Linux Partitie HOWTO

Kristan Koehntopp, kris@koehntopp.de,
Vertaald door Ellen Bokhorst, bokkie@nl.linux.org v2.4, 3 november 1997

Deze Linux Mini-HOWTO leert je hoe diskruimte voor je Linux systeem te plannen en in te delen. Het gaat over disk hardware, partities, grootte ramen van swap space en positioneringsoverwegingen. Bestandssystemen, bestandssysteemtypes en daaraan gerelateerde onderwerpen. De bedoeling is je wat achtergrondkennis bij te brengen, geen procedures.

Inhoudsopgave

1 Introductie

1.1 Wat is dit?

Dit is een Linux Mini-HOWTO tekst. Een Mini-HOWTO is een beknopte tekst die een aantal zaken in een leerprogramma-stijl uitlegt, en gerelateerd is aan installatie en beheer. Het is mini, omdat de tekst of het onderwerp, dat erin wordt besproken, te klein is voor een echte HOWTO of zelfs een boek. Een HOWTO is geen referentie: daar zijn manual pages voor.

1.2 Wat staat er in? en gerelateerde HOWTO documenten

Deze Mini-HOWTO leert je hoe diskruimte te plannen en in te delen voor je Linux systeem. Het gaat over disk hardware, partities, grootte ramen van swapspace en positioneringsoverwegingen, bestandssystemen, bestandssysteemtypes en gerelateerde onderwerpen. De bedoeling is je wat achtergrondkennis bij te brengen. We bespreken in deze tekst hoofdzakelijk grondbeginselen en geen tools.

Het beste zou je dit document kunnen lezen vóór je eerste installatie, maar dit is voor de meeste mensen wat lastig. Beginnelingen hebben naast optimalisatie van de disk-indeling, nog andere problemen. Dus waarschijnlijk ben je iemand die net een Linux installatie heeft uitgevoerd en nu manieren aan het bedenken is, om deze installatie te optimaliseren, of hoe je lastige miscalculaties in de volgende installatie kan voorkomen. Ga er maar vanuit dat je je huidige installatie opnieuw wilt doen, als je deze tekst gelezen hebt. :-)

Deze Mini-HOWTO beperkt zichzelf de meeste tijd tot het plannen en indelen van diskruimte. Het bespreekt niet het gebruik van fdisk, LILO, mke2fs of backup programma's. Er zijn andere HOWTO's die deze problemen aanspreken. Bekijk alsjeblieft de Linux HOWTO Index voor actuele informatie over Linux HOWTO's. Er staan ook instructies in de index hoe je de HOWTO documenten kunt verkrijgen.

Zie de "Linux Multiple Disks Layout mini-HOWTO", door Gjoen Stein <gjoen@nyx.net> om te leren hoe je de diverse grootte en snelheidsvereisten voor verschillende delen van het bestandsssysteem kunt ramen.

Zie de "Linux Large Disk mini-HOWTO", door Andries Brouwer <aeb@cwi.nl> voor instructies en overwegingen met betrekking tot disks met meer dan 1024 cylinders.

Zie de "Linux Quota mini-HOWTO"door Albert M.C. Tam <bertie@scn.org> voor instructies over het beperken van diskruimte per gebruiker (quota's).

Op dit moment is er geen algemeen document over diskbackup, maar er zijn verscheidene documenten met verwijzingen naar specifieke backup oplossingen. Zie de "Linux ADSM Backup mini-HOWTO", door Thomas Koenig <Thomas.Koenig@ciw.uni-karlsruhe.de> voor instructies over het integreren van Linux in een IBM ADSM backup omgeving. Zie de "Linux Backup with MSDOS mini-HOWTO", door Christopher Neufeld <neufeld@physics.utoronto.ca> voor informatie over MS-DOS bestuurde Linux backups.

Zie de Linux HOWTO Index, door Tim Bynum <linux-howto@sunsite.unc.edu> voor instructies over het schrijven en aanleveren van een HOWTO document.

Bladeren door /usr/src/linux/Documentation kan ook erg leerzaam zijn. Zie ide.txt en scsi.txt voor wat achtergrondinformatie over de kenmerken van je diskdrivers en kijk eens in de filesystems/ subdirectory.

2 Wat is een partitie eigenlijk?

Toen PC harddisks werden uitgevonden, wilden mensen al spoedig meerdere besturingssystemen installeren, ook al had hun systeem maar één disk. Dus er was een mechanisme nodig om een enkele fysieke disk in meerdere logische disks in te delen. Dus dat is wat een partitie is: Een opeenvolgend gedeelte van je harddisk die door de meeste besturingssystemen als een complete afzonderlijke disk wordt behandeld.

Het is tamelijk duidelijk dat partities elkaar niet moeten overlappen: Een besturingssysteem zal er beslist niet blij mee zijn, als een ander geïnstalleerd besturingssysteem op dezelfde machine belangrijke informatie zou overschrijven vanwege overlappende partities. Er zou ook geen gat tussen aangrenzende partities moeten zitten. Ondanks dat deze constellatie niet schadelijk is, verspil je kostbare diskruimte door ruimte tussen partities te laten.

Een disk hoeft niet volledig gepartitioneerd te zijn. Het kan zijn dat je besluit om wat ruimte aan het einde van je disk te laten die nog niet aan één van je geïnstalleerde besturingssystemen is toegewezen. Als het je later duidelijk is welke installatie de meeste tijd wordt gebruikt, kun je deze overgebleven ruimte partitioneren en er een bestandssysteem op plaatsen.

Partities kunnen niet worden verplaatst, noch kan de grootte worden gewijzigd zonder dat je het bestandssysteem wat zich erop bevindt, wordt verwijderd. Dus het herpartitioneren brengt gewoonlijk het backuppen en herstellen van alle, tijdens het herpartitioneren, betrokken bestandssystemen met zich mee. In feite is het tamelijk gewoon gedurende het herpartitioneren zaken compleet te verknoeien, dus je zou van alles op iedere disk op die machine een backup moeten maken voordat je zelfs ook maar aan zoiets als `fdisk` komt.

Een aantal partities met daarop bepaalde bestandssysteemtypes *kunnen* in werkelijkheid in tweeën worden gesplitst zonder gegevens te verliezen (als je geluk hebt). Er is bijvoorbeeld een programma met de naam "fips"voor het in tweeën splitsen van MS-DOS partities voor een Linux installatie, zonder dat MS-DOS opnieuw moet worden geïstalleerd. Je raakt deze dingen toch niet aan zonder alles op deze machine zorgvuldig te backuppen, nietwaar?

2.1 Backups zijn belangrijk

Tapes zijn je vriend voor backups. Ze zijn snel, betrouwbaar en gemakkelijk te gebruiken, dus je kunt vaak backups maken, bij voorkeur automatisch en zonder gedoe.

En ik heb het over echte, niet die disk controller bestuurde ftape troep. Overweeg het kopen van SCSI: Linux ondersteunt intern SCSI. Je hoeft geen ASPI drivers te laden, je verliest geen kostbare HMA onder Linux en zodra de SCSI hostadapter éénmaal is geïnstalleerd, sluit je er gewoon extra disks, tapes en CD-ROM's op aan. Geen I/O adressen meer, IRQ gegoochel met Master/Slave en PIO-level overeenstemming.

Plus: Goede SCSI-hostadapters geven je een hoge I/O performance zonder veel CPU load. Zelfs onder zware diskactiviteit zul je goede responsetijden ervaren. Als je van plan bent om je Linux-systeem hoofdzakelijk als USENET nieuwstoevoer te gebruiken of als je op het punt staat in ISP-bedrijf te gaan, denk er dan zelfs niet aan om een systeem zonder SCSI neer te zetten.

2.2 Nummers en namen van apparaten

Het aantal partities op een systeem gebaseerd op Intel was vanaf het allereerste begin beperkt. De originele partitietabel werd geïnstalleerd als onderdeel van de bootsector en gaf ruimte voor slechts vier partitie-ingangen. Deze partities worden nu primaire partities genoemd. Toen het duidelijk werd dat mensen meer partities op hun systemen nodig hadden, werden de logische partities uitgevonden. Het aantal logische partities is niet beperkt: Iedere logische partitie bevat een verwijzing naar de volgende logische partitie, dus je kunt een mogelijk ongelimiteerde reeks partitie-ingangen hebben.

Om compatibiliteits redenen, moest de ruimte die door alle logische partities in beslag werd genomen, worden verantwoord. Als je logische partities gebruikt, wordt één partitie als üitgebreide partitie"gemarkeerd, en het begin- en eindblok markeren het gebied dat door de logische partities in beslag wordt genomen. Dit impliceert dat de ruimte die aan alle logische partities is toegekend, aaneengesloten moet zijn. Er kan slechts één uitgebreide partitie voorkomen: geen enkel `fdisk` programma zal meer dan één uitgebreide partitie aanmaken.

Linux kan niet meer dan een beperkt aantal partities per drive beheren. Dus als je in Linux 4 primaire partities hebt (zijn 3 daarvan te gebruiken, als je gebruik maakt van logische partities) en alles bij elkaar maximaal 15 partities op een SCSI disk (alles bij elkaar 63 op een IDE disk).

Onder Linux worden partities voorgesteld door apparaatbestanden. Een apparaatbestand is een bestand met type c (voor "characteräpparaten, apparaten die geen buffercache gebruiken) of b (voor "blokäpparaten, die door de buffercache gaan). Onder Linux worden alle disks alleen voorgesteld als blokapparaten. In tegenstelling tot Unices, biedt Linux geen "raw"character versies van disks en de partities daarop.

Het enige belangrijke aan een apparaatbestand zijn z'n major en minor devicenummers, die in plaats van de bestandsgroottes worden getoond:

```
$ ls -l /dev/hda
brw-rw----   1 root     disk       3,    0 Jul 18  1994 /dev/hda
                                   ^     ^
                                   |     minor devicenummer
                                   major device nummer
```

Bij het benaderen van een apparaatbestand, selecteert het major nummer welke driver wordt aangeroepen om de input/output bewerking uit te voeren. Deze aanroep wordt gedaan met het secundaire nummer als een parameter en het is helemaal aan de driver hoe het secundaire nummer wordt geïnterpreteerd. De driver-documentatie beschrijft gewoonlijk hoe de driver secundaire nummers gebruikt. Voor IDE disks, staat deze documentatie in `/usr/src/linux/Documentation/ide.txt`. Voor SCSI disks, zou je dergelijke documentatie verwachten in `/usr/src/linux/Documentation/scsi.txt`, maar daar staat het niet. Men moet bij de source van de driver kijken om zeker te zijn (`/usr/src/linux/driver/scsi/sd.c:184-196`). Gelukkig bestaat er de lijst met devicenummers en namen van Peter Anvin in `/usr/src/linux/Documentation/devices.txt`; zie de ingangen voor blokapparaten, major 3, 22, 33, 34 voor IDE en major 8 voor SCSI disks. De major en minor nummers zijn ieder een byte groot en daarom is het aantal partities per disk beperkt.

Volgens afspraak hebben apparaatbestanden bepaalde namen en veel systeemprogramma's hebben de kennis over deze namen ingecompileerd. Ze verwachten dat je IDE disks `/dev/hd*` zijn genoemd en je SCSI disks `/dev/sd*`. Disks zijn genummerd a, b, c en zo verder, dus `/dev/hda` is je eerste IDE disk en `/dev/sda` is je eerste SCSI disk. Beide devices stellen gehele disks voor, beginnend op blok één. Met de verkeerde tools naar deze tools schrijven zal de masterbootloader en partitietabel op deze disks ruïneren, onbruikbare weergave van alle gegevens op de disk of je systeem onopstartbaar maken. Weet wat je doet en, nogmaals, maak een backup voordat je het doet.

Primaire partities op een disk zijn 1, 2, 3 en 4. Dus `/dev/hda1` is de eerste primaire partitie op de eerste IDE disk en zo verder. Logische partities hebben de nummers 5 en hoger, dus `/dev/sdb5` is de eerste logische partitie op de tweede SCSI disk.

Aan iedere partitie-ingang is een begin- en een eind-blokadres en een type toegewezen. Het type is een numerieke code (een byte) die een bepaalde partitie aan een bepaald type besturingssysteem toewijst. Ten bate van computeradviseurs zijn partitie type codes niet echt uniek, dus er is altijd een kans dat twee besturingssystemen dezelfde code gebruiken.

Linux reserveert het type code 0x82 voor swappartities en 0x83 voor "native"bestandssystemen (dat is ext2 voor de meeste van jullie). Het ooit populaire, nu verouderde Linux/Minix bestandssysteem gebruikte het type code 0x81 voor partities. OS/2 markeert zijn partities met een 0x07 type en dit doet NTFS van Windows NT ook. MS-DOS neemt verscheidene type codes in beslag voor z'n diverse soorten FAT bestandssystemen: 0x01, 0x04 en 0x06 zijn bekend. DR-DOS gebruikte 0x81 om beschermde FAT partities aan te geven, daarmee in die tijd een conflict genererend met Linux/Minix, maar noch Linux/Minix noch DR-DOS worden nog veel gebruikt. Tussen twee haakjes, de uitgebreide partitie die als een container voor logische partities wordt gebruikt, is van het type 0x05.

Partities worden aangemaakt en verwijderd met het `fdisk` programma. Ieder zelfrespecterend besturingssysteem komt met een `fdisk` en wordt traditioneel onder bijna alle Os'sen zelfs `fdisk` genoemd (of `FDISK.EXE`) Sommige `fdisks`, merkbaar die van DOS, zijn op één of andere manier beperkt als ze met partitities van andere besturingssystemen om moeten gaan. Een van dergelijke beperkingen is de complete onmogelijkheid om te gaan met alles met een type code van een ander bestandssysteem, de onmogelijkheid om te gaan met cylinder nummers boven 1024 en de onmogelijkheid partities aan te maken of zelfs te begrijpen die niet op een cylinder begrenzing eindigen. Bijvoorbeeld, de MS-DOS fdisk kan geen NTFS partities verwijderen, van de OS/2 fdisk is bekend dat ze in stilte door Linux aangemaakte partities "corrigeerdeäls ze niet eindigden op een cylinder begrenzing en zowel, de DOS en de OS/2 fdisk, hadden problemen met disks met meer dan 1024 cylinders (zie de "large-disk"Mini-Howto voor details over dergelijke disks).

3 Wat voor Partities heb ik nodig

3.1 Hoeveel partities heb ik nodig?

Oke, dus welke partities heb je nodig? Goed, een aantal besturingssystemen geloven niet in het booten vanaf logische partities om volstrekt onbegrijpelijke redenen. Dus je wilt waarschijnlijke je primaire partities reserveren als bootpartities voor je MS-DOS, OS/2 en Linux of wat je ook gebruikt. Denk eraan dat één partitie benodigd is voor een uitgebreide partitie, die als een container optreedt voor de rest van je disk met logische partities.

Het booten van besturingssystemen is een real-mode aangelegenheid die BIOS'sen en 1024 cylinder beperkingen met zich meebrengt. Dus je wilt waarschijnlijk al je bootpartities in de eerste 1024 cylinders van je harddisk plaatsen, gewoon om problemen te voorkomen. Nogmaals, lees de "large-disk"Mini-Howto voor de bloedige details.

Om Linux te installeren heb je op z'n minst één partitie nodig. Als de kernel vanaf deze partitie wordt geladen (bijvoorbeeld door LILO), moet deze partitie leesbaar zijn door je BIOS. Als je andere mogelijkheden gebruikt om je kernel te laden (bijvoorbeeld een opstartdiskette of LOADLIN.EXE, een op MS-DOS gebaseerde Linux loader) kan de partitie zich overal bevinden. In ieder geval zal deze partitie van het type 0x83 "Linux native"zijn.

Je systeem zal wat swapspace nodig hebben. Tenzij je naar bestanden swapt, zul je een hiervoor toegewezen swappartitie nodig hebben. Aangezien deze partitie alleen door de Linux kernel wordt benaderd en de Linux kernel niet aan PC BIOS ontoereikendheden leidt, kan de swap partitie overal worden gepositioneerd. Ik raad je aan er een logische partitie voor te gebruiken (/dev/?d?5 en hoger). Toegewezen Linux swap partities zijn van het type 0x82 "Linux swap".

Dit zijn minimale partitie vereisten. Het zou handig kunnen zijn om meer partities voor Linux aan te maken. Lees verder.

3.2 Hoe groot zou mijn swapspace moeten zijn?

Als je hebt besloten om een toegewezen swap partitie te gebruiken, wat in het algemeen een Goed Idee [tm] is, volg dan deze richtlijnen voor het ramen van z'n grootte:

- Onder Linux worden RAM en swap space bijelkaar opgeteld (Dit geldt niet voor alle Unices). Als je bijvoorbeeld 8 MB RAM en 12 MB swap hebt, heb je een totale hoeveelheid van ongeveer 20 MB virtueel geheugen.
- Wanneer je je swap space passend maakt, zou je in totaal op z'n minst 16 MB virtueel geheugen moeten hebben. Neem dus voor 4 MB RAM tenminste 12 MB swap in overweging, voor 8 MB RAM, 12 MB swap, neem voor 8 MB RAM op z'n minst 8 MB swap in overweging.
- Onder Linux, kan een enkele swappartitie niet groter zijn dan 128 MB. Dat wil zeggen, de partitie kan wel groter zijn dan 128 MB, maar de toegankelijke ruimte wordt nooit gebruikt. Als je meer dan 128 MB swap wilt, zul je meerdere swap-partities aan moeten maken.
- Als je de grootte van swap space bepaalt, houd dan in gedachten dat te veel swapspace helemaal niet handig kan zijn.

  Ieder proces heeft een "werkset". Dit is een set van pagina's in het geheugen waarnaar in de zeer nabije toekomst door de processor zal worden gerefereerd. Linux probeert deze geheugentoegangen te voorspellen (ervan uitgaande dat recent gebruikte pagina's in de nabije toekomst opnieuw zullen worden gebruikt) en houdt deze pagina's zo mogelijk in RAM. Als het programma een goede "referentieligging"heeft, zal deze voorspelling kloppen en het voorspellings algoritme zal werken.

Het in het hoofdgeheugen houden van een werkset, werkt alleen als er voldoende hoofdgeheugen is. Als je te veel processen op een machine hebt draaien, wordt de kernel gedwongen om pagina's op disk te plaatsen waarnaar het opnieuw in de zeer nabije toekomst zal refereren (waarbij een pagina gedwongen uit een andere werkset uit het geheugen wordt gehaald en vervolgens de pagina waarnaar werd gerefereerd in het geheugen wordt geladen). Gewoonlijk resulteert dit in een zeer zware verhoging van paging activiteit en in een aanzienlijke performance vermindering. Van een machine in deze staat wordt gezegd dat er överheadïs. (Voor jullie duitse lezers: Dat is "thrashing"("dreschen", "schlagen", "haemmern") en niet vermorzelen ("muellen")).

Op een machine met overhead draaien de processen in wezen vanaf disk en niet vanuit RAM. Ga er maar vanuit dat de performance zal verminderen met bij benadering de verhouding tussen de toegangssnelheid van het geheugen en de toegangssnelheid van de disk.

Een erg oude praktische methode in de dagen van de PDP en de Vax was, dat de grootte van de werkset van een programma uit ongeveer 25% van zijn feitelijke grootte bestaat. Dus het is waarschijnlijk nutteloos om in meer swap te voorzien dan drie maal je RAM.

Maar houd in gedachten dat dit slechts een praktische methode is. Het is makkelijk mogelijk om scenario's te creéren waarin programma's extreem grote of extreem kleine werksets hebben. Bijvoorbeeld, een simulatie programma met een grote gegevensset die op een zeer willekeurige wijze wordt benaderd, zou bijna geen opmerkelijke referentie-ligging in z'n datasegment kunnen hebben, dus zijn werkset zou nogal groot kunnen zijn.

Aan de andere kant, een xv met veel simultaan geopende JPEG's, alle behalve één tot ikoon verkleind, zou een zeer groot datasegment kunnen hebben. Maar afbeeldingtransformaties worden slechts uitgevoerd op één enkele afbeelding, het meeste geheugen dat door xv in beslag wordt genomen wordt nooit aangeroerd. Hetzelfde geldt voor een editor met veel editor vensters waar slechts één venster tegelijk wordt gewijzigd. Deze programma's hebben - als ze op de juiste wijze zijn ontworpen - een zeer hoge referentie-ligging - en grote delen ervan kunnen op disk worden geplaatst zonder een te hevige performance uitwerking.

Men zou kunnen verwachten dat het 25% nummer vanwege de ouderdom van de commando-regel niet langer geldt voor moderne GUI programma's die meerdere documenten wijzigen, maar ik ken geen nieuwere stukken die deze nummers proberen te verifiëren.

Dus voor een configuratie met 16 MB RAM, is geen swap nodig voor een minimale configuratie en meer dan 48 MB swap is waarschijnlijk nutteloos. De exacte hoeveelheid geheugen hangt af van de mix aan applicaties op de machine (wat verwachtte je anders?).

3.3 Waar zou ik mijn swapspace moeten plaatsen?

- Mechanica is langzaam, elektronica is snel.

  Moderne harddisks hebben veel koppen. Het schakelen tussen koppen van hetzelfde spoor gaat snel, aangezien het zuiver elektronisch is. Schakelen tussen sporen gaat langzaam, aangezien het verplaatsen op een totaal andere wijze plaatsvindt.

  Dus als je een disk met veel koppen hebt en één met minder koppen en beide zijn identiek in andere parameters, zal de disk met de vele koppen sneller zijn.

  Het splitsen van de swap en het op beide disks plaatsen zal echter zelfs sneller zijn.

- Oudere disks hebben eenzelfde aantal sectoren op alle sporen. Bij deze disks zal het het snelste zijn als je je swap in het midden van de disks plaatst, ervan uitgaande dat de kop van je disk zich zal verplaatsen vanaf een willekeurig spoor richting het swapgebied.

- Nieuwere disks gebruiken ZBR (zone bit recording). Ze hebben op de buitenste sporen meer sectoren. Met een constant aantal rpms, levert dit een veel grotere performance op, op de buitenste sporen dan op de binnenste. Plaats je swap op de snelste sporen.
- Natuurlijk zal je diskkop niet willekeurig verplaatsen. Als je swap space in het midden van een disk hebt tussen een constant drukke home partitie en een bijna ongebruikte archief partitie, zou je beter af zijn als je swap zich in het midden van de home partitie bevond voor nog kortere kop-bewegingen. Je zou echter zelfs nog beter af zijn, als je swap zich op een andere ongebruikte disk bevond.

Samenvatting: Plaats je swap op een snelle disk met veel koppen die het niet druk heeft met andere zaken. Als je meerdere disks hebt: Splits swap en verspreid het over al je disks of zelfs verschillende controllers.

Zelfs beter: Koop meer RAM.

3.4 Een aantal feiten over bestandsssytemen en fragmentatie

Diskruimte wordt door het besturingssysteem in eenheden blokken en fragmenten blokken beheerd. In ext2 moeten fragmenten en blokken even groot zijn, dus we kunnen deze verhandeling tot blokken beperken.

Bestanden zijn er in elke grootte. Ze eindigen niet op blokgrenzen. Dus met ieder bestand wordt een deel van het laatste blok van ieder bestand verspild. Ervan uitgaande dat de grootte van bestanden willekeurig is, wordt er bij benadering een half blok verspild voor ieder bestand op je disk. Tanenbaum noemt dit ïnterne fragmentatieïn zijn boek "Operating Systems".

Je kunt het aantal bestanden op je disk raden door het aantal in beslag genomen inodes op een disk. Op mijn disk

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

bevinden zich ongeveer 12000 bestanden op `/` en ongeveer 44000 bestanden op `/var`. Bij een blokgrootte van 1 KB, gaat ongeveer 6+22 = 28 MB diskruimte verloren in de laatste blokken van bestanden. Als ik een blokgrootte van 4 KB had uitgekozen, had ik 4 keer zoveel ruimte verloren.

Gegevenstransport is echter sneller voor grote aaneengesloten blokken gegevens. Daarom probeert ext2 ruimte in eenheden van 8 aaneengesloten blokken vooraf toe te wijzen aan groeiende bestanden. De ongebruikte vooraf toegewezen ruimte wordt vrijgegeven als het bestand wordt gesloten, dus er wordt geen ruimte verspild.

Het dwingt het besturingssysteem de toegang naar een disk te splitsen en de disk om de kop te verplaatsen. Dit wordt ëxterne fragmentatie"genoemd of eenvoudigweg "fragmentatieën het is een algemeen probleem met DOS bestandssystemen.

ext2 heeft verscheidene strategieñ om externe fragmentatie te voorkomen. Normale fragmentatie is in ext2 geen groot probleem, zelfs niet op zwaar gebruikte partities zoals een USENET nieuwsspool. Ondanks dat er een tool voor defragmentatie is voor ext2 bestandssystemen, gebruikt niemand het ooit en het is niet bijgewerkt met de huidige release van ext2. Gebruik het op eigen risico.

Het MS-DOS bestandssysteem is goed bekend om z'n pathologische beheer van diskruimte. In samenwerking met de bodemloze buffer cache die door MS-DOS wordt gebruikt, zijn de effecten van bestandsfragmentatie op de performance zeer opmerkelijk. DOS gebruikers zijn gewend aan het iedere paar weken defragmenteren van hun disks en een aantal hebben zelfs een ritueel geloof met betrekking tot defragmentatie ontwikkeld. Geen van deze gewoonten zou naar Linux en ext2 moeten worden overgedragen. Linux native bestandssystemen hoeven niet te worden gedefragmenteerd bij normaal gebruik en dit behelst iedere voorwaarde met tenminste 5% vrije ruimte op een disk.

Het MS-DOS bestandssysteem is ook bekend om het verlies van grote hoeveelheden diskruimte te wijten aan interne fragmentatie. Voor partities groter dan 256 MB, wordt de DOS blokgrootte zo groot dat ze niet langer bruikbaar zijn (Dit is tot op zekere hoogte gecorrigeerd met FAT32).

ext2 dwingt je niet om grote blokken voor grote bestandssystemen te kiezen, behalve voor zeer grote bestandssystemen in de range 0.5 TB (dat is terabytes waarbij 1 TB gelijk is aan 1024 GB) en daarboven, waar kleine blokgroottes inefficient worden. Dus in tegenstelling tot DOS is het niet nodig om grote disks in meerdere partities onder te verdelen om de blokgrootte laag te houden. Gebruik zo mogelijk de 1 KB standaard blokgrootte. Voor een aantal partities zou het kunnen dat je met een 2 KB blokgrootte wilt experimenteren, maar ga er maar vanuit een aantal zelden in acht genomen bugs tegen te komen: De meeste mensen gebruiken de standaardwaarde.

3.5 Levensduur van bestanden en backupcycli als criteria bij partitioneren

Met ext2, zouden partitionerings beslissingen moeten worden geleid door backup overwegingen en om externe fragmentatie door de verschillende levensduur van bestanden te voorkomen.

Bestanden hebben een levensduur. Nadat een bestand is aangemaakt, zal het een tijd op het systeem blijven bestaan en vervolgens worden verwijderd. De levensduur van een bestand varieert nogal op het systeem en is gedeeltelijk afhankelijk van de padnaam van het bestand. Bijvoorbeeld, bestanden in `/bin`, `/sbin`, `/usr/sbin`, `/usr/bin` en vergelijkbare directory's zullen waarschijnlijk een zeer lange levensduur hebben: vele maanden en meer. Bestanden in `/home` zullen vermoedelijk een gemiddelde levensduur hebben: verscheidene weken of zoiets. Bestanden in `/var` bestaan gewoonlijk maar kort: Bijna geen enkel bestand in `/var/spool/news` zal langer dan een paar dagen behouden blijven, bestanden in `/var/spool/lpd` meten hun levensduur in minuten of minder.

Voor het backuppen is het handig als de hoeveelheid dagelijkse backup kleiner is dan de capaciteit van een enkel backup medium. Een dagelijkse backup kan een complete backup of een incrementele backup zijn.

Je kunt besluiten om je partitiegroottes klein genoeg te houden zodanig dat ze volledig op een backupmedium passen (kies dagelijks volledige backups). In ieder geval zou een partitie klein genoeg moeten zijn dat z'n dagelijkse delta (alle gewijzigde bestanden) op één backupmedium passen. (kies voor incrementele backup en ga er maar vanuit de backup media voor de wekelijkse/maandelijkse volledige dump te moeten wisselen - geen onbeheerde bewerking mogelijk).

Je backupstrategie hangt af van die beslissing.

Als je diskruimte koopt, denk er dan aan voldoende geld voor backup op zij te leggen. Niet gebackupte gegevens zijn waardeloos! De kosten van reproduceren van gegevens zijn voor eigenlijk iedereen veel hoger dan de backupkosten!

Voor performance is het handig om bestanden met verschillende levensduur op verschillende partities te houden. Op deze manier kan het zijn dat de kort bestaande bestanden op de nieuws-partitie heel erg kunnen zijn gefragmenteerd. Dit heeft geen impact op de performance van de `/` of `/home` partitie.

4 Een voorbeeld

4.1 Een aanbevolen model voor ambitieuze beginners

Een algemeen model maakt `/`, `/home` en `/var` partities aan zoals hierboven beschreven. Dit is eenvoudig te installeren en beheren en maakt goed genoeg onderscheid om ongunstige effecten van verschillende levensduur te voorkomen. Het past ook goed in een backupmodel: Bijna niemand bekommert zich om een backup te maken van een USENET newspool en slechts een aantal bestanden in `/var` zijn het waard om te backuppen (ik denk nu aan `/var/spool/mail`). Aan de andere kant, `/` wijzigt niet vaak en hier kan naar behoefte een backup van worden gemaakt (na configuratie wijzingen) en is klein genoeg om op de meeste moderne backupmedia als een volledige backup te passen (reken op 250 tot 500 MB afhankelijk van de hoeveelheid geïnstalleerde software). `/home` bevat waardevolle gebruikersgegevens en hier zou dagelijks een backup van moeten worden gemaakt. Een aantal installaties hebben zeer grote `/homes` en moeten incrementele backups gebruiken.

Een aantal systemen plaatsen `/tmp` ook op een aparte partitie, anderen maken een symlink aan naar `/var/tmp` om hetzelfde effect te bereiken (merk op dat dit effect kan hebben op single user mode, waarin `/var` niet beschikbaar zal zijn en het systeem geen `/tmp` zal hebben, totdat je een `/var` handmatig mount of aanmaakt) of zet het op een RAM disk (Solaris doet dit bijvoorbeeld). Dit houdt `/tmp` buiten `/`, een goed idee.

Dit model is geschikt voor upgrades als ook herinstallatie: Bewaar je configuratiebestanden (of de gehele `/etc`) in een `/home` directory, zet je `/` aan de kant, herinstalleer en haal de oude configuraties vanuit de opgeslagen directory in `/home` op.

5 Hoe ik het deed op mijn machine

Daar stond deze oude ISA bus 386/40 op mijn boekenplank die ik twee jaar geleden had weggedaan omdat hij niet langer voldeed. Ik was van plan het om te zetten in een kleine X-loze server voor mijn Lan thuis.

Zo deed ik het: Ik pakte die 386 en plaatste er 16 MB RAM in. Voegde een goedkope EIDE disk toe, de kleinste die ik kon krijgen (800 MB) en een ethernetkaart. Voegde een oude Hercules toe omdat ik er nog steeds een monitor voor had. Installeerde Linux erop en daar had ik mijn lokale NFS, SMB, HTTP, LPD/LPR en NNTP server als ook mijn mailrouter en POP3 server. Met een extra ISDN kaart werd de machine mijn TCP/IP router en ook firewall.

De meeste diskruimte op deze machine ging naar de `/var` directory's, `/var/spool/mail`, `/var/spool/news` en `/var/httpd/html`. Ik plaatste `/var` op een aparte partitie en maakte deze wat groter. Er zullen bijna geen gebruikers op deze machine zijn, dus ik maakte geen home-partitie aan en mountte `/home` vanuit een ander werkstation via NFS.

Linux zonder X plus verscheidene lokaal geïnstalleerde utility's zal goed gaan met een 250 MB partitie als `/`. De machine heeft 16 MB RAM, maar het zal veel servers gaan draaien. 16 MB swap zou toereikend moeten zijn, 32 MB zou ruim voldoende moeten zijn. We hebben geen tekort aan diskruimte, dus de machine zal 32 MB krijgen. Uit sentimentaliteit blijft er een MS-DOS partitie van pakweg 20 MB op behouden. Ik besloot `/home` vanaf een andere machine te importeren, dus de resterende 500+ MB zal `/var` worden. Dit is meer dan genoeg voor een USENET nieuwstoevoer voor thuisgebruik.

We krijgen

```
Device     Gemount op                Grootte
/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
```

Ik maak een backup van deze machine via het netwerk met gebruik van de tape in `homeserver`. Aangezien alles op deze machine vanaf CD-ROM is geïnstalleerd hoef ik alleen een aantal configuratie bestanden van `/etc`, mijn aangepaste lokaal geïnstalleerde *.tgz bestanden van `/root/Source/Installed` en `/var/spool/mail` als ook `/var/httpd/html` te bewaren. Ik kopieer deze bestanden iedere nacht naar een toegewezen directory `/home/backmeup` op `homeserver`, vanwaar de reguliere homeserver-backup ze ophaalt.