Mini_HOWTO: Multi Disk System Tuning −−>

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

HOWTO: Multi Disk System Tuning

Stein Gjoen, `sgjoen@nyx.net`

v0.33a, 20 May 2002

This document describes how best to use multiple disks and partitions for a Linux system. Although some of this text is Linux specific the general approach outlined here can be applied to many other multi tasking operating systems.

---

1. Introduction

2. Structure

3. Drive Technologies

4. File System Structure

11. Disk Layout

12. Implementation

13. Maintenance

14. Advanced Issues

15. Troubleshooting

16. Further Information

---

1. Introduction

For unclear reasons this brand new release is codenamed the **Taylor3** release.

New code names will appear as per industry standard guidelines to emphasize the state−of−the−art−ness of this document.

This document was written for two reasons, mainly because I got hold of 3 old SCSI disks to set up my Linux system on and I was pondering how best to utilise the inherent possibilities of parallelizing in a SCSI system. Secondly I hear there is a prize for people who write documents...

This is intended to be read in conjunction with the Linux Filesystem Structure Standard (FSSTND). It does not in any way replace it but tries to suggest where physically to place directories detailed in the FSSTND, in terms of drives, partitions, types, RAID, file system (fs), physical sizes and other parameters that should be considered and tuned in a Linux system, ranging from single home systems to large servers on the Internet.

The followup to FSSTND is called the Filesystem Hierarchy Standard (FHS) and covers more than Linux alone. FHS versions 2.0, 2.1 and 2.2 have been released but there are still a few issues to be dealt with. Many recent distributions are now aiming for FHS compliance.

It is also a good idea to read the Linux Installation guides thoroughly and if you are using a PC system, which I guess the majority still does, you can find much relevant and useful information in the FAQs for the newsgroup comp.sys.ibm.pc.hardware especially for storage media.

This is also a learning experience for myself and I hope I can start the ball rolling with this HOWTO and that it perhaps can evolve into a larger more detailed and hopefully even more correct HOWTO.

First of all we need a bit of legalese. Recent development shows it is quite important.

1.1 Copyright

If you have any questions, please contact <{linux−howto@metalab.unc.edu}>

1.2 Disclaimer

Use the information in this document at your own risk. I disavow any potential liability for the contents of this document. Use of the concepts, examples, and/or other content of this document is entirely at your own risk.

All copyrights are owned by their owners, unless specifically noted otherwise. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark.

Naming of particular products or brands should not be seen as endorsements.

You are strongly recommended to take a backup of your system before major installation and backups at regular intervals.

1.3 News

This is a major upgrade featuring a new copyright statement that is intended to be Debian compliant and allow for inclusion in their distribution. A number of mistakes are corrected and new features added such as descriptions of recent ATA features and more.

On the development front people are concentrating their energy towards completing Linux 2.4 and until that is released there is not going to be much news on disk technology for Linux.

Also now the document is available in postscript both for US letter as well as European A4 formats.

The latest version number of this document can be gleaned from my plan entry if you finger my Nyx account.

Also, the latest version will be available on my web space on Nyx in a number of formats:

- HTML.
- plain ASCII text (ca. 6200 lines).
- compressed postscript US letter format (ca. 90 pages).
- compressed postscript European A4 format (ca. 85 pages).
- SGML source (ca. 260 KB).

A European mirror of the Multi Disk HOWTO just went on line.

1.4 Credits

In this version I have the pleasure of acknowledging even more people who have contributed in one way or another:

```
ronnej (at ) ucs.orst.edu
cm (at) kukuruz.ping.at
armbru (at) pond.sub.org
```

R.P.Blake (at) open.ac.uk
neuffer (at) goofy.zdv.Uni-Mainz.de
sjmudd (at) redestb.es
nat (at) nataa.fr.eu.org
sundbyk (at) oslo.geco-prakla.slb.com
ggjoeen (at) online.no
mike (at) i-Connect.Net
roth (at) uiuc.edu
phall (at) ilap.com
szaka (at) mirror.cc.u-szeged.hu
CMckeon (at) swcp.com
kris (at) koentopp.de
edick (at) idcomm.com
pot (at) fly.cnuce.cnr.it
earl (at) sbox.tu-graz.ac.at
ebacon (at) oanet.com
vax (at) linkdead.paranoia.com
tschenk (at) theoffice.net
pjfarley (at) dorsai.org
jean (at) stat.ubc.ca
johnf (at) whitsunday.net.au
clasen (at) unidui.uni-duisburg.de
eeslgw (at) ee.surrey.asc.uk
adam (at) onshore.com
anikolae (at) wega-fddi2.rz.uni-ulm.de
cjaeger (at) dwave.net
eperezte (at) c2i.net
yesteven (at) ms2.hinet.net
cj (at) samurajdata.se
tbotond (at) netx.hu
russel (at) coker.com.au
lars (at) iar.se
GALLAGS3 (at) labs.wyeth.com
morimoto (at) xantia.citroen.org
shulegaa (at) gatekeeper.txl.com
roman.legat (at) stud.uni-hannover.de
ahamish (at) hicks.alien.usr.com
hduff2 (at) worldnet.att.net
mbaehr (at) email.archlab.tuwien.ac.at
adc (at) postoffice.utas.edu.au
pjm (at) bofh.asn.au
jochen.berg (at) ac.com
jpotts (at) us.ibm.com
jarry (at) gmx.net
LeBlanc (at) mcc.ac.uk
masy (at) webmasters.gr.jp
karlheg (at) hegbloom.net
goeran (at) uddeborg.pp.se
wgm (at) telus.net

1.5 Translations

Special thanks go to `nakano (at) apm.seikei.ac.jp` for doing the Japanese translation, general contributions as well as contributing an example of a computer in an academic setting, which is included at the end of this document.

There are now many new translations available and special thanks go to the translators for the job and the input they have given:

- German Translation by `chewie (at) nuernberg.netsurf.de`
- Swedish Translation by `jonah (at) swipnet.se`
- French Translation by `Patrick.Loiseleur (at) lri.fr`
- Chinese Translation by `yesteven (at ) ms2.hinet.net`
- Italian Translation by `bigpaul (at) flashnet.it`

ICP Vortex is gratefully acknowledges for sending in−depth information on their range of RAID controllers.

Also DPT is acknowledged for sending me documentation on their controllers as well as permission to quote from the material. These quotes have been approved before appearing here and will be clearly labelled. No quotes as of yet but that is coming.

Not many still, so please read through this document, make a contribution and join the elite. If I have forgotten anyone, please let me know.

New in this version is an appendix with a few tables you can fill in for your system in order to simplify the design process.

Any comments or suggestions can be mailed to my mail address on Nyx: sgjoen@nyx.net.

So let's cut to the chase where `swap` and `/tmp` are racing along hard drive...

---

2. Structure

As this type of document is supposed to be as much for learning as a technical reference document I have rearranged the structure to this end. For the designer of a system it is more useful to have the information presented in terms of the goals of this exercise than from the point of view of the logical layer structure of the devices themselves. Nevertheless this document would not be complete without such a layer structure the computer field is so full of, so I will include it here as an introduction to how it works.

It is a long time since the *mini* in mini−HOWTO could be defended as proper but I am convinced that this document is as long as it needs to be in order to make the right design decisions, and not longer.

2.1 Logical structure

This is based on how each layer access each other, traditionally with the application on top and the physical layer on the bottom. It is quite useful to show the interrelationship between each of the layers used in controlling drives.

```
___________________________________________________________
|__     File structure          ( /usr /tmp etc)        __|
|__     File system             (ext2fs, vfat etc)      __|
|__     Volume management       (AFS)                   __|
|__     RAID, concatenation     (md)                    __|
|__     Device driver           (SCSI, IDE etc)         __|
|__     Controller              (chip, card)            __|
|__     Connection              (cable, network)        __|
|__     Drive                   (magnetic, optical etc) __|
-----------------------------------------------------------
```

In the above diagram both volume management and RAID and concatenation are optional layers. The 3 lower layers are in hardware. All parts are discussed at length later on in this document.

2.2 Document structure

Most users start out with a given set of hardware and some plans on what they wish to achieve and how big the system should be. This is the point of view I will adopt in this document in presenting the material, starting out with hardware, continuing with design constraints before detailing the design strategy that I have found to work well. I have used this both for my own personal computer at home, a multi purpose server at work and found it worked quite well. In addition my Japanese co−worker in this project have applied the same strategy on a server in an academic setting with similar success.

Finally at the end I have detailed some configuration tables for use in your own design. If you have any comments regarding this or notes from your own design work I would like to hear from you so this document can be upgraded.

2.3 Reading plan

Although not the biggest HOWTO it is nevertheless rather big already and I have been requested to make a reading plan to make it possible to cut down on the volume

Expert

(aka the elite). If you are familiar with Linux as well as disk drive technologies you will find most of what you need in the appendices. Additionally you are recommended to read the FAQ and the Bits'n'pieces chapter.

Experienced

(aka Competent). If you are familiar with computers in general you can go straight to the chapters on technologies and continue from there on.

Newbie

(mostly harmless). You just have to read the whole thing. Sorry. In addition you are also recommended to read all the other disk related HOWTOs.

---

3. Drive Technologies

A far more complete discussion on drive technologies for IBM PCs can be found at the home page of The Enhanced IDE/Fast−ATA FAQ which is also regularly posted on Usenet News. There is also a site dedicated to ATA and ATAPI Information and Software.

Here I will just present what is needed to get an understanding of the technology and get you started on your setup.

3.1 Drives

This is the physical device where your data lives and although the operating system makes the various types seem rather similar they can in actual fact be very different. An understanding of how it works can be very useful in your design work. Floppy drives fall outside the scope of this document, though should there be a big demand I could perhaps be persuaded to add a little here.

3.2 Geometry

Physically disk drives consists of one or more platters containing data that is read in and out using sensors mounted on movable heads that are fixed with respects to themselves. Data transfers therefore happens across all surfaces simultaneously which defines a cylinder of tracks. The drive is also divided into sectors containing a number of data fields.

Drives are therefore often specified in terms of its geometry: the number of Cylinders, Heads and Sectors (CHS).

For various reasons there is now a number of translations between

- the physical CHS of the drive itself
- the logical CHS the drive reports to the BIOS or OS
- the logical CHS used by the OS

Basically it is a mess and a source of much confusion. For more information you are strongly recommended to read the *Large Disk mini−HOWTO*

3.3 Media

The media technology determines important parameters such as read/write rates, seek times, storage size as well as if it is read/write or read only.

Magnetic Drives

This is the typical read−write mass storage medium, and as everything else in the computer world, comes in many flavours with different properties. Usually this is the fastest technology and offers read/write capability. The platter rotates with a constant angular velocity (CAV) with a variable physical sector density for more efficient magnetic media area utilisation. In other words, the number of bits per unit length is kept roughly constant by increasing the number of logical sectors for the outer tracks.

Typical values for rotational speeds are 4500 and 5400 RPM, though 7200 is also used. Very recently also 10000 RPM has entered the mass market. Seek times are around 10 ms, transfer rates quite variable from one type to another but typically 4−40 MB/s. With the extreme high performance drives you should remember that performance costs more electric power which is dissipated as heat, see the point on Power and Heating.

Note that there are several kinds of transfers going on here, and that these are quoted in different units. First of all there is the platter−to−drive cache transfer, usually quoted in Mbits/s. Typical values here is about 50−250 Mbits/s. The second stage is from the built in drive cache to the adapter, and this is typically quoted in MB/s, and typical quoted values here is 3−40 MB/s. Note, however, that this assumed data is already in the cache and hence for maximum readout speed from the drive the effective transfer rate will decrease dramatically.

Optical Drives

Optical read/write drives exist but are slow and not so common. They were used in the NeXT machine but the low speed was a source for much of the complaints. The low speed is mainly due to the thermal nature of the phase change that represents the data storage. Even when using relatively powerful lasers to induce the phase changes the effects are still slower than the magnetic effect used in magnetic drives.

Today many people use CD−ROM drives which, as the name suggests, is read−only. Storage is about 650 MB, transfer speeds are variable, depending on the drive but can exceed 1.5 MB/s. Data is stored on a spiraling single track so it is not useful to talk about geometry for this. Data density is constant so the drive uses constant linear velocity (CLV). Seek is also slower, about 100 ms, partially due to the spiraling track. Recent, high speed drives, use a mix of CLV and CAV in order to maximize performance. This also reduces access time caused by the need to reach correct rotational speed for readout.

A new type (DVD) is on the horizon, offering up to about 18 GB on a single disk.

Solid State Drives

This is a relatively recent addition to the available technology and has been made popular especially in portable computers as well as in embedded systems. Containing no movable parts they are very fast both in terms of access and transfer rates. The most popular type is flash RAM, but also other types of RAM is used. A few years ago many had great hopes for magnetic bubble memories but it turned out to be relatively expensive and is not that common.

In general the use of RAM disks are regarded as a bad idea as it is normally more sensible to add more RAM to the motherboard and let the operating system divide the memory pool into buffers, cache, program and data areas. Only in very special cases, such as real time systems with short time margins, can RAM disks be a sensible solution.

Flash RAM is today available in several 10's of megabytes in storage and one might be tempted to use it for fast, temporary storage in a computer. There is however a huge snag with this: flash RAM has a finite life time in terms of the number of times you can rewrite data, so putting `swap`, `/tmp` or `/var/tmp` on such a device will certainly shorten its lifetime dramatically. Instead, using flash RAM for directories that are read often but rarely written to, will be a big performance win.

In order to get the optimum life time out of flash RAM you will need to use special drivers that will use the RAM evenly and minimize the number of block erases.

This example illustrates the advantages of splitting up your directory structure over several devices.

Solid state drives have no real cylinder/head/sector addressing but for compatibility reasons this is simulated by the driver to give a uniform interface to the operating system.

3.4 Interfaces

There is a plethora of interfaces to chose from widely ranging in price and performance. Most motherboards today include IDE interface which are part of modern chipsets.

Many motherboards also include a SCSI interface chip made by Symbios (formerly NCR) and that is connected directly to the PCI bus. Check what you have and what BIOS support you have with it.

MFM and RLL

Once upon a time this was the established technology, a time when 20 MB was awesome, which compared to todays sizes makes you think that dinosaurs roamed the Earth with these drives. Like the dinosaurs these are outdated and are slow and unreliable compared to what we have today. Linux does support this but you are well advised to think twice about what you would put on this. One might argue that an emergency partition with a suitable vintage of DOS might be fitting.

ESDI

Actually, ESDI was an adaptation of the very widely used SMD interface used on "big" computers to the cable set used with the ST506 interface, which was more convenient to package than the 60−pin + 26−pin connector pair used with SMD. The ST506 was a "dumb" interface which relied entirely on the controller and host computer to do everything from computing head/cylinder/sector locations and keeping track of the head location, etc. ST506 required the controller to extract clock from the recovered data, and control the physical location of detailed track features on the medium, bit by bit. It had about a 10−year life if you include the use of MFM, RLL, and ERLL/ARLL modulation schemes. ESDI, on the other hand, had intelligence, often using three or four separate microprocessors on a single drive, and high−level commands to format a track, transfer data, perform seeks, and so on. Clock recovery from the data stream was accomplished at the drive, which drove the clock line and presented its data in NRZ, though error correction was still the task of the controller. ESDI allowed the use of variable bit density recording, or, for that matter, any other modulation technique, since it was locally generated and resolved at the drive. Though many of the techniques used in ESDI were later incorporated in IDE, it was the increased popularity of SCSI which led to the demise of ESDI in computers. ESDI had a life of about 10 years, though mostly in servers and otherwise "big" systems rather than PC's.

IDE and ATA

Progress made the drive electronics migrate from the ISA slot card over to the drive itself and Integrated Drive Electronics was borne. It was simple, cheap and reasonably fast so the BIOS designers provided the kind of snag that the computer industry is so full of. A combination of an IDE limitation of 16 heads together with the BIOS limitation of 1024 cylinders gave us the infamous 504 MB limit. Following the computer industry traditions again, the snag was patched with a kludge and we got all sorts of translation schemes and BIOS bodges. This means that you need to read the installation documentation very carefully and check up on what BIOS you have and what date it has as the BIOS has to tell Linux what size drive you have. Fortunately with Linux you can also tell the kernel directly what size drive you have with the drive parameters, check the documentation for LILO and Loadlin, thoroughly. Note also that IDE is equivalent to ATA, AT Attachment. IDE uses CPU−intensive Programmed Input/Output (PIO) to transfer data to and from the drives and has no capability for the more efficient Direct Memory Access (DMA) technology. Highest transfer rate is 8.3 MB/s.

EIDE, Fast−ATA and ATA−2

These 3 terms are roughly equivalent, fast−ATA is ATA−2 but EIDE additionally includes ATAPI. ATA−2 is what most use these days which is faster and with DMA. Highest transfer rate is increased to 16.6 MB/s.

Ultra−ATA

A new, faster DMA mode that is approximately twice the speed of EIDE PIO−Mode 4 (33 MB/s). Disks with and without Ultra−ATA can be mixed on the same cable without speed penalty for the faster adapters. The Ultra−ATA interface is electrically identical with the normal Fast−ATA interface, including the maximum cable length.

The ATA/66 was superceeded by ATA/100 and very recently we have now gotten ATA/133. While the interface speed has iproved dramatically the disks are often limited by platter−to−cache limites which today stands at about 40 MB/s.

For more information read up on these overviews and whitepapers from Maxtor: Fast Drives Technology on the ATA/133 interface and Big Drives Technology on breaking the 137 GB limit.

Serial−ATA

A new, standard has been agreed upon, the `Serial−ATA` interface, backed by the The Serial ATA group who made the announcement in August 2001.

Advantages are numerous: simple, thin connectors rather than old cumbersome cable mats that also obstructued air flow, higher speeds (about 150 MB/s) and backward compatibility.

ATAPI

The ATA Packet Interface was designed to support CD−ROM drives using the IDE port and like IDE it is cheap and simple.

SCSI

The Small Computer System Interface is a multi purpose interface that can be used to connect to everything from drives, disk arrays, printers, scanners and more. The name is a bit of a misnomer as it has traditionally been used by the higher end of the market as well as in work stations since it is well suited for multi tasking environments.

The standard interface is 8 bits wide and can address 8 devices. There is a wide version with 16 bit that is twice as fast on the same clock and can address 16 devices. The host adapter always counts as a device and is usually number 7. It is also possible to have 32 bit wide busses but this usually requires a double set of cables to carry all the lines.

The old standard was 5 MB/s and the newer fast−SCSI increased this to 10 MB/s. Recently ultra−SCSI, also known as Fast−20, arrived with 20 MB/s transfer rates for an 8 bit wide bus. New low voltage differential (LVD) signalling allows these high speeds as well as much longer cabling than before.

Even more recently an even faster standard has been introduced: SCSI 160 (originally named SCSI 160/m) which is capable of a monstrous 160 MB/s over a 16 bit wide bus. Support is scarce yet but for a few 10000 RPM drives that can transfer 40 MB/s sustained. Putting 6 such drives on a RAID will keep such a bus saturated and also saturate most PCI busses. Obviously this is only for the very highest end servers per today. More information on this standard is available at The Ultra 160 SCSI home page

Adaptec just announced a Linux driver for their SCSI 160 host adapter. More information will come when more information becomes available.

Now also SCSI/320 is available.

The higher performance comes at a cost that is usually higher than for (E)IDE. The importance of correct termination and good quality cables cannot be overemphasized. SCSI drives also often tend to be of a higher quality than IDE drives. Also adding SCSI devices tend to be easier than adding more IDE drives: Often it is only a matter of plugging or unplugging the device; some people do this without powering down the system. This feature is most convenient when you have multiple systems and you can just take the devices from one system to the other should one of them fail for some reason.

There is a number of useful documents you should read if you use SCSI, the SCSI HOWTO as well as the SCSI FAQ posted on Usenet News.

SCSI also has the advantage you can connect it easily to tape drives for backing up your data, as well as some printers and scanners. It is even possible to use it as a very fast network between computers while simultaneously share SCSI devices on the same bus. Work is under way but due to problems with ensuring cache coherency between the different computers connected, this is a non trivial task.

SCSI numbers are also used for arbitration. If several drives request service, the drive with the lowest number is given priority.

Note that newer SCSI cards will simultaneously support an array of different types of SCSI devices all at individually optimized speeds.

3.5 Cabling

I do not intend to make too many comments on hardware but I feel I should make a little note on cabling. This might seem like a remarkably low technological piece of equipment, yet sadly it is the source of many frustrating problems. At todays high speeds one should think of the cable more of a an RF device with its inherent demands on impedance matching. If you do not take your precautions you will get a much reduced reliability or total failure. Some SCSI host adapters are more sensitive to this than others.

Shielded cables are of course better than unshielded but the price is much higher. With a little care you can get good performance from a cheap unshielded cable.

- For Fast−ATA and Ultra−ATA, the maximum cable length is specified as 45cm (18"). The data lines of both IDE channels are connected on many boards, though, so they count as **one** cable. In any case EIDE cables should be as short as possible. If there are mysterious crashes or spontaneous changes of data, it is well worth investigating your cabling. Try a lower PIO mode or disconnect the second channel and see if the problem still occurs.
- For `Cable Select` (ATA drives) you set the drive jumpers to cable select and use the cable to determine master and slave. This is not much used.
- Do not have a slave on an ATA controller (primary or secondary) without a master on the same controller, behaviour in these cases is undetermined.
- Use as short cable as possible, but do not forget the 30 cm minimum separation for ultra SCSI and 60 cm separation for differential SCSI.
- Avoid long stubs between the cable and the drive, connect the plug on the cable directly to the drive without an extension.

- SCSI Cabling limitations:

```
Bus Speed (MHz)         |    Max Length (m)
−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
 5                      |       6
10  (fast)              |       3
20  (fast−20 / ultra)   |       3 (max 4 devices), 1.5 (max 8 devices)
xx  (differential)      |      25 (max 16 devices
−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
```

- Use correct termination for SCSI devices and at the correct positions: both ends of the SCSI chain. Remember the host adapter itself may have on board termination.
- Do not mix shielded or unshielded cabling, do not wrap cables around metal, try to avoid proximity to metal parts along parts of the cabling. Any such discontinuities can cause impedance mismatching which in turn can cause reflection of signals which increases noise on the cable. This problems gets even more severe in the case of multi channel controllers. Recently someone suggested wrapping bubble plastic around the cables in order to avoid too close proximity to metal, a real problem inside crowded cabinets.

More information on SCSI cabling and termination can be found at various web pages around the net.

3.6 Host Adapters

This is the other end of the interface from the drive, the part that is connected to a computer bus. The speed of the computer bus and that of the drives should be roughly similar, otherwise you have a bottleneck in your system. Connecting a RAID 0 disk−farm to a ISA card is pointless. These days most computers come with 32 bit PCI bus capable of 132 MB/s transfers which should not represent a bottleneck for most people in the near future.

As the drive electronic migrated to the drives the remaining part that became the (E)IDE interface is so small it can easily fit into the PCI chip set. The SCSI host adapter is more complex and often includes a small CPU of its own and is therefore more expensive and not integrated into the PCI chip sets available today. Technological evolution might change this.

Some host adapters come with separate caching and intelligence but as this is basically second guessing the operating system the gains are heavily dependent on which operating system is used. Some of the more primitive ones, that shall remain nameless, experience great gains. Linux, on the other hand, have so much smarts of its own that the gains are much smaller.

Mike Neuffer, who did the drivers for the DPT controllers, states that the DPT controllers are intelligent enough that given enough cache memory it will give you a big push in performance and suggests that people who have experienced little gains with smart controllers just have not used a sufficiently intelligent caching controller.

3.7 Multi Channel Systems

In order to increase throughput it is necessary to identify the most significant bottlenecks and then eliminate them. In some systems, in particular where there are a great number of drives connected, it is advantageous to use several controllers working in parallel, both for SCSI host adapters as well as IDE controllers which usually have 2 channels built in. Linux supports this.

Some RAID controllers feature 2 or 3 channels and it pays to spread the disk load across all channels. In other words, if you have two SCSI drives you want to RAID and a two channel controller, you should put each drive on separate channels.

3.8 Multi Board Systems

In addition to having both a SCSI and an IDE in the same machine it is also possible to have more than one SCSI controller. Check the SCSI−HOWTO on what controllers you can combine. Also you will most likely have to tell the kernel it should probe for more than just a single SCSI or a single IDE controller. This is done using kernel parameters when booting, for instance using LILO. Check the HOWTOs for SCSI and LILO for how to do this.

Multi board systems can offer significant speed gains if you configure your disks right, especially for RAID0. Make sure you interleave the controllers as well as the drives, so that you add drives to the md RAID device in the right order. If controller 1 is connected to drives `sda` and `sdc` while controller 2 is connected to drives `sdb` and `sdd` you will gain more paralellicity by adding in the order of `sda − sdc − sdb − sdd` rather than `sda − sdb − sdc − sdd` because a read or write over more than one cluster will be more likely to span two controllers.

The same methods can also be applied to IDE. Most motherboards come with typically 4 IDE ports:

- `hda` primary master
- `hdb` primary slave
- `hdc` secondary master
- `hdd` secondary slave

where the two primaries share one flat cable and the secondaries share another cable. Modern chipsets keep these independent. Therefore it is best to RAID in the order `hda − hdc − hdb − hdd` as this will most likely parallelise both channels.

3.9 Speed Comparison

The following tables are given just to indicate what speeds are possible but remember that these are the theoretical maximum speeds. All transfer rates are in MB per second and bus widths are measured in bits.

Controllers

```
IDE             :         8.3 − 16.7
Ultra−ATA       :       33 − 66

SCSI            :
                        Bus width (bits)

Bus Speed (MHz)         |        8       16       32
−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
 5                      |        5       10       20
10  (fast)              |       10       20       40
20  (fast−20 / ultra)   |       20       40       80
40  (fast−40 / ultra−2) |       40       80       −−
−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
```

Bus Types

```
ISA              :        8−12
EISA             :      33
VESA             :      40    (Sometimes tuned to 50)

PCI
                        Bus width (bits)

Bus Speed (MHz)        |       32       64
−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
33                     |       132      264
66                     |       264      528
−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−
```

3.10 Benchmarking

This is a very, very difficult topic and I will only make a few cautious comments about this minefield. First of all, it is more difficult to make comparable benchmarks that have any actual meaning. This, however, does not stop people from trying...

Instead one can use benchmarking to diagnose your own system, to check it is going as fast as it should, that is, not slowing down. Also you would expect a significant increase when switching from a simple file system to RAID, so a lack of performance gain will tell you something is wrong.

When you try to benchmark you should not hack up your own, instead look up `iozone` and `bonnie` and read the documentation very carefully. In particular make sure your buffer size is bigger than your RAM size, otherwise you test your RAM rather than your disks which will give you unrealistically high performance.

A very simple benchmark can be obtained using `hdparm −tT` which can be used both on IDE and SCSI drives.

For more information on benchmarking and software for a number of platforms, check out ACNC benchmark page as well as this one and also The Benchmarking−HOWTO.

There are also official home pages for bonnie, bonnie++ and iozone.

Trivia: Bonnie is intended to locate bottlenecks, the name is a tribute to Bonnie Raitt, "who knows how to use one" as the author puts it.

3.11 Comparisons

SCSI offers more performance than EIDE but at a price. Termination is more complex but expansion not too difficult. Having more than 4 (or in some cases 2) IDE drives can be complicated, with wide SCSI you can have up to 15 per adapter. Some SCSI host adapters have several channels thereby multiplying the number of possible drives even further.

For SCSI you have to dedicate one IRQ per host adapter which can control up to 15 drives. With EIDE you need one IRQ for each channel (which can connect up to 2 disks, master and slave) which can cause conflict.

RLL and MFM is in general too old, slow and unreliable to be of much use.

3.12 Future Development

SCSI−3 is under way and will hopefully be released soon. Faster devices are already being announced, recently an 80 MB/s and then a 160 MB/s monster specification has been proposed and also very recently became commercially available. These are based around the Ultra−2 standard (which used a 40 MHz clock) combined with a 16 bit cable.

Some manufacturers already announce SCSI−3 devices but this is currently rather premature as the standard is not yet firm. As the transfer speeds increase the saturation point of the PCI bus is getting closer. Currently the 64 bit version has a limit of 264 MB/s. The PCI transfer rate will in the future be increased from the current 33 MHz to 66 MHz, thereby increasing the limit to 528 MB/s.

The ATA development is continuing and is increasing the performance with the new ATA/100 standard. Since most ATA drives are slower in sustained transfer from platter than this the performance increase will for most people be small.

More interesting is the Serial ATA development, where the flat cable will be replaced with a high speed serial link. This makes cabling far simpler than today and also it solves the problem of cabling obstructing airflow over the drives.

Another trend is for larger and larger drives. I hear it is possible to get 75 GB on a single drive though this is rather expensive. Currently the optimum storage for your money is about 30 GB but also this is continuously increasing. The introduction of DVD will in the near future have a big impact, with nearly 20 GB on a single disk you can have a complete copy of even major FTP sites from around the world. The only thing we can be reasonably sure about the future is that even if it won't get any better, it will definitely be bigger.

Addendum: soon after I first wrote this I read that the maximum useful speed for a CD−ROM was 20x as mechanical stability would be too great a problem at these speeds. About one month after that again the first commercial 24x CD−ROMs were available... Currently you can get 40x and no doubt higher speeds are in the pipeline.

A project to encapsulate SCSI over TCP/IP, called iSCSI has started, and one Linux iSCSI implementation has appeared.

3.13 Recommendations

My personal view is that EIDE or Ultra ATA is the best way to start out on your system, especially if you intend to use DOS as well on your machine. If you plan to expand your system over many years or use it as a server I would strongly recommend you get SCSI drives. Currently wide SCSI is a little more expensive. You are generally more likely to get more for your money with standard width SCSI. There is also differential versions of the SCSI bus which increases maximum length of the cable. The price increase is even more substantial and cannot therefore be recommended for normal users.

In addition to disk drives you can also connect some types of scanners and printers and even networks to a SCSI bus.

Also keep in mind that as you expand your system you will draw ever more power, so make sure your power supply is rated for the job and that you have sufficient cooling. Many SCSI drives offer the option of sequential spin−up which is a good idea for large systems. See also Power and Heating.

---

4. File System Structure

Linux has been multi tasking from the very beginning where a number of programs interact and run continuously. It is therefore important to keep a file structure that everyone can agree on so that the system finds data where it expects to. Historically there has been so many different standards that it was confusing and compatibility was maintained using symbolic links which confused the issue even further and the structure ended looking like a maze.

In the case of Linux a standard was fortunately agreed on early on called the *File Systems Standard* (FSSTND) which today is used by all main Linux distributions.

Later it was decided to make a successor that should also support operating systems other than just Linux, called the *Filesystem Hierarchy Standard* (FHS) at version 2.2 currently. This standard is under continuous development and will soon be adopted by Linux distributions.

I recommend not trying to roll your own structure as a lot of thought has gone into the standards and many software packages comply with the standards. Instead you can read more about this at the FHS home page.

This HOWTO endeavours to comply with FSSTND and will follow FHS when distributions become available.

4.1 File System Features

The various parts of FSSTND have different requirements regarding speed, reliability and size, for instance losing root is a pain but can easily be recovered. Losing `/var/spool/mail` is a rather different issue. Here is a quick summary of some essential parts and their properties and requirements. Note that this is just a guide, there can be binaries in `etc` and `lib` directories, libraries in `bin` directories and so on.

Swap

Speed

Maximum! Though if you rely too much on swap you should consider buying some more RAM. Note, however, that on many old Pentium PC motherboards the cache will not work on RAM above 128 MB.

Size

Similar as for RAM. Quick and dirty algorithm: just as for tea: 16 MB for the machine and 2 MB for each user. Smallest kernel run in 1 MB but is tight, use 4 MB for general work and light applications, 8 MB for X11 or GCC or 16 MB to be comfortable. (The author is known to brew a rather powerful cuppa tea...)

Some suggest that swap space should be 1−2 times the size of the RAM, pointing out that the locality of the programs determines how effective your added swap space is. Note that using the same algorithm as for 4BSD is slightly incorrect as Linux does not allocate space for pages in core.

A more thorough approach is to consider swap space plus RAM as your total working set, so if you know how much space you will need at most, you subtract the physical RAM you have and that is the swap space you will need.

There is also another reason to be generous when dimensioning your swap space: memory leaks. Ill behaving programs that do not free the memory they allocate for themselves are said to have a memory leak. This allocation remains even after the offending program has stopped so this is a source of memory consumption. Only after the program dies is the memory returned. Once all physical RAM and swap space are exhausted the only solution is to kill the offending processes if possible, or failing that, reboot and start over. Thankfully such programs are not too common but should you come across one you will find that extra swap space will buy you extra time between reboots.

Also remember to take into account the type of programs you use. Some programs that have large working sets, such as image processing software have huge data structures loaded in RAM rather than working explicitly on disk files. Data and computing intensive programs like this will cause excessive swapping if you have less RAM than the requirements.

Other types of programs can lock their pages into RAM. This can be for security reasons, preventing copies of data reaching a swap device or for performance reasons such as in a real time module. Either way, locking pages reduces the remaining amount of swappable memory and can cause the system to swap earlier then otherwise expected.

In `man 8 mkswap` it is explained that each swap partition can be a maximum of just under 128 MB in size for 32−bit machines and just under 256 MB for 64−bit machines.

This however changed with kernel 2.2.0 after which the limit is 2 GB. The man page has been updated to reflect this change.

Reliability

Medium. When it fails you know it pretty quickly and failure will cost you some lost work. You save often, don't you?

Note 1

Linux offers the possibility of interleaved swapping across multiple devices, a feature that can gain you much. Check out "`man 8 swapon`" for more details. However, software raiding `swap` across multiple devices adds more overheads than you gain.

Thus the `/etc/fstab` file might look like this:

```
/dev/sda1       swap            swap    pri=1           0       0
/dev/sdc1       swap            swap    pri=1           0       0
```

Remember that the `fstab` file is *very* sensitive to the formatting used, read the man page carefully and do *not* just cut and paste the lines above.

Note 2

Some people use a RAM disk for swapping or some other file systems. However, unless you have some very unusual requirements or setups you are unlikely to gain much from this as this cuts into the memory available for caching and buffering.

Note 2b

There is once exception: on a number of badly designed motherboards the on board cache memory is not able to cache all the RAM that can be addressed. Many older motherboards could accept 128 MB RAM but only cache the lower 64 MB. In such cases it would improve the performance if you used the upper (uncached) 64 MB RAM for RAMdisk based swap or other temporary storage.

Temporary Storage (`/tmp` and `/var/tmp`)

Speed

Very high. On a separate disk/partition this will reduce fragmentation generally, though `ext2fs` handles fragmentation rather well.

Size

Hard to tell, small systems are easy to run with just a few MB but these are notorious hiding places for stashing files away from prying eyes and quota enforcement and can grow without control on larger machines. Suggested: small home machine: 8 MB, large home machine: 32 MB, small server: 128 MB, and large machines up to 500 MB (The machine used by the author at work has 1100 users and a 300 MB `/tmp` directory). Keep an eye on these directories, not only for hidden files but also for old files. Also be prepared that these partitions might be the first reason you might have to resize your partitions.

Reliability

Low. Often programs will warn or fail gracefully when these areas fail or are filled up. Random file errors will of course be more serious, no matter what file area this is.

Files

Mostly short files but there can be a huge number of them. Normally programs delete their old `tmp` files but if somehow an interruption occurs they could survive. Many distributions have a policy regarding cleaning out `tmp` files at boot time, you might want to check out what your setup is.

Note1

In FSSTND there is a note about putting `/tmp` on RAM disk. This, however, is not recommended for the same reasons as stated for swap. Also, as noted earlier, do not use flash RAM drives for these directories. One should also keep in mind that some systems are set to automatically clean `tmp` areas on rebooting.

Note2

Older systems had a `/usr/tmp` but this is no longer recommended and for historical reasons a symbolic link now makes it point to one of the other `tmp` areas.

(* That was 50 lines, I am home and dry! *)

Spool Areas (`/var/spool/news` and `/var/spool/mail`)

Speed

High, especially on large news servers. News transfer and expiring are disk intensive and will benefit from fast drives. Print spools: low. Consider RAID0 for news.

Size

For news/mail servers: whatever you can afford. For single user systems a few MB will be sufficient if you read continuously. Joining a list server and taking a holiday is, on the other hand, not a good idea. (Again the machine I use at work has 100 MB reserved for the entire `/var/spool`)

Reliability

Mail: very high, news: medium, print spool: low. If your mail is very important (isn't it always?) consider RAID for reliability.

Files

Usually a huge number of files that are around a few KB in size. Files in the print spool can on the other hand be few but quite sizable.

Note

Some of the news documentation suggests putting all the `.overview` files on a drive separate from the news files, check out all news FAQs for more information. Typical size is about 3−10 percent of total news spool size.

Home Directories (/home)

Speed

Medium. Although many programs use `/tmp` for temporary storage, others such as some news readers frequently update files in the home directory which can be noticeable on large multiuser systems. For small systems this is not a critical issue.

Size

Tricky! On some systems people pay for storage so this is usually then a question of finance. Large systems such as Nyx.net (which is a free Internet service with mail, news and WWW services) run successfully with a suggested limit of 100 KB per user and 300 KB as enforced maximum. Commercial ISPs offer typically about 5 MB in their standard subscription packages.

If however you are writing books or are doing design work the requirements balloon quickly.

Reliability

Variable. Losing `/home` on a single user machine is annoying but when 2000 users call you to tell you their home directories are gone it is more than just annoying. For some their livelihood relies on what is here. You do regular backups of course?

Files

Equally tricky. The minimum setup for a single user tends to be a dozen files, 0.5 − 5 KB in size. Project related files can be huge though.

Note1

You might consider RAID for either speed or reliability. If you want extremely high speed and reliability you might be looking at other operating system and hardware platforms anyway. (Fault tolerance etc.)

Note2

Web browsers often use a local cache to speed up browsing and this cache can take up a substantial amount of space and cause much disk activity. There are many ways of avoiding this kind of performance hits, for more information see the sections on Home Directories and WWW.

Note3

Users often tend to use up all available space on the `/home` partition. The Linux Quota subsystem is capable of limiting the number of blocks and the number of inode a single user ID can allocate on a per−filesystem basis. See the Linux Quota mini−HOWTO by Albert M.C. Tam `bertie (at) scn.org` for details on setup.

Main Binaries (`/usr/bin` and `/usr/local/bin`)

Speed

Low. Often data is bigger than the programs which are demand loaded anyway so this is not speed critical. Witness the successes of live file systems on CD ROM.

Size

The sky is the limit but 200 MB should give you most of what you want for a comprehensive system. A big system, for software development or a multi purpose server should perhaps reserve 500 MB both for installation and for growth.

Reliability

Low. This is usually mounted under root where all the essentials are collected. Nevertheless losing all the binaries is a pain...

Files

Variable but usually of the order of 10 − 100 KB.

Libraries (`/usr/lib` and `/usr/local/lib`)

Speed

Medium. These are large chunks of data loaded often, ranging from object files to fonts, all susceptible to bloating. Often these are also loaded in their entirety and speed is of some use here.

Size

Variable. This is for instance where word processors store their immense font files. The few that have given me feedback on this report about 70 MB in their various `lib` directories. A rather complete Debian 1.2 installation can take as much as 250 MB which can be taken as an realistic upper limit. The following ones are some of the largest disk space consumers: GCC, Emacs, TeX/LaTeX, X11 and perl.

Reliability

Low. See point Main binaries.

Files

Usually large with many of the order of 1 MB in size.

Note

For historical reasons some programs keep executables in the lib areas. One example is GCC which have some huge binaries in the `/usr/lib/gcc/lib` hierarchy.

Boot

Speed

Quite low: after all booting doesn't happen that often and loading the kernel is just a tiny fraction of the time it takes to get the system up and running.

Size

Quite small, a complete image with some extras fit on a single floppy so 5 MB should be plenty.

Reliability

High. See section below on Root.

Note 1

The most important part about the Boot partition is that on many systems it *must* reside below cylinder 1023. This is a BIOS limitation that Linux cannot get around.

Note 1a

The above is not necessarily true for recent IDE systems and not for any SCSI disks. For more information check the latest Large Disk HOWTO.

Note 2

Recently a new boot loader has been written that overcomes the 1023 sector limit. For more information check out this article on nuni.

Root

Speed

Quite low: only the bare minimum is here, much of which is only run at startup time.

Size

Relatively small. However it is a good idea to keep some essential rescue files and utilities on the root partition and some keep several kernel versions. Feedback suggests about 20 MB would be sufficient.

Reliability

High. A failure here will possibly cause a fair bit of grief and you might end up spending some time rescuing your boot partition. With some practice you can of course do this in an hour or so, but I would think if you have some practice doing this you are also doing something wrong.

Naturally you do have a rescue disk? Of course this is updated since you did your initial installation? There are many ready made rescue disks as well as rescue disk creation tools you might find valuable. Presumably investing some time in this saves you from becoming a root rescue expert.

Note 1

If you have plenty of drives you might consider putting a spare emergency boot partition on a separate physical drive. It will cost you a little bit of space but if your setup is huge the time saved, should something fail, will be well worth the extra space.

Note 2

For simplicity and also in case of emergencies it is not advisable to put the root partition on a RAID level 0 system. Also if you use RAID for your boot partition you have to remember to have the `md` option turned on for your emergency kernel.

Note 3

For simplicity it is quite common to keep Boot and Root on the same partition. If you do that, then in order to boot from LILO it is important that the essential boot files reside wholly within cylinder 1023. This includes the kernel as well as files found in `/boot`.

DOS etc.

At the danger of sounding heretical I have included this little section about something many reading this document have strong feelings about. Unfortunately many hardware items come with setup and maintenance tools based around those systems, so here goes.

Speed

Very low. The systems in question are not famed for speed so there is little point in using prime quality drives. Multitasking or multi−threading are not available so the command queueing facility found in SCSI drives will not be taken advantage of. If you have an old IDE drive it should be good enough. The exception is to some degree Win95 and more notably NT which have multi−threading support which should theoretically be able to take advantage of the more advanced features offered by SCSI devices.

Size

The company behind these operating systems is not famed for writing tight code so you have to be prepared to spend a few tens of MB depending on what version you install of the OS or Windows. With an old version of DOS or Windows you might fit it all in on 50 MB.

Reliability

Ha−ha. As the chain is no stronger than the weakest link you can use any old drive. Since the OS is more likely to scramble itself than the drive is likely to self destruct you will soon learn the importance of keeping backups here.

Put another way: "*Your mission, should you choose to accept it, is to keep this partition working. The warranty will self destruct in 10 seconds...*"

Recently I was asked to justify my claims here. First of all I am not calling DOS and Windows sorry excuses for operating systems. Secondly there are various legal issues to be taken into account. Saying there is a connection between the last two sentences are merely the ravings of the paranoid. Surely. Instead I shall offer the esteemed reader a few key words: DOS 4.0, DOS 6.x and various drive compression tools that shall remain nameless.

4.2 Explanation of Terms

Naturally the faster the better but often the happy installer of Linux has several disks of varying speed and reliability so even though this document describes performance as 'fast' and 'slow' it is just a rough guide since no finer granularity is feasible. Even so there are a few details that should be kept in mind:

Speed

This is really a rather woolly mix of several terms: CPU load, transfer setup overhead, disk seek time and transfer rate. It is in the very nature of tuning that there is no fixed optimum, and in most cases price is the dictating factor. CPU load is only significant for IDE systems where the CPU does the transfer itself but is generally low for SCSI, see SCSI documentation for actual numbers. Disk seek time is also small, usually in the millisecond range. This however is not a problem if you use command queueing on SCSI where you then overlap commands keeping the bus busy all the time. News spools are a special case consisting of a huge number of normally small files so in this case seek time can become more significant.

There are two main parameters that are of interest here:

Seek

> is usually specified in the average time take for the read/write head to seek from one track to another. This parameter is important when dealing with a large number of small files such as found in spool files. There is also the extra seek delay before the desired sector rotates into position under the head. This delay is dependent on the angular velocity of the drive which is why this parameter quite often is quoted for a drive. Common values are 4500, 5400 and 7200 RPM (rotations per minute). Higher RPM reduces the seek time but at a substantial cost. Also drives working at 7200 RPM have been known to be noisy and to generate a lot of heat, a factor that should be kept in mind if you are building a large array or "disk farm". Very recently drives working at 10000 RPM has entered the market and here the cooling requirements are even stricter and minimum figures for air flow are given.

Transfer

> is usually specified in megabytes per second. This parameter is important when handling large files that have to be transferred. Library files, dictionaries and image files are examples of this. Drives featuring a high rotation speed also normally have fast transfers as transfer speed is proportional to angular velocity for the same sector density.

It is therefore important to read the specifications for the drives very carefully, and note that the maximum transfer speed quite often is quoted for transfers out of the on board cache (burst speed) and *not* directly from the platter (sustained speed). See also section on Power and Heating.

Reliability

Naturally no−one would want low reliability disks but one might be better off regarding old disks as unreliable. Also for RAID purposes (See the relevant information) it is suggested to use a mixed set of disks so that simultaneous disk crashes become less likely.

So far I have had only one report of total file system failure but here unstable hardware seemed to be the cause of the problems.

Disks are cheap these days yet people still underestimate the value of the contents of the drives. If you need higher reliability make sure you replace old drives and keep spares. It is not unusual that drives can work more or less continuous for years and years but what often kills a drive in the end is power cycling.

Files

The average file size is important in order to decide the most suitable drive parameters. A large number of small files makes the average seek time important whereas for big files the transfer speed is more important. The command queueing in SCSI devices is very handy for handling large numbers of small files, but for transfer EIDE is not too far behind SCSI and normally much cheaper than SCSI.

5. File Systems

Over time the requirements for file systems have increased and the demands for large structures, large files, long file names and more has prompted ever more advanced file systems, the system that accesses and organises the data on mass storage. Today there is a large number of file systems to choose from and this section will describe these in detail.

The emphasis is on Linux but with more input I will be happy to add information for a wider audience.

5.1 General Purpose File Systems

Most operating systems usually have a general purpose file system for every day use for most kinds of files, reflecting available features in the OS such as permission flags, protection and recovery.

`minix`

This was the original fs for Linux, back in the days Linux was hosted on minix machines. It is simple but limited in features and hardly ever used these days other than in some rescue disks as it is rather compact.

`xiafs` and `extfs`

These are also old and have fallen in disuse and are no longer recommended.

`ext2fs`

This is the established standard for general purpose in the Linux world. It is fast, efficient and mature and is under continuous development and features such as ACL and transparent compression are on the horizon.

For more information check the ext2fs home page.

`ext3fs`

This is the name for the upcoming successor to `ext2fs` due to enter stable kernel in the near future. Many features are added to `ext2fs` but to avoid confusion over the name after such a radical upgrade the name will be changed too. You may have heard of it already but source code is now in beta release .

Patches are available at Linux.org.

`ufs`

This is the fs used by BSD and variants thereof. It is mature but also developed for older types of disk drives where geometries were known. The fs uses a number of tricks to optimise performance but as disk geometries are translated in a number of ways the net effect is no longer so optimal.

`efs`

The Extent File System (efs) is Silicon Graphics' early file system widely used on IRIX before version 6.0 after which xfs has taken over. While migration to xfs is encouraged efs is still supported and much used on

CDs.

There is a Linux driver available in early beta stage, available at Linux extent file system home page.

`XFS`

Silicon Graphics Inc (sgi) has started porting its mainframe grade file system to Linux. Source is not yet available as they are busily cleaning out legal encumbrance but once that is done they will provide the source code under GPL.

More information is already available on the XFS project page at SGI.

`reiserfs`

As of July, 23th 1997 Hans Reiser `reiser (at) RICOCHET.NET` has put up the source to his tree based reiserfs on the web. While his filesystem has some very interesting features and is much faster than `ext2fs` and is in use by a number of people. Hopefully it will be ready for kernel 2.4.0 which might be ready at the end of the year.

`enh-fs`

The Enhanced File System project is now dead.

`Tux2 fs`

This is a variation on the `ext2fs` that adds robustness in case of unexpected interruptions such as power failure. After such an event `Tux2 fs` will restart with the file system in a consistent, recently recorded state without fsck or other recovery operations. To achieve this `Tux2 fs` uses a newly designed algorithm called Phase Tree.

More information can be found at the project home page.

5.2 Microsoft File Systems

This company is responsible for a lot, including a number of filesystems that has at the very least caused confusions.

`fat`

Actually there are 2 `fat`s out there, `fat12` and `fat16` depending on the partition size used but fortunately the difference is so minor that the whole issue is transparent.

On the plus side these are fast and simple and most OSes understands it and can both read and write this fs. And that is about it.

The minus side is limited safety, severely limited permission flags and atrocious scalability. For instance with `fat` you cannot have partitions larger than 2 GB.

`fat32`

After about 10 years Microsoft realised `fat` was about, well, 10 years behind the times and created this fs which scales reasonably well.

Permission flags are still limited. NT 4.0 cannot read this file system but Linux can.

`vfat`

At the same time as Microsoft launched `fat32` they also added support for long file names, known as `vfat`.

Linux reads `vfat` and `fat32` partitions by mounting with type `vfat`.

`ntfs`

This is the native fs of Win−NT but as complete information is not available there is limited support for other OSes.

5.3 Logging and Journaling File Systems

These take a radically different approach to file updates by logging modifications for files in a log and later at some time checkpointing the logs.

Reading is roughly as fast as traditional file systems that always update the files directly. Writing is much faster as only updates are appended to a log. All this is transparent to the user. It is in reliability and particularly in checking file system integrity that these file systems really shine. Since the data before last checkpointing is known to be good only the log has to be checked, and this is much faster than for traditional file systems.

Note that while *logging* filesystems keep track of changes made to both data and inodes, *journaling* filesystems keep track only of inode changes.

Linux has quite a choice in such file systems but none are yet in production quality. Some are also on hold.

- Adam Richter from Yggdrasil posted some time ago that they have been working on a compressed log file based system but that this project is currently on hold. Nevertheless a non−working version is available on their FTP server. Check out the Yggdrasil ftp server where special patched versions of the kernel can be found.
- Another project is the Linux log−structured Filesystem Project which sadly also is on hold. Nevertheless this page contains much information on the topic.
- Then there is the LinLogFS −− A Log−Structured Filesystem For Linux (formerly known as dtfs) which seems to be going strong. Still in alpha but sufficiently complete to make programs run off this file system
- Finally there is the Journaling Flash File System designed for their embedded diskless systems such as their Linux based web camera.

Note that `ext3fs`, `XFS` and `reiserfs` also have features for logging or journaling.

5.4 Read−only File Systems

Read−only media has not escaped the ever increasing complexities seen in more general file systems so again there is a large choice to choose from with corresponding opportunities for exciting mistakes.

Note that `ext2fs` works quite well on a CD−ROM and seems to save space while offering the normal file system features such as long file names and permissions that can be retained when copying files across to read−write media. Also having /dev on a CD−ROM is possible.

Most of these are used with the CD−ROM media but also the new DVD can be used and you can even use it through the loopback device on a hard disk file for verifying an image before burning a ROM.

There is a read−only `romfs` for Linux but as that is not disk related nothing more will be said about it here.

`High Sierra`

This was one of the earliest standards for CD−ROM formats, supposedly named after the hotel where the final agreement took place.

`High Sierra` was so limited in features that new extensions simply had to appear and while there has been no end to new formats the original `High Sierra` remains the common precursor and is therefore still widely supported.

`iso9660`

The International Standards Organisation made their extensions and formalised the standard into what we know as the `iso9660` standard.

The Linux iso9660 file system supports both High Sierra as well as `Rock Ridge` extensions.

`Rock Ridge`

Not everyone accepts limits like short filenames and lack of permissions so very soon the `Rock Ridge` extensions appeared to rectify these shortcomings.

`Joliet`

Microsoft, not be be outdone in the standards extension game, decided it should extend CD−ROM formats with some internationalisation features and called it `Joliet`.

Linux supports this standards in kernels 2.0.34 or newer. You need to enable NLS in order to use it.

Trivia

Joliet is a city outside Chicago; best known for being the site of the prison where Jake was locked up in the movie "Blues Brothers." Rock Ridge (the UNIX extensions to ISO 9660) is named after the (fictional) town in the movie "Blazing Saddles."

UDF

With the arrival of DVD with up to about 17 GB of storage capacity the world seemingly needed another format, this time ambitiously named Universal Disk Format (UDF). This is intended to replace `iso9660` and will be required for DVD.

Currently this is not in the standard Linux kernel but a project is underway to make a http://trylinux.com/projects/udf/index.html name="UDF driver"> for Linux. Patches and documentation are available.

More information is also available at the Linux and DVDs page.

5.5 Networking File Systems

There is a large number of networking technologies available that lets you distribute disks throughout a local or even global networks. This is somewhat peripheral to the topic of this HOWTO but as it can be used with local disks I will cover this briefly. It would be best if someone (else) took this into a separate HOWTO...

NFS

This is one of the earliest systems that allows mounting a file space on one machine onto another. There are a number of problems with NFS ranging from performance to security but it has nevertheless become established.

AFS

This is a system that allows efficient sharing of files across large networks. Starting out as an academic project it is now sold by Transarc whose home page gives you more details.

Derek Atkins, of MIT, ported AFS to Linux and has also set up the Linux AFS mailing List (linux−afs@mit.edu) for this which is open to the public. Requests to join the list should go to linux−afs−request@mit.edu and finally bug reports should be directed to linux−afs−bugs@mit.edu.

Important: as AFS uses encryption it is restricted software and cannot easily be exported from the US.

IBM who owns Transarc, has announced the availability of the latest version of client as well as server for Linux.

Arla is a free AFS implementation, check the Arla homepage for more information as well as documentation.

Coda

A networking filesystem similar to AFS is underway and is called Coda. This is designed to be more robust and fault tolerant than AFS, and supports mobile, disconnected operations. Currently it does not scale very well, and does not really have proper administrative tools, as AFS does and ARLA is beginning to.

`nbd`

The Network Block Device (`nbd`) is available in Linux kernel 2.2 and later and offers reportedly excellent performance. The interesting thing here is that it can be combined with RAID (see later).

`enbd`

The http://www.it.uc3m.es/~ptb/nbd name="Enhanced Network Block Device"> (`enbd`) is a project to enhance the `nbd` with features such as block journaled multi channel communications, internal failover and automatic balancing between channels and more.

The intended use is for RAID over the net.

GFS

The Global File System is a new file system designed for storage across a wide area network. It is currently in the early stages and more information will come later.

5.6 Special File Systems

In addition to the general file systems there is also a number of more specific ones, usually to provide higher performance or other features, usually with a tradeoff in other respects.

`tmpfs` and `swapfs`

For short term fast file storage SunOS offers `tmpfs` which is about the same as the `swapfs` on NeXT. This overcomes the inherent slowness in `ufs` by caching file data and keeping control information in memory. This means that data on such a file system will be lost when rebooting and is therefore mainly suitable for `/tmp` area but not `/var/tmp` which is where temporary data that must survive a reboot, is placed.

SunOS offers very limited tuning for `tmpfs` and the number of files is even limited by total physical memory of the machine.

Linux now features `tmpfs` since kernel version 2.4 and is enabled by turning on virtual memory file system support (former shm fs). Under certain circumstances `tmpfs` can lock up the system in early kerbel versions, make sure you use version 2.4.6 or later.

`userfs`

The user file system (`userfs`) allows a number of extensions to traditional file system use such as FTP based file system, compression (`arcfs`) and fast prototyping and many other features. The `docfs` is based on this filesystem. Check the userfs homepage for more information.

`devfs`

When disks are added, removed or just fail it is likely that disk device names of the remaining disks will change. For instance if `sdb` fails then the old `sdc` becomes `sdb`, the old `sdc` becomes `sdb` and so on. Note that in this case `hda`, `hdb` etc will remain unchanged. Likewise if a new drive is added the reverse may happen.

There is no guarantee that SCSI ID 0 becomes `sda` and that adding disks in increasing ID order will just add a new device name without renaming previous entries, as some SCSI drivers assign from ID 0 and up while others reverse the scanning order. Likewise adding a SCSI host adapter can also cause renaming.

Generally device names are assigned in the order they are found.

The source of the problem lies in the limited number of bits available for major and minor numbering in the device files used to describe the device itself. You an see these in the /dev directory, info on the numbering and allocation can be found in `man MAKEDEV`. Currently there are 2 solutions to this problem in various stages of development:

scsidev

works by creating a database of drives and where they belong, check *man scsifs* and the scsidev home page for more information

devfs

is a more long term project aimed at getting around the whole business of device numbering by making the /dev directory a kernel file system in the same way as /proc is. More information will appear as it becomes available.

`smugfs`

For a number of reasons it is currently difficult to have files bigger than 2 GB. One file system that tries to overcome this limit is `smugfs` which is very fast but also simple. For instance there are no directories and the block allocation is simple.

It is available as compressed tarred source code and while it worked with kernel version 2.1.85 it is quite possible some work is required to make it fit into newer kernels. Also the low version number (0.0) suggests extra care is required.

5.7 File System Recommendations

There is a jungle of choices but generally it is recommended to use the general file system that comes with your distribution. If you use `ufs` and have some kind of `tmpfs` available you should first start off with the general file system to get an idea of the space requirements and if necessary buy more RAM to support the size of `tmpfs` you need. Otherwise you will end up with mysterious crashes and lost time.

If you use dual boot and need to transfer data between the two OSes one of the simplest ways is to use an appropriately sized partition formatted with `fat` as most systems can reliably read and write this. Remember the limit of 2 GB for `fat` partitions.

For more information of file system interconnectivity you can check out the file system page which has been superseded by file system and the article Kragen's Amazing List of Filesystems.

That guide is being superseded by a HOWTO which is underway and a link will be added when it is ready.

To avoid total havoc with device renaming if a drive fails check out the scanning order of your system and try to keep your root system on `hda` or `sda` and removable media such as ZIP drives at the end of the scanning order.

---

6. Technologies

In order to decide how to get the most of your devices you need to know what technologies are available and their implications. As always there can be some tradeoffs with respect to speed, reliability, power, flexibility, ease of use and complexity.

Many of the techniques described below can be stacked in a number of ways to maximise performance and reliability, though at the cost of added complexity.

6.1 RAID

This is a method of increasing reliability, speed or both by using multiple disks in parallel thereby decreasing access time and increasing transfer speed. A checksum or mirroring system can be used to increase reliability. Large servers can take advantage of such a setup but it might be overkill for a single user system unless you already have a large number of disks available. See other documents and FAQs for more information.

For Linux one can set up a RAID system using either software (the `md` module in the kernel), a Linux compatible controller card (PCI−to−SCSI) or a SCSI−to−SCSI controller. Check the documentation for what controllers can be used. A hardware solution is usually faster, and perhaps also safer, but comes at a significant cost.

A summary of available hardware RAID solutions for Linux is available at Linux Consulting.

SCSI−to−SCSI

SCSI−to−SCSI controllers are usually implemented as complete cabinets with drives and a controller that connects to the computer with a second SCSI bus. This makes the entire cabinet of drives look like a single large, fast SCSI drive and requires no special RAID driver. The disadvantage is that the SCSI bus connecting the cabinet to the computer becomes a bottleneck.

A significant disadvantage for people with large disk farms is that there is a limit to how many SCSI entries there can be in the /dev directory. In these cases using SCSI−to−SCSI will conserve entries.

Usually they are configured via the front panel or with a terminal connected to their on−board serial interface.

Some manufacturers of such systems are CMD and Syred whose web pages describe several systems.

PCI−to−SCSI

PCI−to−SCSI controllers are, as the name suggests, connected to the high speed PCI bus and is therefore not suffering from the same bottleneck as the SCSI−to−SCSI controllers. These controllers require special drivers but you also get the means of controlling the RAID configuration over the network which simplifies management.

Currently only a few families of PCI−to−SCSI host adapters are supported under Linux.

DPT

The oldest and most mature is a range of controllers from DPT including SmartCache I/III/IV and SmartRAID I/III/IV controller families. These controllers are supported by the EATA−DMA driver in the standard kernel. This company also has an informative home page which also describes various general aspects of RAID and SCSI in addition to the product related information.

More information from the author of the DPT controller drivers (EATA* drivers) can be found at his pages on SCSI and DPT.

These are not the fastest but have a good track record of proven reliability.

Note that the maintenance tools for DPT controllers currently run under DOS/Win only so you will need a small DOS/Win partition for some of the software. This also means you have to boot the system into Windows in order to maintain your RAID system.

ICP−Vortex

A very recent addition is a range of controllers from ICP−Vortex featuring up to 5 independent channels and very fast hardware based on the i960 chip. The Linux driver was written by the company itself which shows they support Linux.

As ICP−Vortex supplies the maintenance software for Linux it is not necessary with a reboot to other operating systems for the setup and maintenance of your RAID system. This saves you also extra downtime.

Mylex DAC−960

This is one of the latest entries which is out in early beta. More information as well as drivers are available at Dandelion Digital's Linux DAC960 Page.

Compaq Smart−2 PCI Disk Array Controllers

Another very recent entry and currently in beta release is the Smart−2 driver.

IBM ServeRAID

IBM has released their driver as GPL.

Software RAID

A number of operating systems offer software RAID using ordinary disks and controllers. Cost is low and performance for raw disk IO can be very high. As this can be very CPU intensive it increases the load noticeably so if the machine is CPU bound in performance rather then IO bound you might be better off with a hardware PCI−to−RAID controller.

Real cost, performance and especially reliability of software vs. hardware RAID is a very controversial topic. Reliability on Linux systems have been very good so far.

The current software RAID project on Linux is the `md` system (multiple devices) which offers much more than RAID so it is described in more details later.

RAID Levels

RAID comes in many levels and flavours which I will give a brief overview of this here. Much has been written about it and the interested reader is recommended to read more about this in the Software RAID HOWTO.

- RAID *0* is not redundant at all but offers the best throughput of all levels here. Data is striped across a number of drives so read and write operations take place in parallel across all drives. On the other hand if a single drive fail then everything is lost. Did I mention backups?
- RAID *1* is the most primitive method of obtaining redundancy by duplicating data across all drives. Naturally this is massively wasteful but you get one substantial advantage which is fast access. The drive that access the data first wins. Transfers are not any faster than for a single drive, even though you might get some faster read transfers by using one track reading per drive. Also if you have only 2 drives this is the only method of achieving redundancy.
- RAID *2* and *4* are not so common and are not covered here.
- RAID *3* uses a number of disks (at least 2) to store data in a striped RAID 0 fashion. It also uses an additional redundancy disk to store the XOR sum of the data from the data disks. Should the redundancy disk fail, the system can continue to operate as if nothing happened. Should any single data disk fail the system can compute the data on this disk from the information on the redundancy disk and all remaining disks. Any double fault will bring the whole RAID set off−line. RAID 3 makes sense only with at least 2 data disks (3 disks including the redundancy disk). Theoretically there is no limit for the number of disks in the set, but the probability of a fault increases with the number of disks in the RAID set. Usually the upper limit is 5 to 7 disks in a single RAID set. Since RAID 3 stores all redundancy information on a dedicated disk and since this information has to be updated whenever a write to any data disk occurs, the overall write speed of a RAID 3 set is limited by the write speed of the redundancy disk. This, too, is a limit for the number of disks in a RAID set. The overall read speed of a RAID 3 set with all data disks up and running is that of a RAID 0 set with that number of data disks. If the set has to reconstruct data stored on a failed disk from redundant information, the performance will be severely limited: All disks in the set have to be read and XOR−ed to compute the missing information.
- RAID *5* is just like RAID 3, but the redundancy information is spread on all disks of the RAID set. This improves write performance, because load is distributed more evenly between all available disks.

There are also hybrids available based on RAID 0 or 1 and one other level. Many combinations are possible but I have only seen a few referred to. These are more complex than the above mentioned RAID levels.

RAID *0/1* combines striping with duplication which gives very high transfers combined with fast seeks as well as redundancy. The disadvantage is high disk consumption as well as the above mentioned complexity.

RAID *1/5* combines the speed and redundancy benefits of RAID5 with the fast seek of RAID1. Redundancy is improved compared to RAID 0/1 but disk consumption is still substantial. Implementing such a system would involve typically more than 6 drives, perhaps even several controllers or SCSI channels.

6.2 Volume Management

Volume management is a way of overcoming the constraints of fixed sized partitions and disks while still having a control of where various parts of file space resides. With such a system you can add new disks to your system and add space from this drive to parts of the file space where needed, as well as migrating data out from a disk developing faults to other drives before catastrophic failure occurs.

The system developed by Veritas has become the defacto standard for logical volume management.

Volume management is for the time being an area where Linux is lacking.

One is the virtual partition system project VPS that will reimplement many of the volume management functions found in IBM's AIX system. Unfortunately this project is currently on hold.

Another project is the Logical Volume Manager project that is similar to a project by HP.

6.3 Linux `md` Kernel Patch

The Linux Multi Disk (md) provides a number of block level features in various stages of development.

RAID 0 (striping) and concatenation are very solid and in production quality and also RAID 4 and 5 are quite mature.

It is also possible to stack some levels, for instance mirroring (RAID 1) two pairs of drives, each pair set up as striped disks (RAID 0), which offers the speed of RAID 0 combined with the reliability of RAID 1.

In addition to RAID this system offers (in alpha stage) block level volume management and soon also translucent file space. Since this is done on the block level it can be used in combination with any file system, even for `fat` using Wine.

Think very carefully what drives you combine so you can operate all drives in parallel, which gives you better performance and less wear. Read more about this in the documentation that comes with `md`.

Unfortunately The Linux software RAID has split into two trees, the old stable versions 0.35 and 0.42 which are documented in the official Software−RAID HOWTO and the newer less stable 0.90 series which is documented in the unofficial Software RAID HOWTO which is a work in progress.

A patch for online growth of ext2fs is available in early stages and related work is taking place at the ext2fs resize project at Sourceforge.

Hint: if you cannot get it to work properly you have forgotten to set the `persistent−block` flag. Your best documentation is currently the source code.

6.4 Compression

Disk compression versus file compression is a hotly debated topic especially regarding the added danger of file corruption. Nevertheless there are several options available for the adventurous administrators. These take on many forms, from kernel modules and patches to extra libraries but note that most suffer various forms of limitations such as being read−only. As development takes place at neck breaking speed the specs have undoubtedly changed by the time you read this. As always: check the latest updates yourself. Here only a few references are given.

- DouBle features file compression with some limitations.
- Zlibc adds transparent on−the−fly decompression of files as they load.

- there are many modules available for reading compressed files or partitions that are native to various other operating systems though currently most of these are read−only.
- dmsdos (currently in version 0.9.2.0) offer many of the compression options available for DOS and Windows. It is not yet complete but work is ongoing and new features added regularly.
- `e2compr` is a package that extends `ext2fs` with compression capabilities. It is still under testing and will therefore mainly be of interest for kernel hackers but should soon gain stability for wider use. Check the http://e2compr.memalpha.cx/e2compr/ name="e2compr homepage"> for more information. I have reports of speed and good stability which is why it is mentioned here.

6.5 ACL

Access Control List (ACL) offers finer control over file access on a user by user basis, rather than the traditional owner, group and others, as seen in directory listings (`drwxr−xr−x`). This is currently not available in Linux but is expected in kernel 2.3 as hooks are already in place in `ext2fs`.

6.6 `cachefs`

This uses part of a hard disk to cache slower media such as CD−ROM. It is available under SunOS but not yet for Linux.

6.7 Translucent or Inheriting File Systems

This is a copy−on−write system where writes go to a different system than the original source while making it look like an ordinary file space. Thus the file space inherits the original data and the translucent write back buffer can be private to each user.

There is a number of applications:

- updating a live file system on CD−ROM, making it flexible, fast while also conserving space,
- original skeleton files for each new user, saving space since the original data is kept in a single space and shared out,
- parallel project development prototyping where every user can seemingly modify the system globally while not affecting other users.

SunOS offers this feature and this is under development for Linux. There was an old project called the Inheriting File Systems (`ifs`) but this project has stopped. One current project is part of the `md` system and offers block level translucence so it can be applied to any file system.

Sun has an informative page on translucent file system.

It should be noted that Clearcase (now owned by Rational) pioneered and popularized translucent filesystems for software configuration management by writing their own UNIX filesystem.

6.8 Physical Track Positioning

This trick used to be very important when drives were slow and small, and some file systems used to take the varying characteristics into account when placing files. Although higher overall speed, on board drive and controller caches and intelligence has reduced the effect of this.

Nevertheless there is still a little to be gained even today. As we know, "*world dominance*" is soon within reach but to achieve this "*fast*" we need to employ all the tricks we can use .

To understand the strategy we need to recall this near ancient piece of knowledge and the properties of the various track locations. This is based on the fact that transfer speeds generally increase for tracks further away from the spindle, as well as the fact that it is faster to seek to or from the central tracks than to or from the inner or outer tracks.

Most drives use disks running at constant angular velocity but use (fairly) constant data density across all tracks. This means that you will get much higher transfer rates on the outer tracks than on the inner tracks; a characteristics which fits the requirements for large libraries well.

Newer disks use a logical geometry mapping which differs from the actual physical mapping which is transparently mapped by the drive itself. This makes the estimation of the "middle" tracks a little harder.

In most cases track 0 is at the outermost track and this is the general assumption most people use. Still, it should be kept in mind that there are no guarantees this is so.

Inner

tracks are usually slow in transfer, and lying at one end of the seeking position it is also slow to seek to.

This is more suitable to the low end directories such as DOS, root and print spools.

Middle

tracks are on average faster with respect to transfers than inner tracks and being in the middle also on average faster to seek to.

This characteristics is ideal for the most demanding parts such as `swap`, `/tmp` and `/var/tmp`.

Outer

tracks have on average even faster transfer characteristics but like the inner tracks are at the end of the seek so statistically it is equally slow to seek to as the inner tracks.

Large files such as libraries would benefit from a place here.

Hence seek time reduction can be achieved by positioning frequently accessed tracks in the middle so that the average seek distance and therefore the seek time is short. This can be done either by using `fdisk` or `cfdisk` to make a partition on the middle tracks or by first making a file (using `dd`) equal to half the size of the entire disk before creating the files that are frequently accessed, after which the dummy file can be deleted. Both cases assume starting from an empty disk.

The latter trick is suitable for news spools where the empty directory structure can be placed in the middle before putting in the data files. This also helps reducing fragmentation a little.

This little trick can be used both on ordinary drives as well as RAID systems. In the latter case the calculation for centring the tracks will be different, if possible. Consult the latest RAID manual.

The speed difference this makes depends on the drives, but a 50 percent improvement is a typical value.

Disk Speed Values

The same mechanical head disk assembly (HDA) is often available with a number of interfaces (IDE, SCSI etc) and the mechanical parameters are therefore often comparable. The mechanics is today often the limiting factor but development is improving things steadily. There are two main parameters, usually quoted in milliseconds (ms):

- Head movement − the speed at which the read−write head is able to move from one track to the next, called access time. If you do the mathematics and doubly integrate the seek first across all possible starting tracks and then across all possible target tracks you will find that this is equivalent of a stroke across a third of all tracks.
- Rotational speed − which determines the time taken to get to the right sector, called latency.

After voice coils replaced stepper motors for the head movement the improvements seem to have levelled off and more energy is now spent (literally) at improving rotational speed. This has the secondary benefit of also improving transfer rates.

Some typical values:

	Drive type		
Access time (ms)	Fast	Typical	Old
Track−to−track	<1	2	8
Average seek	10	15	30
End−to−end	10	30	70

This shows that the very high end drives offer only marginally better access times then the average drives but that the old drives based on stepper motors are significantly worse.

Rotational speed (RPM)	3600	4500	4800	5400	7200	10000
Latency (ms)	17	13	12.5	11.1	8.3	6.0

As latency is the average time taken to reach a given sector, the formula is quite simply

```
latency (ms) = 60000 / speed (RPM)
```

Clearly this too is an example of diminishing returns for the efforts put into development. However, what really takes off here is the power consumption, heat and noise.

6.9 Yoke

There is also a Linux Yoke Driver available in beta which is intended to do hot−swappable transparent binding of one Linux block device to another. This means that if you bind two block devices together, say `/dev/hda` and `/dev/loop0`, writing to one device will mean also writing to the other and reading from either will yield the same result.

6.10 Stacking

One of the advantages of a layered design of an operating system is that you have the flexibility to put the pieces together in a number of ways. For instance you can cache a CD−ROM with `cachefs` that is a volume striped over 2 drives. This in turn can be set up translucently with a volume that is NFS mounted from another machine. RAID can be stacked in several layers to offer very fast seek and transfer in such a way that it will work if even 3 drives fail. The choices are many, limited only by imagination and, probably more importantly, money.

6.11 Recommendations

There is a near infinite number of combinations available but my recommendation is to start off with a simple setup without any fancy add−ons. Get a feel for what is needed, where the maximum performance is required, if it is access time or transfer speed that is the bottle neck, and so on. Then phase in each component in turn. As you can stack quite freely you should be able to retrofit most components in as time goes by with relatively few difficulties.

RAID is usually a good idea but make sure you have a thorough grasp of the technology and a solid back up system.

---

7. Other Operating Systems

Many Linux users have several operating systems installed, often necessitated by hardware setup systems that run under other operating systems, typically DOS or some flavour of Windows. A small section on how best to deal with this is therefore included here.

7.1 DOS

Leaving aside the debate on weather or not DOS qualifies as an operating system one can in general say that it has little sophistication with respect to disk operations. The more important result of this is that there can be severe difficulties in running various versions of DOS on large drives, and you are therefore strongly recommended in reading the *Large Drives mini−HOWTO*. One effect is that you are often better off placing DOS on low track numbers.

Having been designed for small drives it has a rather unsophisticated file system (`fat`) which when used on large drives will allocate enormous block sizes. It is also prone to block fragmentation which will after a while cause excessive seeks and slow effective transfers.

One solution to this is to use a defragmentation program regularly but it is strongly recommended to back up data and verify the disk before defragmenting. All versions of DOS have `chkdsk` that can do some disk checking, newer versions also have `scandisk` which is somewhat better. There are many defragmentation programs available, some versions have one called `defrag`. Norton Utilities have a large suite of disk tools and there are many others available too.

As always there are snags, and this particular snake in our drive paradise is called *hidden files*. Some vendors started to use these for copy protection schemes and would not take kindly to being moved to a different place on the drive, even if it remained in the same place in the directory structure. The result of this was that newer defragmentation programs will not touch any hidden file, which in turn reduces the effect of defragmentation.

Being a single tasking, single threading and single most other things operating system there is very little gains in using multiple drives unless you use a drive controller with built in RAID support of some kind.

There are a few utilities called `join` and `subst` which can do some multiple drive configuration but there is very little gains for a lot of work. Some of these commands have been removed in newer versions.

In the end there is very little you can do, but not all hope is lost. Many programs need fast, temporary storage, and the better behaved ones will look for environment variables called `TMPDIR` or `TEMPDIR` which you can set to point to another drive. This is often best done in `autoexec.bat`.

```
SET TMPDIR=E:/TMP
SET TEMPDIR=E:/TEMP
```

Not only will this possibly gain you some speed but also it can reduce fragmentation.

There have been reports about difficulties in removing multiple primary partitions using the `fdisk` program that comes with DOS. Should this happen you can instead use a Linux rescue disk with Linux `fdisk` to repair the system.

Don't forget there are other alternatives to DOS, the most well known being DR−DOS from Caldera. This is a direct descendant from DR−DOS from Digital Research. It offers many features not found in the more common DOS, such as multi tasking and long filenames.

Another alternative which also is free is Free DOS which is a project under development. A number of free utilities are also available.

7.2 Windows

Most of the above points are valid for Windows too, with the exception of Windows95 which apparently has better disk handling, which will get better performance out of SCSI drives.

A useful thing is the introduction of long filenames, to read these from Linux you will need the `vfat` file system for mounting these partitions.

Disk fragmentation is still a problem. Some of this can be avoided by doing a defragmentation immediately before and immediately after installing large programs or systems. I use this scheme at work and have found it to work quite well. Purging unused files and emptying the waste basket first can improve defragmentation further.

Windows also use swap drives, redirecting this to another drive can give you some performance gains. There are several mini−HOWTOs telling you how best to share swap space between various operating systems.

The trick of setting `TEMPDIR` can still be used but not all programs will honour this setting. Some do, though. To get a good overview of the settings in the control files you can run `sysedit` which will open a number of files for editing, one of which is the `autoexec` file where you can add the `TEMPDIR` settings.

Much of the temporary files are located in the `/windows/temp` directory and changing this is more tricky. To achieve this you can use `regedit` which is rather powerful and quite capable of rendering your system in a state you will not enjoy, or more precisely, in a state much less enjoyable than windows in general. Registry database error is a message that means seriously bad news. Also you will see that many programs have their own private temporary directories scattered around the system.

Setting the swap file to a separate partition is a better idea and much less risky. Keep in mind that this partition cannot be used for anything else, even if there should appear to be space left there.

It is now possible to read `ext2fs` partitions from Windows, either by mounting the partition using FSDEXT2 or by using a file explorer like tool called Explore2fs.

7.3 OS/2

The only special note here is that you can get file system driver for OS/2 that can read an `ext2fs` partition. Matthieu Willm's ext2fs Installable File System for OS/2 can be found at ftp−os2.nmsu.edu, Sunsite, ftp.leo.org and ftp−os2.cdrom.com.

The IFS has read and write capabilities.

7.4 NT

This is a more serious system featuring most buzzwords known to marketing. It is well worth noting that it features software striping and other more sophisticated setups. Check out the drive manager in the control panel. I do not have easy access to NT, more details on this can take a bit of time.

One important snag was recently reported by acahalan at cs.uml.edu : (reformatted from a Usenet News posting)

NT DiskManager has a serious bug that can corrupt your disk when you have several (more than one?) extended partitions. Microsoft provides an emergency fix program at their web site. See the knowledge base for more. (This affects Linux users, because Linux users have extra partitions)

You can now read `ext2fs` partitions from NT using Explore2fs.

7.5 Windows 2000

Most points regarding Windows NT also applies to its descendant Windows 2000 though at the time of writing this I do not know if the aforementioned bugs have been fixed or not.

While Windows 2000, like its predecessor, features RAID, at least one company, RAID Toolbox, has found the bundled RAID somewhat lacking and made their own commercial alternative.

7.6 Sun OS

There is a little bit of confusion in this area between Sun OS vs. Solaris. Strictly speaking Solaris is just Sun OS 5.x packaged with Openwindows and a few other things. If you run Solaris, just type `uname −a` to see your version. Parts of the reason for this confusion is that Sun Microsystems used to use an OS from the BSD family, albeight with a few bits and pieces from elsewhere as well as things made by themselves. This was the situation up to Sun OS 4.x.y when they did a "strategic roadmap decision" and decided to switch over to the official Unix, System V, Release 4 (aka SVR5), and Sun OS 5 was created. This made a lot of people unhappy. Also this was bundled with other things and marketed under the name Solaris, which currently stands at release 7 which just recently replaced version 2.6 as the latest and greatest. In spite of the large jump in version number this is actually a minor technical upgrade but a giant leap for marketing.

Sun OS 4

This is quite familiar to most Linux users. The last release is 4.1.4 plus various patches. Note however that the file system structure is quite different and does not conform to FSSTND so any planning must be based on the traditional structure. You can get some information by the man page on this: `man hier`. This is, like most man pages, rather brief but should give you a good start. If you are still confused by the structure it will at least be at a higher level.

Sun OS 5 (aka Solaris)

This comes with a snazzy installation system that runs under Openwindows, it will help you in partitioning and formatting the drives before installing the system from CD−ROM. It will also fail if your drive setup is too far out, and as it takes a complete installation run from a full CD−ROM in a 1x only drive this failure will dawn on you after too long time. That is the experience we had where I used to work. Instead we installed everything onto one drive and then moved directories across.

The default settings are sensible for most things, yet there remains a little oddity: swap drives. Even though the official manual recommends multiple swap drives (which are used in a similar fashion as on Linux) the default is to use only a single drive. It is recommended to change this as soon as possible.

Sun OS 5 offers also a file system especially designed for temporary files, `tmpfs`. It offers significant speed improvements over `ufs` but does not survive rebooting.

The only comment so far is: beware! Under Solaris 2.0 it seem that creating too big files in `/tmp` can cause an out of swap space kernel panic trap. As the evidence of what has happened is as lost as any data on a RAMdisk after powering down it can be hard to find out what has happened. What is worse, it seems that user space processes can cause this kernel panic and unless this problem is taken care of it is best not to use `tmpfs` in potentially hostile environments.

Also see the notes on tmpfs.

Trivia: There is a movie also called Solaris, a science fiction movie that is very, very long, slow and incomprehensible. This was often pointed out at the time Solaris (the OS) appeared...

7.7 BeOS

This operating system is one of the more recent one to arrive and it features a file system that has some database like features.

There is a BFS file system driver being developed for Linux and is available in alpha stage. For more information check the Linux BFS page where patches also are available.

8. Clusters

In this section I will briefly touch on the ways machines can be connected together but this is so big a topic it could be a separate HOWTO in its own right, hint, hint. Also, strictly speaking, this section lies outside the scope of this HOWTO, so if you feel like getting fame etc. *you* could contact me and take over this part and turn it into a new document.

These days computers gets outdated at an incredible rate. There is however no reason why old hardware could not be put to good use with Linux. Using an old and otherwise outdated computer as a network server can be both useful in its own right as well as a valuable educational exercise. Such a local networked cluster of computers can take on many forms but to remain within the charter of this HOWTO I will limit myself to the disk strategies. Nevertheless I would hope someone else could take on this topic and turn it into a document on its own.

This is an exciting area of activity today, and many forms of clustering is available today, ranging from automatic workload balancing over local network to more exotic hardware such as Scalable Coherent Interface (SCI) which gives a tight integration of machines, effectively turning them into a single machine. Various kinds of clustering has been available for larger machines for some time and the VAXcluster is perhaps a well known example of this. Clustering is done usually in order to share resources such as disk drives, printers and terminals etc, but also processing resources equally transparently between the computational nodes.

There is no universal definition of clustering, in here it is taken to mean a network of machines that combine their resources to serve users. Admittedly this is a rather loose definition but this will change later.

These days also Linux offers some clustering features but for a starter I will just describe a simple local network. It is a good way of putting old and otherwise unusable hardware to good use, as long as they can run Linux or something similar.

One of the best ways of using an old machine is as a network server in which case the effective speed is more likely to be limited by network bandwidth rather than pure computational performance. For home use you can move the following functionality off to an older machine used as a server:

- news
- mail
- web proxy
- printer server
- modem server (PPP, SLIP, FAX, Voice mail)

You can also `NFS mount` drives from the server onto your workstation thereby reducing drive space requirements. Still read the FSSTND to see what directories should *not* be exported. The best candidates for exporting to all machines are `/usr` and `/var/spool` and possibly `/usr/local` but probably not `/var/spool/lpd`.

Most of the time even slow disks will deliver sufficient performance. On the other hand, if you do processing directly on the disks on the server or have very fast networking, you might want to rethink your strategy and use faster drives. Searching features on a web server or news database searches are two examples of this.

Such a network can be an excellent way of learning system administration and building up your own toaster network, as it often is called. You can get more information on this in other HOWTOs but there are two important things you should keep in mind:

- Do not pull IP numbers out of thin air. Configure your inside net using IP numbers reserved for private use, and use your network server as a router that handles this IP masquerading.
- Remember that if you additionally configure the router as a firewall you might not be able to get to your own data from the outside, depending on the firewall configuration.

The *Nyx* network provides an example of a cluster in the sense defined here. It consists of the following machines:

nyx

is one of the two user login machines and also provides some of the networking services.

nox

(aka nyx10) is the main user login machine and is also the mail server.

noc

is a dedicated news server. The news spool is made accessible through NFS mounting to nyx and nox.

arachne

(aka www) is the web server. Web pages are written by NFS mounting onto nox.

There are also some more advanced clustering projects going, notably

- The Beowulf Project
- The Genoa Active Message Machine (GAMMA)

High−tech clustering requires high−tech interconnect, and SCI is one of them. To find out more you can either look up the home page of Dolphin Interconnect Solutions which is one of the main actors in this field, or you can have a look at scizzl.

Centralised mail servers using IMAP are becoming more and more popular as disks become large enough to keep all mail stored indefinitely and also cheap enough to make it a feasible option. Unfortunately it has become clear that `NFS` mounting the mail archives from another machine can cause corruption of the IMAP database as the server software does not handle NFS timeouts too well, and NFS timeouts are a rather common occurrence. Keep therefore the mail archive local to the IMAP server.

---

9. Mount Points

In designing the disk layout it is important not to split off the directory tree structure at the wrong points, hence this section. As it is highly dependent on the FSSTND it has been put aside in a separate section, and will most likely have to be totally rewritten when FHS is adopted in a Linux distribution. In the meanwhile this will do.

Remember that this is a list of where a separation *can* take place, not where it *has* to be. As always, good judgement is always required.

Again only a rough indication can be given here. The values indicate

```
0=don't separate here
1=not recommended
 ...
4=useful
5=recommended
```

In order to keep the list short, the uninteresting parts are removed.

```
Directory   Suitability
/
|
+-bin       0
+-boot      5
+-dev       0
+-etc       0
+-home      5
+-lib       0
+-mnt       0
+-proc      0
+-root      0
+-sbin      0
+-tmp       5
+-usr       5
| \
| +-X11R6     3
| +-bin       3
| +-lib       4
| +-local     4
| | \
| | +bin        2
| | +lib        4
| +-src       3
|
+-var       5
  \
  +-adm       0
  +-lib       2
  +-lock      1
  +-log       0
  +-preserve  1
  +-run       1
  +-spool     4
  | \
  | +-mail      3
  | +-mqueue    3
  | +-news      5
  | +-smail     3
  | +-uucp      3
  +-tmp       5
```

There is of course plenty of adjustments possible, for instance a home user would not bother with splitting off the `/var/spool` hierarchy but a serious ISP should. The key here is *usage*.

QUIZ! Why should `/etc` never be on a separate partition? Answer: Mounting instructions during boot is found in the file `/etc/fstab` so if this is on a separate and unmounted partition it is like the key to a locked drawer is inside that drawer, a hopeless situation. (Yes, I'll do nearly anything to liven up this HOWTO.)

---

10. Considerations and Dimensioning

The starting point in this will be to consider where you are and what you want to do. The typical home system starts out with existing hardware and the newly converted Linux user will want to get the most out of existing hardware. Someone setting up a new system for a specific purpose (such as an Internet provider) will instead have to consider what the goal is and buy accordingly. Being ambitious I will try to cover the entire range.

Various purposes will also have different requirements regarding file system placement on the drives, a large multiuser machine would probably be best off with the `/home` directory on a separate disk, just to give an example.

In general, for performance it is advantageous to split most things over as many disks as possible but there is a limited number of devices that can live on a SCSI bus and cost is naturally also a factor. Equally important, file system maintenance becomes more complicated as the number of partitions and physical drives increases.

10.1 Home Systems

With the cheap hardware available today it is possible to have quite a big system at home that is still cheap, systems that rival major servers of yesteryear. While many started out with old, discarded disks to build a Linux server (which is how this HOWTO came into existence), many can now afford to buy 40 GB disks up front.

Size remains important for some, and here are a few guidelines:

Testing

> Linux is simple and you don't even need a hard disk to try it out, if you can get the boot floppies to work you are likely to get it to work on your hardware. If the standard kernel does not work for you, do not forget that often there can be special boot disk versions available for unusual hardware combinations that can solve your initial problems until you can compile your own kernel.

Learning

> about operating system is something Linux excels in, there is plenty of documentation and the source is available. A single drive with 50 MB is enough to get you started with a shell, a few of the most frequently used commands and utilities.

Hobby

> use or more serious learning requires more commands and utilities but a single drive is still all it takes, 500 MB should give you plenty of room, also for sources and documentation.

Serious

software development or just serious hobby work requires even more space. At this stage you have probably a mail and news feed that requires spool files and plenty of space. Separate drives for various tasks will begin to show a benefit. At this stage you have probably already gotten hold of a few drives too. Drive requirements gets harder to estimate but I would expect 2−4 GB to be plenty, even for a small server.

Servers

come in many flavours, ranging from mail servers to full sized ISP servers. A base of 2 GB for the main system should be sufficient, then add space and perhaps also drives for separate features you will offer. Cost is the main limiting factor here but be prepared to spend a bit if you wish to justify the "S" in ISP. Admittedly, not all do it.

Basically a server is dimensioned like any machine for serious use with added space for the services offered, and tends to be IO bound rather than CPU bound.

With cheap networking technology both for land lines as well as through radio nets, it is quite likely that very soon home users will have their own servers more or less permanently hooked onto the net.

10.2 Servers

Big tasks require big drives and a separate section here. If possible keep as much as possible on separate drives. Some of the appendices detail the setup of a small departmental server for 10−100 users. Here I will present a few consideration for the higher end servers. In general you should not be afraid of using RAID, not only because it is fast and safe but also because it can make growth a little less painful. All the notes below come as additions to the points mentioned earlier.

Popular servers rarely just happens, rather they grow over time and this demands both generous amounts of disk space as well as a good net connection. In many of these cases it might be a good idea to reserve entire SCSI drives, in singles or as arrays, for each task. This way you can move the data should the computer fail. Note that transferring drives across computers is not simple and might not always work, especially in the case of IDE drives. Drive arrays require careful setup in order to reconstruct the data correctly, so you might want to keep a paper copy of your `fstab` file as well as a note of SCSI IDs.

Home Directories

Estimate how many drives you will need, if this is more than 2 I would recommend RAID, strongly. If not you should separate users across your drives dedicated to users based on some kind of simple hashing algorithm. For instance you could use the first 2 letters in the user name, so `jbloggs` is put on `/u/j/b/jbloggs` where `/u/j` is a symbolic link to a physical drive so you can get a balanced load on your drives.

Anonymous FTP

This is an essential service if you are serious about service. Good servers are well maintained, documented, kept up to date, and immensely popular no matter where in the world they are located. The big server ftp.funet.fi is an excellent example of this.

In general this is not a question of CPU but of network bandwidth. Size is hard to estimate, mainly it is a question of ambition and service attitudes. I believe the big archive at ftp.cdrom.com is a *BSD machine with 50 GB disk. Also memory is important for a dedicated FTP server, about 256 MB RAM would be sufficient for a very big server, whereas smaller servers can get the job done well with 64 MB RAM. Network connections would still be the most important factor.

WWW

For many this is the main reason to get onto the Internet, in fact many now seem to equate the two. In addition to being network intensive there is also a fair bit of drive activity related to this, mainly regarding the caches. Keeping the cache on a separate, fast drive would be beneficial. Even better would be installing a caching proxy server. This way you can reduce the cache size for each user and speed up the service while at the same time cut down on the bandwidth requirements.

With a caching proxy server you need a fast set of drives, RAID0 would be ideal as reliability is not important here. Higher capacity is better but about 2 GB should be sufficient for most. Remember to match the cache period to the capacity and demand. Too long periods would on the other hand be a disadvantage, if possible try to adjust based on the URL. For more information check up on the most used servers such as `Harvest`, Squid and the one from Netscape.

Mail

Handling mail is something most machines do to some extent. The big mail servers, however, come into a class of their own. This is a demanding task and a big server can be slow even when connected to fast drives and a good net feed. In the Linux world the big server at `vger.rutgers.edu` is a well known example. Unlike a news service which is distributed and which can partially reconstruct the spool using other machines as a feed, the mail servers are centralised. This makes safety much more important, so for a major server you should consider a RAID solution with emphasize on reliability. Size is hard to estimate, it all depends on how many lists you run as well as how many subscribers you have.

Big mail servers can be IO limited in performance and for this reason some use huge silicon disks connected to the SCSI bus to hold all mail related files including temporary files. For extra safety these are battery backed and filesystems like `udf` are preferred since they always flush metadata to disk. This added cost to performance is offset by the very fast disk.

Note that these days more and more switch over from using `POP` to pull mail to local machine from mail server and instead use `IMAP` to serve mail while keeping the mail archive centralised. This means that mail is no longer spooled in its original sense but often builds up, requiring huge disk space. Also more and more (ab)use mail attachments to send all sorts of things across, even a small word processor document can easily end up over 1 MB. Size your disks generously and keep an eye on how much space is left.

News

This is definitely a high volume task, and very dependent on what news groups you subscribe to. On Nyx there is a fairly complete feed and the spool files consume about 17 GB. The biggest groups are no doubt in the `alt.binary.*` hierarchy, so if you for some reason decide not to get these you can get a good service with perhaps 12 GB. Still others, that shall remain nameless, feel 2 GB is sufficient to claim ISP status. In this case news expires so fast I feel the spelling IsP is barely justified. A full newsfeed means a traffic of a few GB every day and this is an ever growing number.

Others

There are many services available on the net and even though many have been put somewhat in the shadows by the web. Nevertheless, services like *archie*, *gopher* and *wais* just to name a few, still exist and remain valuable tools on the net. If you are serious about starting a major server you should also consider these services. Determining the required volumes is hard, it all depends on popularity and demand. Providing good service inevitably has its costs, disk space is just one of them.

Server Recommendations

Servers today require large numbers of large disks to function satisfactorily in commercial settings. As mean time between failure (MTBF) decreases rapidly as the number of components increase it is advisable to look into using RAID for protection and use a number of medium sized drives rather than one single huge disk. Also look into the High Availability (HA) project for more information. More information is available at

High Availability HOWTO and also at related web pages.

There is also an article in Byte called How Big Does Your Unix Server Have To Be? with many points that are relevant to Linux.

10.3 Pitfalls

The dangers of splitting up everything into separate partitions are briefly mentioned in the section about volume management. Still, several people have asked me to emphasize this point more strongly: when one partition fills up it cannot grow any further, no matter if there is plenty of space in other partitions.

In particular look out for explosive growth in the news spool (`/var/spool/news`). For multi user machines with quotas keep an eye on `/tmp` and `/var/tmp` as some people try to hide their files there, just look out for filenames ending in gif or jpeg...

In fact, for single physical drives this scheme offers very little gains at all, other than making file growth monitoring easier (using '`df`') and physical track positioning. Most importantly there is no scope for parallel disk access. A freely available volume management system would solve this but this is still some time in the future. However, when more specialised file systems become available even a single disk could benefit from being divided into several partitions.

For more information see section Troubleshooting.

---

11. Disk Layout

With all this in mind we are now ready to embark on the layout. I have based this on my own method developed when I got hold of 3 old SCSI disks and boggled over the possibilities.

The tables in the appendices are designed to simplify the mapping process. They have been designed to help you go through the process of optimizations as well as making an useful log in case of system repair. A few examples are also given.

11.1 Selection for Partitioning

Determine your needs and set up a list of all the parts of the file system you want to be on separate partitions and sort them in descending order of speed requirement and how much space you want to give each partition.

The table in Appendix A section is a useful tool to select what directories you should put on different partitions. It is sorted in a logical order with space for your own additions and notes about mounting points and additional systems. It is therefore NOT sorted in order of speed, instead the speed requirements are indicated by bullets ('o').

If you plan to RAID make a note of the disks you want to use and what partitions you want to RAID. Remember various RAID solutions offers different speeds and degrees of reliability.

(Just to make it simple I'll assume we have a set of identical SCSI disks and no RAID)

11.2 Mapping Partitions to Drives

Then we want to place the partitions onto physical disks. The point of the following algorithm is to maximise parallelizing and bus capacity. In this example the drives are A, B and C and the partitions are 987654321 where 9 is the partition with the highest speed requirement. Starting at one drive we 'meander' the partition line over and over the drives in this way:

```
        A : 9 4 3
        B : 8 5 2
        C : 7 6 1
```

This makes the 'sum of speed requirements' the most equal across each drive.

Use the table in Appendix B section to select what drives to use for each partition in order to optimize for paralellicity.

Note the speed characteristics of your drives and note each directory under the appropriate column. Be prepared to shuffle directories, partitions and drives around a few times before you are satisfied.

11.3 Sorting Partitions on Drives

After that it is recommended to select partition numbering for each drive.

Use the table in Appendix C section to select partition numbers in order to optimize for track characteristics. At the end of this you should have a table sorted in ascending partition number. Fill these numbers back into the tables in appendix A and B.

You will find these tables useful when running the partitioning program (`fdisk` or `cfdisk`) and when doing the installation.

11.4 Optimizing

After this there are usually a few partitions that have to be 'shuffled' over the drives either to make them fit or if there are special considerations regarding speed, reliability, special file systems etc. Nevertheless this gives what this author believes is a good starting point for the complete setup of the drives and the partitions. In the end it is actual use that will determine the real needs after we have made so many assumptions. After commencing operations one should assume a time comes when a repartitioning will be beneficial.

For instance if one of the 3 drives in the above mentioned example is very slow compared to the two others a better plan would be as follows:

```
A : 9 6 5
B : 8 7 4
C : 3 2 1
```

Optimizing by Characteristics

Often drives can be similar in apparent overall speed but some advantage can be gained by matching drives to the file size distribution and frequency of access. Thus binaries are suited to drives with fast access that offer command queueing, and libraries are better suited to drives with larger transfer speeds where IDE offers good performance for the money.

Optimizing by Drive Parallelising

Avoid drive contention by looking at tasks: for instance if you are accessing `/usr/local/bin` chances are you will soon also need files from `/usr/local/lib` so placing these at separate drives allows less seeking and possible parallel operation and drive caching. It is quite possible that choosing what may appear less than ideal drive characteristics will still be advantageous if you can gain parallel operations. Identify common tasks, what partitions they use and try to keep these on separate physical drives.

Just to illustrate my point I will give a few examples of task analysis here.

Office software

such as editing, word processing and spreadsheets are typical examples of low intensity software both in terms of CPU and disk intensity. However, should you have a single server for a huge number of users you should not forget that most such software have auto save facilities which cause extra traffic, usually on the home directories. Splitting users over several drives would reduce contention.

News

readers also feature auto save features on home directories so ISPs should consider separating home directories

News spools are notorious for their deeply nested directories and their large number of very small files. Loss of a news spool partition is not a big problem for most people, too, so they are good candidates for a RAID 0 setup with many small disks to distribute the many seeks among multiple spindles. It is recommended in the manuals and FAQs for the INN news server to put news spool and `.overview` files on separate drives for larger installations.

Some notes on INN optimising under Tru64 UNIX also applies to a wider audience, including Linux users.

Database

applications can be demanding both in terms of drive usage and speed requirements. The details are naturally application specific, read the documentation carefully with disk requirements in mind. Also consider RAID both for performance and reliability.

E−mail

reading and sending involves home directories as well as in− and outgoing spool files. If possible keep home directories and spool files on separate drives. If you are a mail server or a mail hub consider putting in− and outgoing spool directories on separate drives.

Losing mail is an extremely bad thing, if you are managing an ISP or major hub. Think about RAIDing your mail spool and consider frequent backups.

Software development

can require a large number of directories for binaries, libraries, include files as well as source and project files. If possible split as much as possible across separate drives. On small systems you can place `/usr/src` and project files on the same drive as the home directories.

Web browsing

is becoming more and more popular. Many browsers have a local cache which can expand to rather large volumes. As this is used when reloading pages or returning to the previous page, speed is quite important here. If however you are connected via a well configured proxy server you do not need more than typically a few megabytes per user for a session. See also the sections on Home Directories and WWW.

11.5 Compromises

One way to avoid the aforementioned pitfalls is to only set off fixed partitions to directories with a fairly well known size such as swap, `/tmp` and `/var/tmp` and group together the remainders into the remaining partitions using symbolic links.

Example: a slow disk (`slowdisk`), a fast disk (`fastdisk`) and an assortment of files. Having set up `swap` and `tmp` on `fastdisk`; and `/home` and root on slowdisk we have (the fictitious) directories `/a/slow`, `/a/fast`, `/b/slow` and `/b/fast` left to allocate on the partitions `/mnt.slowdisk` and `/mnt.fastdisk` which represents the remaining partitions of the two drives.

Putting `/a` or `/b` directly on either drive gives the same properties to the subdirectories. We could make all 4 directories separate partitions but would lose some flexibility in managing the size of each directory. A better solution is to make these 4 directories symbolic links to appropriate directories on the respective drives.

Thus we make

```
/a/fast point to /mnt.fastdisk/a/fast   or   /mnt.fastdisk/a.fast
/a/slow point to /mnt.slowdisk/a/slow   or   /mnt.slowdisk/a.slow
/b/fast point to /mnt.fastdisk/b/fast   or   /mnt.fastdisk/b.fast
/b/slow point to /mnt.slowdisk/b/slow   or   /mnt.slowdisk/b.slow
```

and we get all fast directories on the fast drive without having to set up a partition for all 4 directories. The second (right hand) alternative gives us a flatter files system which in this case can make it simpler to keep an overview of the structure.

The disadvantage is that it is a complicated scheme to set up and plan in the first place and that all mount points and partitions have to be defined before the system installation.

Important: note that the `/usr` partition must be mounted directly onto root and not via an indirect link as described above. The reason for this are the long backward links used extensively in X11 that go from deep within `/usr` all the way to root and then down into `/etc` directories.

---

12. Implementation

Having done the layout you should now have a detailed description on what goes where. Most likely this will be on paper but hopefully someone will make a more automated system that can deal with everything from the design, through partitioning to formatting and installation. This is the route one will have to take to realise the design.

Modern distributions come with installation tools that will guide you through partitioning and formatting and also set up `/etc/fstab` for you automatically. For later modifications, however, you will need to understand the underlying mechanisms.

12.1 Checklist

Before starting make sure you have the following:

- Written notes of what goes where, your design
- A functioning, tested rescue disk
- A fresh backup of your precious data
- At least two formatted, tested and empty floppies
- Read and understood the man page for fdisk or equivalent
- Patience, concentration and elbow grease

12.2 Drives and Partitions

When you start DOS or the like you will find all partitions labeled `C:` and onwards, with no differentiation on IDE, SCSI, network or whatever type of media you have. In the world of Linux this is rather different. During booting you will see partitions described like this:

```
Dec  6 23:45:18 demos kernel: Partition check:
Dec  6 23:45:18 demos kernel:  sda: sda1
Dec  6 23:45:18 demos kernel:  hda: hda1 hda2
```

SCSI drives are labelled `sda`, `sdb`, `sdc` etc, and (E)IDE drives are labelled `hda`, `hdb`, `hdc` etc. There are also standard names for all devices, full information can be found in `/dev/MAKEDEV` and `/usr/src/linux/Documentation/devices.txt`.

Partitions are labelled numerically for each drive `hda1`, `hda2` and so on. On SCSI drives there can be 15 partitions per drive, on EIDE drives there can be 63 partitions per drive. Both limits exceed what is currently useful for most disks.

These are then mounted according to the file `/etc/fstab` before they appear as a part of the file system.

12.3 Partitioning

It feels so good / It's a marginal risk / when I clear off / windows with fdisk! (the Dustbunny in an issue of User Friendly in the song "Refund this")

First you have to partition each drive into a number of separate partitions. Under Linux there are two main methods, `fdisk` and the more screen oriented `cfdisk`. These are complex programs, read the manual *very* carefully. For the experts there is now also `sfdisk`.

Partitions come in 3 flavours, `primary`, `extended` and `logical`. You have to use `primary` partitions for booting, but there is a maximum of 4 primary partitions. If you want more you have to define an `extended` partition within which you define your `logical` partitions.

Each partition has an identifier number which tells the operating system what it is, for Linux the types `swap(82)` and `ext2fs(83)` are the ones you will need to know. If you want to use RAID with autostart you have to check the documentation for the appropriate type number for the RAID partition.

There is a readme file that comes with `fdisk` that gives more in−depth information on partitioning.

Someone has just made a *Partitioning HOWTO* which contains excellent, in depth information on the nitty−gritty of partitioning. Rather than repeating it here and bloating this document further, I will instead refer you to it instead.

Redhat has written a screen oriented utility called *Disk Druid* which is supposed to be a user friendly alternative to `fdisk` and `cfdisk` and also automates a few other things. Unfortunately this product is not quite mature so if you use it and cannot get it to work you are well advised to try `fdisk` or `cfdisk`.

Not to be outdone, Mandrakesoft has made an even more graphic alternative called Diskdrake that also offers numerous features.

Also the GNU project offers a partitioning tool called GNU Parted

The Ranish Partition Manager is another free alternative, while Partition Magic is a popular commercial alternative which also offers some support for resizing `ext2fs` partitions.

Note that Windows will complain if it finds more than one primary partition on a drive. Also it appears to assign drive letters to primary partitions as it finds disks before starting over from the first disk to assign subsequent drive names to logical partitions.

If you want DOS/Windows on your system you should make that partition first, a primary one to boot to, made with the DOS `fdisk` program. Then if you want NT you put that one in. Finally, for Linux, you create those partitions with the Linux `fdisk` program or equivalents. Linux is flexible enough to boot from both primary as well as logical partitions.

In depth information on DOS `fdisk` can be found at Fdisk.com and MS−DOS 5.00 − 7.10 Undocumented, Secret + Hidden Features which details even more bugs and pitfalls.

12.4 Repartitioning

Sometimes it is necessary to change the sizes of existing partitions while keeping the contents intact. One way is of course to back up everything, recreate new partitions and then restore the old contents, and while this gives your back up system a good test it is also rather time consuming.

Partition resizing is a simpler alternative where a file system is first shrunk to desired volume and then the partition table is updated to reflect the new end of partition position. This process is therefore very file system sensitive.

Repartitioning requires there to be free space at the end of the file space so to ensure you are able to shrink the size you should first defragment your drive and empty any wastebaskets.

Using fips you can resize a `fat` partition, and the latest version 1.6 of `fips` or `fips 2.0` are also able to resize `fat32` partition. Note that these programs actually run under DOS.

Resizing other file systems are much more complicated but one popular commercial system Partition Magic is able to resize more file system types, including `ext2fs` using the `resize2fs` program. Make sure you get the latest updates to this program as recent versions had problems with large disks.

In order to get the most out of `fips` you should first delete unnecessary files, empty wastebaskets etc. before defragmenting your drive. This way you can allocate more space to other partitions. If the program complains there are still files at the end of your drive it is probably hidden files generated by Microsoft Mirror or Norton Image. These are probably called `image.idx` and `image.dat` and contain backups of some system files.

There are reports that in some Windows defragmentation programs you should make sure the box "allow Windows to move files around" is *not* checked, otherwise you will end up with some files in the last cylinder of the partition which will prevent FIPS from reclaiming space.

If you still have unmovable files at the end of your DOS partition you should get the DOS program showfat version 3.0 or higher. This shows you what files are where so you can deal with them directly.

A freeware alternative is Partition Resizer which can shrink, grow and move partitions.

Some versions of DOS / Windows have a hidden flag for `defrag`, `"/P` that causes `defrag` to move even hidden files. Use at own risk.

Repartitioning is as dangerous process as any other partitioning so you are advised to have a fresh backup handy.

12.5 Microsoft Partition Bug

In Microsoft products all the way up to Win 98 there is a tricky bug that can cause you a bit of trouble: if you have several primary `fat` partitions and the last extended partition is not a `fat` partition the Microsoft system will try to mount the last partition as if it were a FAT partition in place of the last primary FAT partition.

There is more information available on the net on this.

To avoid this you can place a small logical `fat` partition at the very end of your disk.

More information on multi OS installations are available at V Communications but they keep rearranging the links continuously so no direct links can be offered here.

Since some hardware comes with setup software that is available under DOS only this could come in handy anyway. Notable examples are RAID controllers from DPT and a number of networking cards.

12.6 Multiple Devices (md)

Being in a state of flux you should make sure to read the latest documentation on this kernel feature. It is not yet stable, beware.

Briefly explained it works by adding partitions together into new devices `md0`, `md1` etc. using `mdadd` before you activate them using `mdrun`. This process can be automated using the file `/etc/mdtab`.

The latest `md` system uses a /etc/raidtab and a different syntax. Make sure your RAID−tools package matches the `md` version as the internal protocol has changed.

Then you then treat these like any other partition on a drive. Proceed with formatting etc. as described below using these new devices.

There is now also a HOWTO in development for RAID using `md` you should read.

12.7 Formatting

Next comes partition formatting, putting down the data structures that will describe the files and where they are located. If this is the first time it is recommended you use formatting with verify. Strictly speaking it should not be necessary but this exercises the I/O hard enough that it can uncover potential problems, such as incorrect termination, before you store your precious data. Look up the command `mkfs` for more details.

Linux can support a great number of file systems, rather than repeating the details you can read the man page for `fs` which describes them in some details. Note that your kernel has to have the drivers compiled in or made as modules in order to be able to use these features. When the time comes for kernel compiling you should read carefully through the file system feature list. If you use `make menuconfig` you can get online help for each file system type.

Note that some rescue disk systems require `minix`, `msdos` and `ext2fs` to be compiled into the kernel.

Also swap partitions have to be prepared, and for this you use `mkswap`.

Some important notes on formatting with DOS and Windows can be found in MS−DOS 5.00 − 7.10 Undocumented, Secret + Hidden Features.

Note that this formatting is high level formatting, that writes the file system to the disk, as opposed to low level formatting that lays down tracks and sectors. The latter is hardly ever needed these days.

12.8 Mounting

Data on a partition is not available to the file system until it is mounted on a mount point. This can be done manually using `mount` or automatically during booting by adding appropriate lines to `/etc/fstab`. Read the manual for `mount` and pay close attention to the tabulation.

12.9 `fstab`

During the booting process the system mounts all partitions as described in the `fstab` file which can look something like this:

```
# <file system>   <mount point>   <type>  <options>   <dump>  <pass>
/dev/hda2          /               ext2    defaults    0       1
None               none            swap    sw          0       0
proc               /proc           proc    defaults    0       0
/dev/hda1          /dosc           vfat    defaults    0       1
```

This file is somewhat sensitive to the formatting used so it is best and also most convenient to edit it using one of the editing tools made for this purpose, such as on the netfstool, a Tcl/Tk−based file system mounter, and kfstab, an editing tool for KDE.

Briefly, the fields are partition name, where to mount the partition, type of file system, mount options, when to dump for backup and when to do `fsck`.

Linux offers the possibility of parallel file checking (`fsck`) but to be efficient it is important not to `fsck` more than one partition on a drive at a time.

12.10 Mount options

Mounting, either by hand or using the `fstab`, allows for a number of options that offers extra protection. Below are some of the more useful options.

nodev

Do not interpret character or block special devices on the file system.

noexec

This disallows execution of any binaries on the mounted file system. Useful in spool areas.

nosuid

This disallows set−user−identifier or set−group−identifier on the mounted file system. Useful in home directories.

For more information and cautions refer to the man page for `mount` and `fstab`.

12.11 Recommendations

Having constructed and implemented your clever scheme you are well advised to make a complete record of it all, on paper. After all having all the necessary information on disk is no use if the machine is down.

Partition tables can be damaged or lost, in which case it is excruciatingly important that you enter the exact same numbers into `fdisk` so you can rescue your system. You can use the program `printpar` to make a clear record of the tables. Also write down the SCSI numbers or IDE names for each disk so you can put the system together again in the right order.

There is also a small script in appendix Appendix M: Disk System Documenter which will generate a summary of your disk configurations.

For checking your hard disks you can use the Disk Advisor boot disk available on the net. The disk builder required Windows to run. This system is useful to diagnose failed disks.

You are strongly recommended to make a rescue disk and *test* it. Most distributions make on available and is often part of the installation disks. For some, such as the one for Redhat 6.1 the way to invoke the disk as a rescue disk is to type *linux rescue* at the boot prompt.

There are also specialised rescue disk distributions available on the net.

When need for it comes you will need to know where your root and boot partitions reside which you need to write down and keep safe.

Note: the difference between a boot disk and a rescue disk is that a boot disk will fail if it cannot mount the file system, typically on your hard disk. A rescue disk is self contained and will work even if there are no hard disks.

---

13. Maintenance

It is the duty of the system manager to keep an eye on the drives and partitions. Should any of the partitions overflow, the system is likely to stop working properly, no matter how much space is available on other partitions, until space is reclaimed.

Partitions and disks are easily monitored using `df` and should be done frequently, perhaps using a cron job or some other general system management tool.

Do not forget the swap partitions, these are best monitored using one of the memory statistics programs such as `free`, `procinfo` or `top`.

Drive usage monitoring is more difficult but it is important for the sake of performance to avoid contention − placing too much demand on a single drive if others are available and idle.

It is important when installing software packages to have a clear idea where the various files go. As previously mentioned GCC keeps binaries in a library directory and there are also other programs that for historical reasons are hard to figure out, X11 for instance has an unusually complex structure.

When your system is about to fill up it is about time to check and prune old logging messages as well as hunt down core files. Proper use of `ulimit` in global shell settings can help saving you from having core files littered around the system.

13.1 Backup

The observant reader might have noticed a few hints about the usefulness of making backups. Horror stories are legio about accidents and what happened to the person responsible when the backup turned out to be non−functional or even non existent. You might find it simpler to invest in proper backups than a second, secret identity.

There are many options and also a mini−HOWTO (`Backup−With−MSDOS`) detailling what you need to know. In addition to the DOS specifics it also contains general information and further leads.

In addition to making these backups you should also make sure you can restore the data. Not all systems verify that the data written is correct and many administrators have started restoring the system after an accident happy in the belief that everything is working, only to discover to their horror that the backups were useless. Be careful.

There are both free and commercial backup systems available for Linux. One commercial example is the disk image level backup system from QuickStart offering a full function 30 day Linux demo available online.

13.2 Defragmentation

This is very dependent on the file system design, some suffer fast and nearly debilitating fragmentation. Fortunately for us, `ext2fs` does not belong to this group and therefore there has been very little talk about defragmentation tools. It does in fact exist but is hardly ever needed.

If for some reason you feel this is necessary, the quick and easy solution is to do a backup and a restore. If only a small area is affected, for instance the home directories, you could `tar` it over to a temporary area on another partition, *verify* the archive, delete the original and then untar it back again.

13.3 Deletions

Quite often disk space shortages can be remedied simply by deleting unnecessary files that accumulate around the system. Quite often programs that terminate abnormally cause all kinds of mess lying around the oddest places. Normally a core dump results after such an incident and unless you are going to debug it you can simply delete it. These can be found everywhere so you are advised to do a global search for them now and then. The `locate` command is useful for this.

Unexpected termination can also cause all sorts of temporary files remaining in places like `/tmp` or `/var/tmp`, files that are automatically removed when the program ends normally. Rebooting cleans up some of these areas but not necessary all and if you have a long uptime you could end up with a lot of old junk. If space is short you have to delete with care, make sure the file is not in active use first. Utilities like `file` can often tell you what kind of file you are looking at.

Many things are logged when the system is running, mostly to files in the `/var/log` area. In particular the file `/var/log/messages` tends to grow until deleted. It is a good idea to keep a small archive of old log files around for comparison should the system start to behave oddly.

If the mail or news system is not working properly you could have excessive growth in their spool areas, `/var/spool/mail` and `/var/spool/news` respectively. Beware of the overview files as these have a leading dot which makes them invisible to `ls −l`, it is always better to use `ls −Al` which will reveal them.

User space overflow is a particularly tricky topic. Wars have been waged between system administrators and users. Tact, diplomacy and a generous budget for new drives is what is needed. Make use of the message−of−the−day feature, information displayed during login from the `/etc/motd` file to tell users when space is short. Setting the default shell settings to prevent core files being dumped can save you a lot of work too.

Certain kinds of people try to hide files around the system, usually trying to take advantage of the fact that files with a leading dot in the name are invisible to the `ls` command. One common example are files that look like `...` that normally either are not seen, or, when using `ls −al` disappear in the noise of normal files like `.` or `..` that are in every directory. There is however a countermeasure to this, use `ls −Al` that suppresses `.` or `..` but shows all other dot−files.

13.4 Upgrades

No matter how large your drives, time will come when you will find you need more. As technology progresses you can get ever more for your money. At the time of writing this, it appears that 6.4 GB drives gives you the most bang for your bucks.

Note that with IDE drives you might have to remove an old drive, as the maximum number supported on your mother board is normally only 2 or some times 4. With SCSI you can have up to 7 for narrow (8−bit) SCSI or up to 15 for wide (15 bit) SCSI, per channel. Some host adapters can support more than a single channel and in any case you can have more than one host adapter per system. My personal recommendation is that you will most likely be better off with SCSI in the long run.

The question comes, where should you put this new drive? In many cases the reason for expansion is that you want a larger spool area, and in that case the fast, simple solution is to mount the drive somewhere under `/var/spool`. On the other hand newer drives are likely to be faster than older ones so in the long run you might find it worth your time to do a full reorganizing, possibly using your old design sheets.

If the upgrade is forced by running out of space in partitions used for things like `/usr` or `/var` the upgrade is a little more involved. You might consider the option of a full re−installation from your favourite (and hopefully upgraded) distribution. In this case you will have to be careful not to overwrite your essential setups. Usually these things are in the `/etc` directory. Proceed with care, fresh backups and working rescue disks. The other possibility is to simply copy the old directory over to the new directory which is mounted on a temporary mount point, edit your `/etc/fstab` file, reboot with your new partition in place and check that it works. Should it fail you can reboot with your rescue disk, re−edit `/etc/fstab` and try again.

Until volume management becomes available to Linux this is both complicated and dangerous. Do not get too surprised if you discover you need to restore your system from a backup.

The Tips−HOWTO gives the following example on how to move an entire directory structure across:

```
(cd /source/directory; tar cf − . ) | (cd /dest/directory; tar xvfp −)
```

While this approach to moving directory trees is portable among many Unix systems, it is inconvenient to remember. Also, it fails for deeply nested directory trees when pathnames become to long to handle for tar (GNU tar has special provisions to deal with long pathnames).

If you have access to GNU cp (which is always the case on Linux systems), you could as well use

```
cp −av /source/directory /dest/directory
```

GNU cp knows specifically about symbolic links, hard links, FIFOs and device files and will copy them correctly.

Remember that it might not be a good idea to try to transfer `/dev` or `/proc`.

There is also a Hard Disk Upgrade mini−HOWTO that gives you a step by step guide on migrating an entire Linux system, including LILO, form one hard disk to another.

13.5 Recovery

System crashes come in many and entertaining flavours, and partition table corruption always guarantees plenty of excitement. A recent and undoubtedly useful tool for those of us who are happy with the normal level of excitement, is gpart which means "Guess PC−Type hard disk partitions". Useful.

In addition there are some partition utilities available under DOS.

13.6 Rescue Disk

Upgrades of kernel and hardware is not uncommon in the Linux world and it is therefore important that you prepare an updated rescue disk especially when you use special drivers to access your hardware. Rescue disks can be gotten off the net, from your distribution or you can put one together yourself. Do make sure the boot and root parameters are set so the kernel will know where to find your system.

If you don't have a recovery floppy you can use the GRUB boot loader to load from a Linux kernel somewhere on disk, with arguments.

---

14. Advanced Issues

Linux and related systems offer plenty of possibilities for fast, efficient and devastating destruction. This document is no exception. With power comes dangers and the following sections describe a few more esoteric issues that should not be attempted before reading and understanding the documentation, the issues and the dangers. You should also make a backup. Also remember to try to restore the system from scratch from your backup at least once. Otherwise you might not be the first to be found with a perfect backup of your system and no tools available to reinstall it (or, even more embarrassing, some critical files missing on tape).

The techniques described here are rarely necessary but can be used for very specific setups. Think very clearly through what you wish to accomplish before playing around with this.

14.1 Hard Disk Tuning

The hard drive parameters can be tuned using the `hdparms` utility. Here the most interesting parameter is probably the read−ahead parameter which determines how much prefetch should be done in sequential reading.

If you want to try this out it makes most sense to tune for the characteristic file size on your drive but remember that this tuning is for the *entire* drive which makes it a bit more difficult. Probably this is only of use on large servers using dedicated news drives etc.

For safety the default hdparm settings are rather conservative. The disadvantage is that this mean you can get lost interrupts if you have a high frequency of IRQs as you would when using the serial port and an IDE disk as IRQs from the latter would mask other IRQs. This would be noticeable as less then ideal performance when downloading data from the net to disk. Setting `hdparm −u1 device` would prevent this masking and either improve your performance or, depending on hardware, corrupt the data on your disk. Experiment with caution and fresh backups.

For more information read the article The Need For Speed on tuning with `hdparm`.

14.2 File System Tuning

Most file systems come with a tuning utility and for `ext2fs` there is the `tune2fs` utility. Several parameters can be modified but perhaps the most useful parameter here is what size should be reserved and who should be able to take advantage of this which could help you getting more useful space out of your drives, possibly at the cost of less room for repairing a system should it crash.

14.3 Spindle Synchronizing

This should not in itself be dangerous, other than the peculiar fact that the exact details of the connections remain unclear for many drives. The theory is simple: keeping a fixed phase difference between the different drives in a RAID setup makes for less waiting for the right track to come into position for the read/write head. In practice it now seems that with large read−ahead buffers in the drives the effect is negligible.

Spindle synchronisation should not be used on RAID0 or RAID 0/1 as you would then lose the benefit of having the read heads over different areas of the mirrored sectors.

---

15. Troubleshooting

Much can go wrong and this is the start of a growing list of symptoms, problems and solutions:

15.1 During Installation

Locating Disks

Symptoms

Cannot find disk

Problem

How to find what drive letter corresponds to what disk/partition

Solution

Remember Linux does not use drive letters but device names. More information can be found in Drive names.

Symptoms

Cannot partition disk

Problem

Most likely wrong input to the command line for `fdisk` or similar tool.

Solution

Remember to use `/dev/hda` rather than just `hda`. Also do not use numbers behind `hda`, those indicate partitions.

Formatting

Symptoms

Cannot format disk.

Problem

Strictly speaking you format partitions not disks.

Solution

Make sure you add the partition number after the device name of the disk, for instance `/dev/hda1` to the command line.

15.2 During Booting

Booting fails

Symptoms

Number keep scrolling up the screen.

Problem

Possibly corrupt disk.

Solution

Try another disk, you might have to reinstall. Check for loose cables and possible data corruption.

Symptoms

Get `LI` and then it hangs.

Problem

You use LILO to load Linux but LILO cannot find your root.

Solution

Read the LILO HOWTO.

Symptoms

Kernel panics, something about missing root file system.

Problem

The kernel does not know where the root partition is.

Solution

Use `rdev` or (if applicable) LILO to add information to the kernel image where your root is.

Getting into Single User Mode

Symptoms

System boots but get into a root shell in single user mode.

Problem

Something went wrong in the later stages of booting and the system has come far enough to let you open a shell to repair the system.

Solution

Locate the problems from the boot log. Note that file system can be in read−only mode. Remount read−write if you have to. Often the reason is that the `/etc/fstab` contained an entry that was mismapped such as trying to mount a swap partition as your normal file space.

15.3 During Running

Swap

Symptoms

Short on memory

Problem

Swap space is not available

Solution

Type free and check the output. If you get

```
             total       used       free     shared    buffers     cached
Mem:         46920      30136      16784       7480      11788       5764
-/+ buffers/cache:      12584      34336
Swap:       128484       9176     119308
```

then system is running normal. If the line with `Swap:` contains zeros you have either not mounted the swap space (partition or swap file) (see `swapon(8)`) or not formatted the swap space (see `mkswap(8)`).

Partitions

Symptoms

No room amidst plenty 1

Problem

Partitionitis:Underdimensioned partition sizes has caused overflow in some areas

Solution

Examine your partition usage using `df(1)` and locate problem areas. Normally the problem can be solved by removing old junk but you might have to repartition your system, see section Repartitioning.

Symptoms

No room amidst plenty 2

Problem

Running out of i−nodes has caused overflow in some ares, often in areas with many small files such as news spool.

Solution

Examine your partition usage using `df -i` and locate problem areas. Normally the problem is solved by reformatting using a higher number of i−nodes, see `mkfs(8)` and related man pages.

16. Further Information

There is wealth of information one should go through when setting up a major system, for instance for a news or general Internet service provider. The FAQs in the following groups are useful:

16.1 News groups

Some of the most interesting news groups are:

- Storage.
- PC storage.
- AFS.
- SCSI.
- Linux setup.

Most newsgroups have their own FAQ that are designed to answer most of your questions, as the name Frequently Asked Questions indicate. Fresh versions should be posted regularly to the relevant newsgroups. If you cannot find it in your news spool you could go directly to the FAQ main archive FTP site. The WWW versions can be browsed at FAQ main archive WWW site.

Some FAQs have their own home site, of particular interest here are

- SCSI FAQ and
- comp.arch.storage FAQ.

16.2 Mailing Lists

These are low noise channels mainly for developers. Think twice before asking questions there as noise delays the development. Some relevant lists are `linux−raid`, `linux−scsi` and `linux−ext2fs`. Many of the most useful mailing lists run on the `vger.rutgers.edu` server but this is notoriously overloaded, so try to find a mirror. There are some lists mirrored at The Redhat Home Page. Many lists are also accessible at linuxhq, and the rest of the web site is a gold mine of useful information.

If you want to find out more about the lists available you can send a message with the line `lists` to the list server at vger.rutgers.edu (majordomo@vger.rutgers.edu). If you need help on how to use the mail server just send the line `help` to the same address. Due to the popularity of this server it is likely it takes a bit to time before you get a reply or even get messages after you send a `subscribe` command.

There is also a number of other majordomo list servers that can be of interest such as the EATA driver list (linux−eata@mail.uni−mainz.de) and the Intelligent IO list linux−i2o@dpt.com.

Mailing lists are in a state of flux but you can find links to a number of interesting lists from the Linux Documentation Homepage.

16.3 HOWTO

These are intended as the primary starting points to get the background information as well as show you how to solve a specific problem. Some relevant HOWTOs are `Bootdisk`, `Installation`, `SCSI` and `UMSDOS`. The main site for these is the LDP archive.

There is a a new HOWTO out that deals with setting up a DPT RAID system, check out the DPT RAID HOWTO homepage.

16.4 Mini−HOWTO

These are the smaller free text relatives to the HOWTOs. Some relevant mini−HOWTOs are `Backup−With−MSDOS`, `Diskless`, `LILO`, `Large Disk`, `Linux+DOS+Win95+OS2`, `Linux+OS2+DOS`, `Linux+Win95`, `NFS−Root`, `Win95+Win+Linux`, `ZIP Drive` . You can find these at the same place as the HOWTOs, usually in a sub directory called `mini`. Note that these are scheduled to be converted into SGML and become proper HOWTOs in the near future.

The old `Linux Large IDE mini−HOWTO` is no longer valid, instead read `/usr/src/linux/drivers/block/README.ide` or `/usr/src/linux/Documentation/ide.txt`.

16.5 Local Resources

In most distributions of Linux there is a document directory installed, have a look in the /usr/doc directory. where most packages store their main documentation and README files etc. Also you will here find the HOWTO archive (/usr/doc/HOWTO) of ready formatted HOWTOs and also the mini−HOWTO archive (/usr/doc/HOWTO/mini) of plain text documents.

Many of the configuration files mentioned earlier can be found in the /etc directory. In particular you will want to work with the /etc/fstab file that sets up the mounting of partitions and possibly also /etc/mdtab file that is used for the `md` system to set up RAID.

The kernel source in /usr/src/linux is, of course, the ultimate documentation. In other words, *use the source, Luke*. It should also be pointed out that the kernel comes not only with source code which is even commented (well, partially at least) but also an informative documentation directory. If you are about to ask any questions about the kernel you should read this first, it will save you and many others a lot of time and possibly embarrassment.

Also have a look in your system log file (/var/log/messages) to see what is going on and in particular how the booting went if too much scrolled off your screen. Using `tail −f /var/log/messages` in a separate window or screen will give you a continuous update of what is going on in your system.

You can also take advantage of the /proc file system that is a window into the inner workings of your system. Use `cat` rather than `more` to view the files as they are reported as being zero length. Reports are that `less` works well here.

16.6 Web Pages

There is a huge number of informative web pages out there and by their very nature they change quickly so don't be too surprised if these links become quickly outdated.

A good starting point is of course the Linux Documentation Homepage. that is a information central for documentation, project pages and much, much more.

- Mike Neuffer, the author of the DPT caching RAID controller drivers, has some interesting pages on SCSI and DPT.
- Software RAID development information can be found at Linux Kernel site along with patches and utilities.
- Disk related information on benchmarking, RAID, reliability and much, much more can be found at Linas Vepstas project page.
- There is also information available on how to RAID the root partition and what software packages are needed to achieve this.
- In depth documentation on ext2fs is also available.
- People who looking for information on VFAT, FAT32 and Joliet could have a look at the development page. These drivers are in the 2.1.x kernel development series as well as in 2.0.34 and later.

For diagrams and information on all sorts of disk drives, controllers etc. both for current and discontinued lines The Ref is the site you need. There is a lot of useful information here, a real treasure trove.

Please let me know if you have any other leads that can be of interest.

16.7 Search Engines

When all fails try the internet search engines. There is a huge number of them, all a little different from each other. It falls outside the scope of this HOWTO to describe how best to use them. Instead you could turn to the Troubleshooting on the Internet mini−HOWTO, and the Updated mini−HOWTO.

If you have to ask for help you are most likely to get help in the Linux Setup news group. Due to large workload and a slow network connection I am not able to follow that newsgroup so if you want to contact me you have to do so by e−mail.

---

17. Getting Help

In the end you might find yourself unable to solve your problems and need help from someone else. The most efficient way is either to ask someone local or in your nearest Linux user group, search the web for the nearest one.

Another possibility is to ask on Usenet News in one of the many, many newsgroups available. The problem is that these have such a high volume and noise (called low signal−to−noise ratio) that your question can easily fall through unanswered.

No matter where you ask it is important to ask well or you will not be taken seriously. Saying just *my disk does not work* is not going to help you and instead the noise level is increased even further and if you are lucky someone will ask you to clarify.

Instead describe your problems in some detail that will enable people to help you. The problem could lie somewhere you did not expect. Therefore you are advised to list up the following information on your system:

Hardware

- Processor
- DMA
- IRQ
- Chip set (LX, BX etc)
- Bus (ISA, VESA, PCI etc)
- Expansion cards used (Disk controllers, video, IO etc)

Software

- BIOS (On motherboard and possibly SCSI host adapters)
- LILO, if used
- Linux kernel version as well as possible modifications and patches
- Kernel parameters, if any
- Software that shows the error (with version number or date)

Peripherals

- Type of disk drives with manufacturer name, version and type
- Other relevant peripherals connected to the same busses

As an example of how interrelated these problems are: an old chip set caused problems with a certain combination of video controller and SCSI host adapter.

Remember that booting text is logged to `/var/log/messages` which can answer most of the questions above. Obviously if the drives fail you might not be able to get the log saved to disk but you can at least scroll back up the screen using the `SHIFT` and `PAGE UP` keys. It may also be useful to include part of this in your request for help but do not go overboard, keep it *brief* as a complete log file dumped to Usenet News is more than a little annoying.

---

18. Concluding Remarks

Disk tuning and partition decisions are difficult to make, and there are no hard rules here. Nevertheless it is a good idea to work more on this as the payoffs can be considerable. Maximizing usage on one drive only while the others are idle is unlikely to be optimal, watch the drive light, they are not there just for decoration. For a properly set up system the lights should look like Christmas in a disco. Linux offers software RAID but also support for some hardware base SCSI RAID controllers. Check what is available. As your system and experiences evolve you are likely to repartition and you might look on this document again. Additions are always welcome.

Finally I'd like to sum up my recommendations:

- Disks are cheap but the data they contain could be much more valuable, use and test your backup system.
- Work is also expensive, make sure you get large enough disks as refitting new or repartitioning old disks takes time.
- Think reliability, replace old disks before they fail.
- Keep a paper copy of your setup, having it all on disk when the machine is down will not help you much.

- Start out with a simple design with a minimum of fancy technology and rather fit it in later. In general adding is easier than replacing, be it disks, technology or other features.

18.1 Coming Soon

There are a few more important things that are about to appear here. In particular I will add more example tables as I am about to set up two fairly large and general systems, one at work and one at home. These should give some general feeling on how a system can be set up for either of these two purposes. Examples of smooth running existing systems are also welcome.

There is also a fair bit of work left to do on the various kinds of file systems and utilities.

There will be a big addition on drive technologies coming soon as well as a more in depth description on using `fdisk`, `cfdisk` and `sfdisk`. The file systems will be beefed up as more features become available as well as more on RAID and what directories can benefit from what RAID level.

There is some minor overlapping with the Linux Filesystem Structure Standard and FHS that I hope to integrate better soon, which will probably mean a big reworking of all the tables at the end of this document.

As more people start reading this I should get some more comments and feedback. I am also thinking of making a program that can automate a fair bit of this decision making process and although it is unlikely to be optimum it should provide a simpler, more complete starting point.

18.2 Request for Information

It has taken a fair bit of time to write this document and although most pieces are beginning to come together there are still some information needed before we are out of the beta stage.

- More information on swap sizing policies is needed as well as information on the largest swap size possible under the various kernel versions.
- How common is drive or file system corruption? So far I have only heard of problems caused by flaky hardware.
- References to speed and drives is needed.
- Are any other Linux compatible RAID controllers available?
- What relevant monitoring, management and maintenance tools are available?
- General references to information sources are needed, perhaps this should be a separate document?
- Usage of `/tmp` and `/var/tmp` has been hard to determine, in fact what programs use which directory is not well defined and more information here is required. Still, it seems at least clear that these should reside on different physical drives in order to increase paralellicity.

18.3 Suggested Project Work

Now and then people post on comp.os.linux.*, looking for good project ideas. Here I will list a few that comes to mind that are relevant to this document. Plans about big projects such as new file systems should still be posted in order to either find co−workers or see if someone is already working on it.

Planning tools

that can automate the design process outlines earlier would probably make a medium sized project, perhaps as an exercise in constraint based programming.

Partitioning tools

that take the output of the previously mentioned program and format drives in parallel and apply the appropriate symbolic links to the directory structure. It would probably be best if this were integrated in existing system installation software. The drive partitioning setup used in Solaris is an example of what it can look like.

Surveillance tools

that keep an eye on the partition sizes and warn before a partition overflows.

Migration tools

that safely lets you move old structures to new (for instance RAID) systems. This could probably be done as a shell script controlling a back up program and would be rather simple. Still, be sure it is safe and that the changes can be undone.

---

19. Questions and Answers

This is just a collection of what I believe are the most common questions people might have. Give me more feedback and I will turn this section into a proper FAQ.

- Q:How many physical disk drives (spindles) does a Linux system need?

  A: Linux can run just fine on one drive (spindle). Having enough RAM (around 32 MB, and up to 64 MB) to support swapping is a better price/performance choice than getting a second disk. (E)IDE disk is usually cheaper (but a little slower) than SCSI.
- Q: I have a single drive, will this HOWTO help me?

  A: Yes, although only to a minor degree. Still, section Physical Track Positioning will offer you some gains.
- Q: Are there any disadvantages in this scheme?

  A: There is only a minor snag: if even a single partition overflows the system might stop working properly. The severity depends of course on what partition is affected. Still this is not hard to monitor, the command `df` gives you a good overview of the situation. Also check the swap partition(s) using `free` to make sure you are not about to run out of virtual memory.
- Q: OK, so should I split the system into as many partitions as possible for a single drive?

  A: No, there are several disadvantages to that. First of all maintenance becomes needlessly complex and you gain very little in this. In fact if your partitions are too big you will seek across larger areas than needed. This is a balance and dependent on the number of physical drives you have.

- Q: Does that mean more drives allows more partitions?

  A: To some degree, yes. Still, some directories should not be split off from root, check out the file system standards for more details.

- Q: What if I have many drives I want to use?

  A: If you have more than 3−4 drives you should consider using RAID of some form. Still, it is a good idea to keep your root partition on a simple partition without RAID, see section RAID for more details.

- Q: I have installed the latest Windows95 but cannot access this partition from within the Linux system, what is wrong?

  A: Most likely you are using `FAT32` in your windows partition. It seems that Microsoft decided we needed yet another format, and this was introduced in their latest version of Windows95, called OSR2. The advantage is that this format is better suited to large drives.

  You might also be interested to hear that Microsoft NT 4.0 does not support it yet either.

- Q: I cannot get the disk size and partition sizes to match, something is missing. What has happened?

  A:It is possible you have mounted a partition onto a mount point that was not an empty directory. Mount points are directories and if it is not empty the mounting will mask the contents. If you do the sums you will see the amount of disk space used in this directory is missing from the observed total.

  To solve this you can boot from a rescue disk and see what is hiding behind your mount points and remove or transfer the contents by mounting the offending partition on a temporary mounting point. You might find it useful to have "spare" emergency mounting points ready made.

- Q: It doesn't look like my swap partition is in use, how come?

  A: It is possible that it has not been necessary to swap out, especially if you have plenty of RAM. Check your log files to see if you ran out of memory at one point or another, in that case your swap space should have been put to use. If not it is possible that either the swap partition was not assigned the right number, that you did not prepare it with `mkswap` or that you have not done `swapon` or added it to your /etc/fstab file.

- Q: What is this Nyx that is mentioned several times here?

  A: It is a large free Unix system with currently about 10000 users. I use it for my web pages for this HOWTO as well as a source of ideas for a setup of large Unix systems. It has been running for many years and has a quite stable setup. For more information you can view the Nyx homepage which also gives you information on how to get your own free account.

---

20. Bits and Pieces

This is basically a section where I stuff all the bits I have not yet decided where should go, yet that I feel is worth knowing about. It is a kind of transient area.

20.1 Swap Partition: to Use or Not to Use

In many cases you do not need a swap partition, for instance if you have plenty of RAM, say, more than 64 MB, and you are the sole user of the machine. In this case you can experiment running without a swap partition and check the system logs to see if you ran out of virtual memory at any point.

Removing swap partitions have two advantages:

- you save disk space (rather obvious really)
- you save seek time as swap partitions otherwise would lie in the middle of your disk space.

In the end, having a swap partition is like having a heated toilet: you do not use it very often, but you sure appreciate it when you require it.

20.2 Mount Point and `/mnt`

In an earlier version of this document I proposed to put all permanently mounted partitions under `/mnt`. That, however, is not such a good idea as this itself can be used as a mount point, which leads to all mounted partitions becoming unavailable. Instead I will propose mounting straight from root using a meaningful name like `/mnt.descriptive−name`.

Lately I have become aware that some Linux distributions use mount points at subdirectories *under* `/mnt`, such as `/mnt/floppy` and `/mnt/cdrom`, which just shows how confused the whole issue is. Hopefully FHS should clarify this.

20.3 Power and Heating

Not many years ago a machine with the equivalent power of a modern PC required 3−phase power and cooling, usually by air conditioning the machine room, some times also by water cooling. Technology has progressed very quickly giving not only high speed but also low power components. Still, there is a definite limit to the technology, something one should keep in mind as the system is expanded with yet another disk drive or PCI card. When the power supply is running at full rated power, keep in mind that all this energy is going somewhere, mostly into heat. Unless this is dissipated using fans you will get a serious heating inside the cabinet followed by a reduced reliability and also life time of the electronics. Manufacturers state minimum cooling requirements for their drives, usually in terms of cubic feet per minute (CFM). You are well advised to take this serious.

Keep air flow passages open, clean out dust and check the temperature of your system running. If it is too hot to touch it is probably running too hot.

If possible use sequential spin up for the drives. It is during spin up, when the drive platters accelerate up to normal speed, that a drive consumes maximum power and if all drives start up simultaneously you could go beyond the rated power maximum of your power supply.

20.4 Deja

This is an Internet system that no doubt most of you are familiar with. It searches and serves *Usenet News* articles from 1995 and to the latest postings and also offers a web based reading and posting interface. There is a lot more, check out Deja for more information. It changed name from Dejanews.

What perhaps is less known, is that they use about 120 Linux SMP computers many of which use the `md` module to manage between 4 and 24 Gig of disk space (over 1200 Gig altogether) for this service. The system is continuously growing but at the time of writing they use mostly dual Pentium Pro 200MHz and Pentium II 300 MHz systems with 256 MB RAM or more.

A production database machine normally has 1 disk for the operating system and between 4 and 6 disks managed by the `md` module where the articles are archived. The drives are connected to BusLogic Model BT−946C and BT−958 PCI SCSI adapters, usually one to a machine.

For the production systems (which are up 365 days a year) the downtime attributable to disk errors is less than 0.25 % (that is a quarter of 1%, not 25%).

Just in case: this is not an advertisement, it is stated as an example of how much is required for what is a major Internet service.

20.5 Crash Recovery

Occationally hard disks crash. A crash causing data scrambling can often be at least partially recovered from and there are already HOWTOs describing this.

In case of hardware failure things are far more serious, and you have two options: either send the drive to a professional data recovery company, or try recovering yourself. The latter is of course *high risk* and can cause more damage.

If a disk stops rotating or fails to spin up, the number one advice is first to turn off the system as fast as safely possible.

Next you could try disconnecting the drives and power up the machine, just to check power with a multimeter that power is present. Quite often connectors can get unseated and cause all sorts of problems.

If you decide to risk trying it yourself you could check all connectors and then reapply power and see if the drive spins up and responds. If it still is dead turn off power quickly, preferrably before the operating system boots. Make sure that delayed spinup is not deceiving you here.

If you decide to progress even further (and take higher risks) you could remove the drive, give it a firm tap on the side so that the disk moves a little with respect to the casing. This can help in unsticking the head from the surface, allowing the platter to move freely as the motor power is not sufficient to unstick a stuck head on its own.

Also if a drive has been turned off for a while after running for long periods of time, or if it has overheated, the lubricant can harden of drain out of the bearings. In this case warming the drive slowly and gently up to normal operating temperature will possibly recover the lubrication problems.

If after this the drive still does not respond the last possible and the highest risk suggestion is to replace the circuit board of the drive with a board from am identical model drive.

Often the contents of a drive is worth far more than the media itself, so do consider professional help. These companies have advanced equipment and know−how obtained from the manufacturers on how to recover a damaged drive, far beyond that of a hobbyist.

21. Appendix A: Partitioning Layout Table: Mounting and Linking

The following table is designed to make layout a simpler paper and pencil exercise. It is probably best to print it out (using NON PROPORTIONAL fonts) and adjust the numbers until you are happy with them.

Mount point is what directory you wish to mount a partition on or the actual device. This is also a good place to note how you plan to use symbolic links.

The size given corresponds to a fairly big Debian 1.2.6 installation. Other examples are coming later.

Mainly you use this table to select what structure and drives you will use, the partition numbers and letters will come from the next two tables.

Directory	Mount point	speed	seek	transfer	size	SIZE
swap	__________	ooooo	ooooo	ooooo	32	____
/	__________	o	o	o	20	____
/tmp	__________	oooo	oooo	oooo		____
/var	__________	oo	oo	oo	25	____
/var/tmp	__________	oooo	oooo	oooo		____
/var/spool	__________					____
/var/spool/mail	__________	o	o	o		____
/var/spool/news	__________	ooo	ooo	oo		____
/var/spool/____	__________	____	____	____		____
/home	__________	oo	oo	oo		____
/usr	__________				500	____
/usr/bin	__________	o	oo	o	250	____
/usr/lib	__________	oo	oo	ooo	200	____
/usr/local	__________					____
/usr/local/bin	__________	o	oo	o		____
/usr/local/lib	__________	oo	oo	ooo		____
/usr/local/____	__________					____
/usr/src	__________	o	oo	o	50	____
DOS	__________	o	o	o		____
Win	__________	oo	oo	oo		____
NT	__________	ooo	ooo	ooo		____
/mnt._________	__________	____	____	____		____
/mnt._________	__________	____	____	____		____
/mnt._________	__________	____	____	____		____
/_____________	__________	____	____	____		____
/_____________	__________	____	____	____		____
/_____________	__________	____	____	____		____

Total capacity:

22. Appendix B: Partitioning Layout Table: Numbering and Sizing

This table follows the same logical structure as the table above where you decided what disk to use. Here you select the physical tracking, keeping in mind the effect of track positioning mentioned earlier in Physical Track Positioning.

The final partition number will come out of the table after this.

```
  Drive           sda     sdb     sdc     hda     hdb     hdc     ___

SCSI ID         |  __   |  __   |  __   |

Directory
swap            |       |       |       |       |       |       |

/               |       |       |       |       |       |       |

/tmp            |       |       |       |       |       |       |

/var            :       :       :       :       :       :       :
/var/tmp        |       |       |       |       |       |       |
/var/spool      :       :       :       :       :       :       :
/var/spool/mail |       |       |       |       |       |       |
/var/spool/news :       :       :       :       :       :       :
/var/spool/____ |       |       |       |       |       |       |

/home           |       |       |       |       |       |       |

/usr            |       |       |       |       |       |       |
/usr/bin        :       :       :       :       :       :       :
/usr/lib        |       |       |       |       |       |       |
/usr/local      :       :       :       :       :       :       :
/usr/local/bin  |       |       |       |       |       |       |
/usr/local/lib  :       :       :       :       :       :       :
/usr/local/____ |       |       |       |       |       |       |
/usr/src        :       :       :       :

DOS             |       |       |       |       |       |       |
Win             :       :       :       :       :       :       :
NT              |       |       |       |       |       |       |

/mnt.___/_____  |       |       |       |       |       |       |
/mnt.___/_____  :       :       :       :       :       :       :
/mnt.___/_____  |       |       |       |       |       |       |
/_____________  :       :       :       :       :       :       :
/_____________  |       |       |       |       |       |       |
/_____________  :       :       :       :       :       :       :


Total capacity:
```

---

23. Appendix C: Partitioning Layout Table: Partition Placement

This is just to sort the partition numbers in ascending order ready to input to fdisk or cfdisk. Here you take physical track positioning into account when finalizing your design. Unless you get specific information otherwise, you can assume track 0 is the outermost track.

These numbers and letters are then used to update the previous tables, all of which you will find very useful in later maintenance.

In case of disk crash you might find it handy to know what SCSI id belongs to which drive, consider keeping a paper copy of this.

```
        Drive :   sda     sdb     sdc     hda     hdb     hdc     ___

Total capacity: |  ___  |  ___  |  ___  |  ___  |  ___  |  ___  |  ___
SCSI ID         |  __   |  __   |  __   |

Partition

1               |       |       |       |       |       |       |
2               :       :       :       :       :       :       :
3               |       |       |       |       |       |       |
4               :       :       :       :       :       :       :
5               |       |       |       |       |       |       |
6               :       :       :       :       :       :       :
7               |       |       |       |       |       |       |
8               :       :       :       :       :       :       :
9               |       |       |       |       |       |       |
10              :       :       :       :       :       :       :
11              |       |       |       |       |       |       |
12              :       :       :       :       :       :       :
13              |       |       |       |       |       |       |
14              :       :       :       :       :       :       :
15              |       |       |       |       |       |       |
16              :       :       :       :       :       :       :
```

---

24. Appendix D: Example: Multipurpose Server

The following table is from the setup of a medium sized multipurpose server where I once worked. Aside from being a general Linux machine it will also be a network related server (DNS, mail, FTP, news, printers etc.) X server for various CAD programs, CD ROM burner and many other things. The files reside on 3 SCSI drives with a capacity of 600, 1000 and 1300 MB.

Some further speed could possibly be gained by splitting `/usr/local` from the rest of the `/usr` system but we deemed the further added complexity would not be worth it. With another couple of drives this could be more worthwhile. In this setup drive sda is old and slow and could just a well be replaced by an IDE drive. The other two drives are both rather fast. Basically we split most of the load between these two. To reduce dangers of imbalance in partition sizing we have decided to keep `/usr/bin` and `/usr/local/bin` in one drive and `/usr/lib` and `/usr/local/lib` on another separate drive which also affords us some drive parallelizing.

Even more could be gained by using RAID but we felt that as a server we needed more reliability than was then afforded by the `md` patch and a dedicated RAID controller was out of our reach.

---

25. Appendix E: Example: Mounting and Linking

Directory	Mount point	speed	seek	transfer	size	SIZE
swap	sdb2, sdc2	ooooo	ooooo	ooooo	32	2x64
/	sda2	o	o	o	20	100
/tmp	sdb3	oooo	oooo	oooo		300
/var	___________	oo	oo	oo		____
/var/tmp	sdc3	oooo	oooo	oooo		300
/var/spool	sdb1					436
/var/spool/mail	___________	o	o	o		____
/var/spool/news	___________	ooo	ooo	oo		____
/var/spool/____	___________	____	____	____		____
/home	sda3	oo	oo	oo		400
/usr	sdb4				230	200
/usr/bin	___________	o	oo	o	30	____
/usr/lib	-> libdisk	oo	oo	ooo	70	____
/usr/local	___________					____
/usr/local/bin	___________	o	oo	o		____
/usr/local/lib	-> libdisk	oo	oo	ooo		____
/usr/local/____	___________					____
/usr/src	->/home/usr.src	o	oo	o	10	____
DOS	sda1	o	o	o		100
Win	___________	oo	oo	oo		____
NT	___________	ooo	ooo	ooo		____
/mnt.libdisk	sdc4	oo	oo	ooo		226
/mnt.cd	sdc1	o	o	oo		710

Total capacity: 2900 MB

---

26. Appendix F: Example: Numbering and Sizing

Here we do the adjustment of sizes and positioning.

```
Directory         sda     sdb     sdc


swap            |       |   64  |   64  |

/               |  100  |       |       |

/tmp            |       |  300  |       |

/var            :       :       :       :
/var/tmp        |       |       |  300  |
/var/spool      :       :  436  :       :
/var/spool/mail |       |       |       |
/var/spool/news :       :       :       :
/var/spool/____ |       |       |       |
/home            |  400  |       |       |

/usr             |       |  200  |       |
/usr/bin         :       :       :       :
/usr/lib         |       |       |       |
/usr/local       :       :       :       :
/usr/local/bin   |       |       |       |
/usr/local/lib   :       :       :       :
/usr/local/____  |       |       |       |
/usr/src         :       :       :       :

DOS              |  100  |       |       |
Win              :       :       :       :
NT               |       |       |       |

/mnt.libdisk     |       |       |  226  |
/mnt.cd          :       :       :  710  :
/mnt.___/_____   |       |       |       |


Total capacity:  |  600  | 1000  | 1300  |
```

---

27. Appendix G: Example: Partition Placement

This is just to sort the partition numbers in ascending order ready to input to fdisk or cfdisk. Remember to optimize for physical track positioning (not done here).

```
              Drive :   sda     sdb     sdc

      Total capacity:  |   600 |  1000 |  1300 |

      Partition

      1                |   100 |   436 |   710 |
      2                :   100 :    64 :    64 :
      3                |   400 |   300 |   300 |
      4                :       :   200 :   226 :
```

---

28. Appendix H: Example II

The following is an example of a server setup in an academic setting, and is contributed by `nakano (at) apm.seikei.ac.jp`. I have only done minor editing to this section.

`/var/spool/delegate` is a directory for storing logs and cache files of an WWW proxy server program, "delegated". Since I don't notice it widely, there are 1000−−1500 requests/day currently, and average disk usage is 15−−30% with expiration of caches each day.

`/mnt.archive` is used for data files which are big and not frequently referenced such a s experimental data (especially graphic ones), various source archives, and Win95 backups (growing very fast...).

`/mnt.root` is backup root file system containing rescue utilities. A boot floppy is also prepared to boot with this partition.

```
=================================================
Directory               sda      sdb     hda

swap                    |    64 |    64 |       |
/                       |       |       |    20 |
/tmp                    |       |       |   180 |

/var                    :   300 :       :       :
/var/tmp                |       |   300 |       |
/var/spool/delegate     |   300 |       |       |

/home                   |       |       |   850 |
/usr                    |   360 |       |       |
/usr/lib                -> /mnt.lib/usr.lib
/usr/local/lib          -> /mnt.lib/usr.local.lib

/mnt.lib                |       |   350 |       |
/mnt.archive            :       :  1300 :       :
/mnt.root               |       |    20 |       |

Total capacity:            1024    2034    1050


=================================================
        Drive :           sda     sdb     hda
Total capacity:         |  1024 |  2034 |  1050 |

Partition
1                       |   300 |    20 |    20 |
2                       :    64 :  1300 :   180 :
3                       |   300 |    64 |   850 |
4                       :   360 :   ext :       :
5                       |       |   300 |       |
6                       :       :   350 :       :


Filesystem         1024−blocks  Used Available Capacity Mounted on
/dev/hda1              19485   10534     7945     57%   /
/dev/hda2             178598      13   169362      0%   /tmp
/dev/hda3             826640  440814   343138     56%   /home
/dev/sda1             306088   33580   256700     12%   /var
/dev/sda3             297925   47730   234807     17%   /var/spool/delegate
/dev/sda4             363272  170872   173640     50%   /usr
/dev/sdb5             297598       2   282228      0%   /var/tmp
/dev/sdb2            1339248  302564   967520     24%   /mnt.archive
/dev/sdb6             323716   78792   228208     26%   /mnt.lib
```

Apparently `/tmp` and `/var/tmp` is too big. These directories shall be packed together into one partition when disk space shortage comes.

`/mnt.lib` is also seemed to be, but I plan to install newer TeX and ghostscript archives, so `/usr/local/lib` may grow about 100 MB or so (since we must use Japanese fonts!).

Whole system is backed up by Seagate Tapestore 8000 (Travan TR−4, 4G/8G).

---

29. Appendix I: Example III: SPARC Solaris

The following section is the basic design used at work for a number of Sun SPARC servers running Solaris 2.5.1 in an industrial development environment. It serves a number of database and cad applications in addition to the normal services such as mail.

Simplicity is emphasized here so `/usr/lib` has not been split off from `/usr`.

This is the basic layout, planned for about 100 users.

```
Drive:         SCSI 0                       SCSI 1

Partition      Size (MB)   Mount point     Size (MB)   Mount point

  0            160         swap            160         swap
  1            100         /tmp            100         /var/tmp
  2            400         /usr
  3            100         /
  4             50         /var
  5
  6            remainder   /local0         remainder   /local1
```

Due to specific requirements at this place it is at times necessary to have large partitions available on a short notice. Therefore drive 0 is given as many tasks as feasible, leaving a large `/local1` partition.

This setup has been in use for some time now and found satisfactorily.

For a more general and balanced system it would be better to swap `/tmp` and `/var/tmp` and then move `/var` to drive 1.

---

30. Appendix J: Example IV: Server with 4 Drives

This gives an example of using all techniques described earlier, short of RAID. It is admittedly rather complicated but offers in return high performance from modest hardware. Dimensioning are skipped but reasonable figures can be found in previous examples.

```
Partition       sda             sdb             sdc             sdd
                ----            ----            ----            ----
        1       root            overview        lib             news
        2       swap            swap            swap            swap
        3       home            /usr            /var/tmp        /tmp
        4                       spare root      mail            /var
```

Setup is optimised with respect to track positioning but also for minimising drive seeks.

If you want DOS or Windows too you will have to use `sda1` for this and move the other partitions after that. It will be advantageous to use the swap partitions on `sdb2`, `sdc2` and `sdd2` for Windows swap, `TEMPDIR` and Windows temporary directory under these sessions. A number of other HOWTOs describe how you can make several operating systems coexist on your machine.

For completeness a 4 drive example using several types of RAID is also given which is even more complex than the example above.

```
Partition       sda             sdb             sdc             sdd
                ----            ----            ----            ----
        1       boot            overview        news            news
        2       overview        swap            swap            swap
        3       swap            lib             lib             lib
        4       lib             overview        /tmp            /tmp
        5       /var/tmp        /var/tmp        mail            /usr
        6       /home           /usr            /usr            mail
        7       /usr            /home           /var
        8       / (root)        spare root
```

Here all duplicates are parts of a RAID 0 set with two exceptions, swap which is interleaved and home and mail which are implemented as RAID 1 for safety.

Note that boot and root are separated: only the boot file with the kernel has to reside within the 1023 cylinder limit. The rest of the root files can be anywhere and here they are placed on the slowest outermost partition. For simplicity and safety the root partition is not on a RAID system.

With such a complicated comes an equally complicated `fstab` file. The large number of partitions makes it important to do the `fsck` passes in the right order, otherwise the process can take perhaps ten times as long time to complete as the optimal solution.

```
/dev/sda8       /               ?       ?               1 1 (a)
/dev/sdb8       /               ?       noauto          1 2 (b)
/dev/sda1       boot            ?       ?               1 2 (a)
/dev/sdc7       /var            ?       ?               1 2 (c)
/dev/md1        news            ?       ?               1 3 (c+d)
/dev/md2        /var/tmp        ?       ?               1 3 (a+b)
/dev/md3        mail            ?       ?               1 4 (c+d)
/dev/md4        /home           ?       ?               1 4 (a+b)
/dev/md5        /tmp            ?       ?               1 5 (c+d)
/dev/md6        /usr            ?       ?               1 6 (a+b+c+d)
/dev/md7        /lib            ?       ?               1 7 (a+b+c+d)
```

The letters in the brackets indicate what drives will be active for each `fsck` entry and pass. These letters are *not* present in a real `fstab` file. All in all there are 7 passes.

---

31. Appendix K: Example V: Dual Drive System

A dual drive system offers less opportunity for clever schemes but the following should provide a simple starting point.

```
Partition       sda             sdb
                ----            ----
        1       boot            lib
        2       swap            news
        3       /tmp            swap
        4       /usr            /var/tmp
        5       /var            /home
        6       / (root)
```

If you use a dual OS system you have to keep in mind that many other systems must boot from the first partition on the first drive. A simple DOS / Linux system could look like this:

```
Partition       sda             sdb
                ----            ----
        1       DOS             lib
        2       boot            news
        3       swap            swap
        4       /tmp            /var/tmp
        5       /usr            /home
        6       /var            DOSTEMP
        7       / (root)
```

Also remember that DOS and Windows prefer there to be just a single primary partition which has to be the first one where it boots from. As Linux can happily exist in logical partitions this is not a big problem.

32. Appendix L: Example VI: Single Drive System

Although this falls somewhat outside the scope of this HOWTO it cannot be denied that recently some rather large drives have become very affordable. Drives with 10 − 20 GB are becoming common and the question often is how best to partition such monsters. Interestingly enough very few seem to have any problems in filling up such drives and the future looks generally quite rosy for manufacturers planning on even bigger drives.

Opportunities for optimisations are of course even smaller than for 2 drive systems but some tricks can still be used to optimise track positions while minimising head movements.

```
Partition       hda             Size estimate (MB)
                ----            ------------------
         1      DOS             500
         2      boot            20
         3      Winswap         200
         4      data            The bulk of the drive
         5      lib             50 - 500
         6      news            300+
         7      swap            128     (Maximum size for 32-bit CPU)
         8      tmp             300+    (/tmp and /var/tmp)
         9      /usr            50 - 500
        10      /home           300+
        11      /var            50 - 300
        12      mail            300+
        13      / (root)        30
        14      dosdata         10      ( Windows bug workaround!)
```

Remember that the `dosdata` partition is a DOS filesystem that must be the very last partition on the drive, otherwise Windows gets confused.

33. Appendix M: Disk System Documenter

This shell script was very kindly provided by Steffen Hulegaard. Run it as root (superuser) and it will generate a summary of your disk setup. Run it after you have implemented your design and compare it with what you designed to check for mistakes. Should your system develop defects this document will also be a useful starting point for recovery.

```
#!/bin/bash
#$Header$
#
# makediskdoc                Collects storage/disk info via df, mount,
#                            /etc/fstab and fdisk.  Creates a single
#                            reference file −− /root/sysop/doc/README.diskdoc
#                            Especially good for documenting storage
#                            config/partioning
#
# 11/11/1999  SC Hulegaard  Created just before RedHat 5.2 to
#                            RedHat 6.1 upgrade
# 12/31/1999  SC Hulegaard  Added sfdisk −glx usage just prior to
#                            collapse of my Quantum Grand Prix (4.3 Gb)
#
# SEE ALSO  Other /root/bin/make*doc commands to produce other /root/sysop/doc/README.*
#           files.  For example, /root/bin/makenetdoc.
#
FILE=/root/sysop/doc/README.diskdoc
echo Creating $FILE ...
echo ' ' > $FILE
echo $FILE >> $FILE
echo Produced By $0 >> $FILE
echo `date` >> $FILE
echo ' ' >> $FILE
echo $Header$ >> $FILE
echo ' ' >> $FILE
echo DESCRIPTION:  df −a >> $FILE
df −a >> $FILE 2>&1
echo ' ' >> $FILE
echo DESCRIPTION:  df −ia >> $FILE
df −ia >> $FILE 2>&1
echo ' ' >> $FILE
echo DESCRIPTION:  mount >> $FILE
mount >> $FILE 2>&1
echo ' ' >> $FILE
echo DESCRIPTION:  /etc/fstab >> $FILE
cat /etc/fstab >> $FILE
echo ' ' >> $FILE
echo DESCRIPTION:  sfdisk −s disk device size summary >> $FILE
sfdisk −s >> $FILE
echo ' ' >> $FILE
echo DESCRIPTION:  sfdisk −glx info for all disks listed in /etc/fstab >> $FILE
for x in `cat /etc/fstab | egrep /dev/[sh] | cut −c 0−8 | uniq`; do
  echo ' ' >> $FILE
  echo $x ============================ >> $FILE
  sfdisk −glx $x >> $FILE
done
echo ' ' >> $FILE
echo DESCRIPTION:  fdisk −l info for all disks listed in /etc/fstab >> $FILE
for x in `cat /etc/fstab | egrep /dev/[sh] | cut −c 0−8 | uniq`; do
  echo ' ' >> $FILE
  echo $x ============================ >> $FILE
  fdisk −l $x >> $FILE
done
echo ' ' >> $FILE
echo DESCRIPTION:  dmesg info on both sd and hd drives >> $FILE
dmesg | egrep [hs]d[a−z] >> $FILE
echo '' >> $FILE
echo Done >> $FILE
echo Done
exit
```