre zone swap sur un disque rapide équipé de plusieurs têtes de lecture et qui n'est pas trop accaparé par d'autres tâches. Si vous avez plusieurs disques, répartissez la zone swap sur tous ces disques, même si leurs contrôleurs sont différents.

Encore mieux : Achetez plus de RAM.

3.4 Quelques bricoles au sujet des systèmes de fichiers et de la fragmentation

L'espace disque est administré par le système d'exploitation en unités de blocs et fragments de blocs. En ext2, fragments et blocs doivent être de la même taille, aussi nous limiterons la discussion aux blocs.

Les fichiers ont des tailles très variables qui ne coïncident pas nécessairement avec la fin d'un bloc. Par conséquent, pour chaque fichier, un partie du dernier bloc est gaspillée. Supposons que la taille des fichiers soit aléatoire, il y a en moyenne un demi-bloc perdu pour chaque fichier présent sur le disque. Dans son livre "Operating systems", Tanenbaum appelle ça la "fragmentation interne".

On peut déduire le nombre de fichiers présents sur le disque à partir du nombre d'inodes alloués. Par exemple sur mon disque :

```
# df -i
Filesystem           Inodes   IUsed   IFree  %IUsed Mounted on
/dev/hda3              64256   12234   52022    19%  /
/dev/hda5              96000   43058   52942    45%  /var
```

Il y a donc environ 12000 fichiers sur `/` et près de 44000 sur `/var`. Pour des blocs d'une taille de 1 Ko, à peu près 6+22 = 28 Mo d'espace disque sont perdus dans les derniers blocs des fichiers. Si j'avais choisi des blocs d'une taille de 4 Ko, j'aurais perdu 4 fois plus de place.

Les transferts de données sont plus rapides avec de grands tronçons contigus de données. C'est pourquoi l'ext2 s'efforce de pré-allouer l'espace en unités de 8 blocs contigus pour les fichiers en cours d'écriture. L'espace pré-alloué non utilisé est libéré lors de la fermeture du fichier, ainsi il n'y a pas de gaspillage.

Un rangement non contigu des blocs dans un fichier est préjudiciable pour les performances, du fait qu'on accède généralement aux fichiers de manière séquentielle. Cela oblige le système d'exploitation à découper les accès disque et le disque à déplacer la tête de lecture. On appelle cela la "fragmentation externe", ou simplement la "fragmentation", qui est un problème courant avec les systèmes de fichiers de type DOS.

ext2 utilise plusieurs stratégies afin d'éviter la fragmentation externe. Normalement la fragmentation n'est pas un gros problème en ext2, même avec des partitions très utilisées, comme une file d'attente news. Bien qu'il existe un utilitaire de défragmentation des systèmes de fichier ext2, personne ne l'utilise et il n'est pas à jour avec la dernière version de ext2. Utilisez le si vous y tenez, mais à vos risques et périls.

Le système de fichiers MS-DOS est réputé pour sa gestion pathologique de l'espace disque. La conjugaison d'un cache tampon abyssal et de la fragmentation a des conséquences tout à fait dommageables sur les performances. Les utilisateurs de DOS sont habitués à défragmenter leurs disques toutes les quelques semaines et certains ont même mis au point un rituel quasi religieux concernant la défragmentation. Aucune de ces habitudes ne devrait être transposée sous Linux et ext2. Le système de fichiers natif de Linux n'a pas besoin de défragmentation en utilisation normale, ce qui inclut n'importe quelle condition du moment que 5 % de l'espace disque reste libre.

Le système de fichiers MS-DOS est aussi réputé pour perdre une grande quantité d'espace disque en raison de la fragmentation interne. Pour des partitions d'une taille supérieure à 256 Mo, la taille des blocs DOS devient si importante qu'ils ne sont plus d'aucune utilité (cela a été corrigé jusqu'à un certain point avec la FAT32).

ext2 ne force pas l'utilisation de grands blocs dans le cas de grand systèmes de fichiers, à l'exception des très grands systèmes de fichier de l'ordre de 0.5 To (1 Tera-octet = 1024 Go) et plus, pour lesquels les blocs de petite taille deviennent inefficaces. Donc, contrairement au DOS, il n'est pas nécessaire de découper les grands disques en plusieurs partitions pour conserver des blocs de petite taille. Dans la mesure du possible, utilisez la taille par défaut de 1 Ko. Vous voudrez peut être expérimenter des blocs de 2 Ko pour certaines partitions, mais attendez vous à rencontrer quelques bugs peu courants : presque tout le monde utilise la taille par défaut.

3.5 Durée de vie des fichiers et cycles de sauvegarde sont des critères dans le choix des partitions

Sous ext2, les décisions concernant le choix des partitions devraient être dirigées par des considérations liées aux sauvegardes, et de manière à éviter la fragmentation externe due aux durées de vie des différents fichiers.

Les fichiers ont une durée de vie. Une fois créé, un fichier restera un certain temps sur le système avant d'être supprimé. La durée de vie des fichiers varie considérablement au sein du système, et dépend en partie du chemin d'accès du fichier. Par exemple, les fichiers présents dans `/bin`, `/sbin`, `/usr/sbin`, `/usr/bin` ou quelqu'autre répertoire du même type ont une durée de vie très longue : de nombreux mois, voire plus. Les fichiers présents dans `/home` ont une durée de vie intermédiaire : à peu près quelques semaines. Les fichiers présents dans `/var` ont généralement une durée de vie courte : quasiment aucun fichier dans `/var/spool/news` ne restera plus de quelques jours, et dans `/var/spool/lpd` le temps de vie se mesure en minutes voire moins.

Pour sauvegarder, il peut être utile de s'assurer que la taille d'une sauvegarde journalière reste inférieure à la taille du support de sauvegarde. Une sauvegarde journalière peut être complète ou différentielle.

Vous pouvez décider de conserver des tailles de partitions suffisamment petites pour tenir complètement sur un seul support de sauvegarde (auquel cas, faites des sauvegardes journalières complètes). Dans tous les cas, la taille d'une partition devrait être telle que son "delta" journalier (tous les fichiers modifiés) puisse tenir sur un seul support de sauvegarde (faites une sauvegarde différentielle, et prévoyez de changer le support pour la sauvegarde hebdomadaire/mensuelle complète).

Votre stratégie de sauvegarde repose sur ces décisions.

Lorsque vous achetez et organisez de l'espace disque, pensez à mettre de coté une somme suffisante pour les sauvegardes afférentes ! Des données non sauvegardées sont sans valeur ! Le coût de reproduction des données est de loin plus élevé que celui de la sauvegarde, pour qui que ce soit !

Pour des raisons de performances, il est utile de conserver des fichiers ayant des durées de vie différentes sur des partitions différentes. De cette manière, les fichiers éphémères de la partition `.../news` peuvent être très lourdement fragmentés. Cela n'aura aucune incidence sur les performances des partitions `/` ou `/home`.

4 Un exemple

4.1 Un modèle à suivre pour débutant ambitieux

Un modèle courant propose la création des partitions `/`, `/home` et `/var` pour des raisons abordées plus haut. Cela simplifie tant l'installation que la maintenance, et la différenciation est suffisante pour éviter les effets pervers des durées de vie différentes. C'est aussi un bon modèle en ce qui concerne la sauvegarde : personne ne se soucie de sauvegarder les files d'attente "news" et seulement quelques fichiers de `/var` peuvent être utilement sauvegardés (comme `/var/spool/mail`). D'un autre coté, `/` change très peu souvent et peut n'être sauvegardé que ponctuellement (après un changement de configuration), et sa taille relativement faible permet, pour la plupart des supports modernes, de faire une sauvegarde complète (prévoyez de 250 à 500 Mo en fonctions des logiciels installés). `/home` contient les précieuses données des utilisateurs et devrait être sauvegardé chaque jour. Certaines configurations présentent un `/home` très important et doivent par conséquent faire appel au sauvegardes différentielles.

Certains systèmes prévoient une partition séparée pour `/tmp`, d'autres créent un lien symbolique sur `/var/tmp` pour obtenir un résultat similaire (notez que cela peut affecter le mode "single user" pour lequel `/var` ne sera pas disponible, à moins de le créer ou de le monter manuellement) ; ou encore placez le sur disque RAM (comme c'est le cas sous Solaris). Cela tient `/tmp` séparé de `/`, ce qui es une bonne idée.

Ce modèle est tout à fait adapté aux mises à jour ou aux réinstallations : sauvez vos fichiers de configuration (ou la totalité de `/etc`) dans un répertoire de `/home`, débarrassez vous de `/`, réinstallez et récupérez votre ancienne configuration à partir du répertoire de sauvegarde sur `/home`.

5 Comment je m'y suis pris personnellement

Un vieux 386/40 sur bus ISA traînait sur mon étagère depuis deux ans. J'avais l'intention de le transformer en un petit serveur non-X pour mon réseau local.

Voici comment je m'y suis pris : j'ai récupéré ce 386 et l'ai doté de 16 Mo de RAM. J'y ai ajouté le disque le moins cher et le plus petit que j'ai pu trouver (800 MB), une carte Ethernet et une vieille Hercules parce que j'avais toujours le moniteur. J'ai installé Linux, ce qui m'a permis de disposer d'un serveur NFS, SMB, HTTP, LPD/LPR et NNTP familial ainsi que d'un routeur mail et d'un serveur POP3. Avec en plus une carte RNIS, cette machine me sert maintenant en plus de routeur TCP/IP et de pare-flamme.

L'essentiel de l'espace disque sur cette machine est passé dans les répertoires de `/var`, `/var/spool/mail`, `/var/spool/news` et `/var/httpd/html`. J'ai placé `/var` sur un partition séparée, que j'ai créée suffisamment grande. Comme il n'y aura autant dire pas d'utilisateurs sur cette machine, je n'ai pas créé de partition home, et j'ai donc monté `/home` depuis une autre station de travail via NFS.

Une partition / de 250 Mo est amplement suffisante pour Linux sans X, doté de quelques utilitaires locaux supplémentaires. Cette machine a 16 Mo de RAM, mais elle est destinée à piloter de nombreux serveurs. 16 Mo de swap serait correct, 32 Mo l'abondance. Comme l'espace disque le permet, disons 32 Mo de swap. Conservons une partition MS-DOS de 20 Mo. Comme j'ai décidé d'importer `/home` depuis une autre machine, les 500+ Mo constitueront `/var`. C'est plus que suffisant pour un centre de distribution de news familial.

Nous avons donc :

```
Device       Mounted on                     Size
/dev/hda1    /dos_c                          25 MB
/dev/hda2    - (Swapspace)                   32 MB
/dev/hda3    /                              250 MB
/dev/hda4    - (Extended Container)         500 MB
/dev/hda5    /var                           500 MB

homeserver:/home /home                      1.6 GB
```

J'effectue les sauvegardes de cette machine via le réseau en utilisant le lecteur de bande de `homeserveur`. Du fait que l'installation a été faite à partir d'un CDROM, je n'ai besoin de sauvegarder que quelques fichiers de `/etc`, mes fichiers *.tgz personnalisés installés localement sur `/root/Source/Installed` et `/var/spool/mail` ainsi que `/var/httpd/html`. Je copie chaque nuit ces fichiers dans un répertoire dédié `/home/backmeup` sur `homeserver`, où la sauvegarde régulière de `homeserver` les récupère.