Linux Disk HOWTO

Stein Gjoen, sgjoen@nyx.net v0.13c, 12 August 1997

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.

Contents

1 Introduction **5**

1.1 Copyright . . . 5

1.2 Disclaimer . . . 6

1.3 News . . . 6

1.4 Credits . . . 7

2 Structure **7**

2.1 Logical structure . . . 8

2.2 Document structure . . . 8

3 Drive technologies **8**

3.1 Drives . . . 9

3.2 Geometry . . . 9

3.3 Media . . . 9

3.3.1 Magnetic Drives . . . 9

3.3.2 Optical drives . . . 10

3.3.3 Solid State Drives . . . 10

3.4 Interfaces . . . 11

3.4.1 MFM and RLL . . . 11

3.4.2 ESDI . . . 11

3.4.3 IDE and ATA . . . 11

3.4.4 EIDE, Fast-ATA and ATA-2 . . . 12

3.4.5 ATAPI . . . 12

3.4.6 SCSI . . . 12

3.5 Cabling . . . 12

3.6 Host Adapters . . . 13
3.7 Multi Channel Systems . . . 13
3.8 Comparisons . . . 13
3.9 Future Development . . . 14
3.10 Recommendations . . . 14

4 Considerations **15**

4.1 File system features . . . 15
4.1.1 Swap . . . 15
4.1.2 Temporary storage (`/tmp` and `/var/tmp`) . . . 16
4.1.3 Spool areas (`/var/spool/news` and `/var/spool/mail`) . . . 17
4.1.4 Home directories (`/home`) . . . 17
4.1.5 Main binaries (`/usr/bin` and `/usr/local/bin`) . . . 18
4.1.6 Libraries (`/usr/lib` and `/usr/local/lib`) . . . 18
4.1.7 Root . . . 19
4.1.8 DOS etc. . . . 20
4.2 Explanation of terms . . . 20
4.2.1 Speed . . . 20
4.2.2 Reliability . . . 21
4.2.3 Files . . . 21
4.3 Technologies . . . 22
4.3.1 RAID . . . 22
4.3.2 AFS, Veritas and Other Volume Management Systems . . . 23
4.3.3 Linux `md` Kernel Patch . . . 23
4.3.4 General File System Consideration . . . 23
4.3.5 CD-ROM File Systems . . . 24
4.3.6 Compression . . . 24
4.3.7 Physical Track Positioning . . . 25

5 Other Operating System **26**

5.1 DOS . . . 26
5.2 Windows . . . 27
5.3 OS/2 . . . 28

5.4 NT . . . 28
5.5 Sun OS . . . 28
5.5.1 Sun OS 4 . . . 28
5.5.2 Sun OS 5 (aka Solaris) . . . 28

6 Clusters **29**

7 Mounting Points **31**

8 Disk Layout **32**
8.1 Selection . . . 32
8.2 Mapping . . . 32
8.3 Optimizing . . . 33
8.3.1 Optimizing by characteristics . . . 33
8.3.2 Optimizing by drive parallelising . . . 33
8.4 Usage requirements . . . 34
8.5 Servers . . . 35
8.5.1 Home directories . . . 35
8.5.2 Anonymous FTP . . . 36
8.5.3 WWW . . . 36
8.5.4 Mail . . . 36
8.5.5 News . . . 36
8.5.6 Others . . . 37
8.6 Pitfalls . . . 37
8.7 Compromises . . . 37

9 Implementation **38**
9.1 Drives and Partitions . . . 38
9.2 Partitioning . . . 38
9.3 Multiple devices (`md`) . . . 39
9.4 Formatting . . . 39
9.5 Mounting . . . 40

10 Maintenance **40**

10.1 Backup . . . 40

10.2 Defragmentation . . . 41

10.3 Upgrades . . . 41

11 Advanced Issues **41**

11.1 Hard Disk Tuning . . . 42

11.2 File System Tuning . . . 42

12 Further Information **42**

13 Concluding Remarks **44**

13.1 Coming Soon . . . 44

13.2 Request for Information . . . 45

13.3 Suggested Project Work . . . 45

14 Questions and Answers **46**

15 Bits and Pieces **47**

15.1 Combining `swap` and `/tmp` . . . 47

15.2 Interleaved `swap` drives. . . . 48

15.3 Swap partition: to use or not to use . . . 48

15.4 Mount point and `/mnt` . . . 48

15.5 SCSI id numbers and names . . . 48

15.6 Power and Heating . . . 49

15.7 Dejanews . . . 49

15.8 File system structure . . . 49

15.9 Track numbering and optimizing schemes . . . 50

16 Appendix A: Partitioning layout table: mounting and linking **50**

17 Appendix B: Partitioning layout table: numbering and sizing **51**

18 Appendix C: Partitioning layout table: partition placement **52**

19 Appendix D: Example: Multipurpose server **53**

20 Appendix E: Example: mounting and linking **53**

21 Appendix F: Example: numbering and sizing **54**

22 Appendix G: Example: partition placement **55**

23 Appendix H: Example II **55**

24 Appendix I: Example III: SPARC Solaris **57**

1 Introduction

In commemoration of the "*Linux Hacker V2.0 - The New Generation*" this brand new release is code named the **Patricia Miranda** release. After all, this is a growing project... 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.

Even though it is now more than a year since last release of the FSSTND work is still continuing, under a new name, and will encompass more than Linux, fill in a few blanks hinted at in FSSTND version 1.2 as well as other general improvements. The development mailing list is currently private but a general release is hopefully in the near future. The new issue will be named Filesystem Hierarchy Standard (FHS) and will cover more than Linux alone.

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 Mini-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

This HOWTO is copyrighted 1996 Stein Gjoen.

Unless otherwise stated, Linux HOWTO documents are copyrighted by their respective authors. Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or

electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs.

If you have questions, please contact Greg Hankins, the Linux HOWTO coordinator, at gregh@sunsite.unc.edu via email.

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.

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

1.3 News

Since the 0.12 version was released there have been too many changes to list here. The document has grown a lot, actually beyond expectations. There are many new chapters, old sections expanded into separate chapters and many other improvements.

I have also upgraded my system to Debian 1.2.6 and have replaced the old Slackware values with the Debian values for disk space requirements for the various directory. I will use Debian as a base for discussions and examples here, though the HOWTO is equally applicable to other distributions, even other operating systems. At the time of writing this Debian 1.3 is out in beta and will soon be used as the test bench for further versions of this document.

More news: there has been a fair bit of interest in new kinds of file systems in the comp.os.linux newsgroups, in particular logging, journaling and inherited file systems. Watch out for updates. Projects on volume management is also under way. The old defragmentation program for `ext2fs` is being updated and there is continuing interests for compression.

The latest version number of this document can be gleaned from my plan entry if you do "finger sgjoen@nox.nyx.net"

Also, the latest version will be available on my web space on nyx: *The Multiple Disk System Tuning mini-HOWTO Homepage* `<http://www.nyx.net/ sgjoen/disk.html>`.

A text-only version as well as the SGML source can also be downloaded there. A nicely formatted postscript version is also available now.

Also planned is a series of URLs to helpful software referred to in this document. A mirror in Europe will be announced soon.

1.4 Credits

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

ronnej@ucs.orst.edu
cm@kukuruz.ping.at
armbru@pond.sub.org
R.P.Blake@open.ac.uk
neuffer@goofy.zdv.Uni-Mainz.de
sjmudd@redestb.es
nat@nataa.fr.eu.org
sundbyk@horten.geco-prakla.slb.com
gjoen@sn.no
mike@i-Connect.Net
roth@uiuc.edu
phall@ilap.com
szaka@mirror.cc.u-szeged.hu
CMckeon@swcp.com

Special thanks go to `nakano@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.

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...

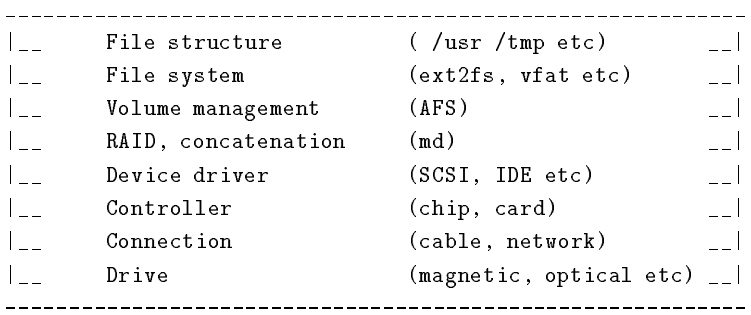
2 Structure

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.



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.

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* <http://thef-nym.sci.kun.nl/ pieterh/storage.html> which is also regularly posted on Usenet News. 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.

3.3.1 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 10ms, 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 15.6.

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.

Drives are usually described by the geometry or drive parameters which is the number of heads, sectors and cylinders, which is confused by translation schemes between physical and various logical geometries. This is a mine field which is described in painful details in many storage related FAQs. Read and weep.

3.3.2 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 650MB, transfer speeds are variable, depending on the drive but can exceed 1.5MB/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 100ms, 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 18GB on a single disk.

3.3.3 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 or better, Intel supports it through the Triton PCI chip set which is very popular these days. Many motherboards also include a SCSI interface chip made by NCR and that is connected directly to the PCI bus. Check what you have and what BIOS support you have with it.

3.4.1 MFM and RLL

Once upon a time this was the established technology, a time when 20MB 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.

3.4.2 ESDI

This technology became outdated almost before it got popular, so you are unlikely to come across it these days. Basically it was an attempt of increasing the upper limit of the old interfaces. You might get such a drive to work under Linux if it is compatible with the `ST506` standard.

3.4.3 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 504MB 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 8.3MB/s.

3.4.4 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.

3.4.5 ATAPI

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

3.4.6 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.

The old standard was 5MB/s and the newer fast-SCSI increased this to 10MB/s. Recently ultra-SCSI, also known as Fast-20, arrived with 20 MB/s transfer rates for an 8 bit wide bus.

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.

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.

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.

- Use as short cable as possible, but do not forget the 30cm minimum separation for ultra SCSI.
- Avoid long stubs between the cable and the drive, connect the plug on the cable directly to the drive without an extension.
- Use correct termination for SCSI devices and at the correct position: the end of the SCSI chain.
- 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.

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 132MB/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.

3.8 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. Some SCSI host adapters have several channels thereby multiplying the number of possible drives even further.

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

3.9 Future Development

The general trend is for faster and faster devices for every update in the specifications. ATA-3 is just out but does not define faster transfers, that could happen in ATA-4 which is under way. Quantum has already released DMA/33 and recent motherboard chip sets now supports this standard.

SCSI-3 is under way and will hopefully be released soon. Faster devices are already being announced, most recently an 80MB/s monster specification has been proposed. This is based around the ultra-2 standard (which used a 40MHz 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 264MB/s. The PCI transfer rate will in the future be increased from the current 33MHz to 66MHz, thereby increasing the limit to 528MB/s.

Another trend is for larger and larger drives. I hear it is possible to get 55GB on a single drive though this is rather expensive. Currently the optimum storage for your money is about 5GB but also this is continuously increasing. The introduction of DVD will in the near future have a big impact, with nearly 20GB 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...

3.10 Recommendations

My personal view is that EIDE 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 the point on 15.6.

4 Considerations

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.

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.

4.1.1 Swap

Speed

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

Size

Similar as for RAM. Quick and dirty algorithm: just as for tea: 16MB for the machine and 2MB for each user. Smallest kernel run in 1MB but is tight, use 4MB for general work and light applications, 8MB for X11 or GCC or 16MB 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.

Also remember to take into account the type of programs you use. Some programs that have large working sets, such as finite element modeling (FEM) 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.

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 `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.

4.1.2 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 enforcements and can grow without control on larger machines. Suggested: small home machine: 8MB, large home machine: 32MB, small server: 128MB, and large machines up to 500MB (The machine used by the author at work has 1100 users and a 300MB `/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.

Note

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.

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

4.1.3 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 100MB 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.

4.1.4 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* <`http://www.nyx.net/`> (which is a free Internet service with mail, news and WWW services) run successfully with a suggested limit of 100K per user and 300K as enforced maximum. Commercial ISPs offer typically about 5MB 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 8.5.1 and 8.5.3.

4.1.5 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 200MB 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 500MB 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.

4.1.6 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 70MB in their various `lib` directories. A rather complete Debian 1.2 installation can take as much as 250MB 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 4.1.5.

Files

Usually large with many of the order of 100 kB 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.

4.1.7 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 20MB 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.

4.1.8 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 50MB.

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:

4.2.1 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 15.6.

4.2.2 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.

4.2.3 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.

4.3 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.

4.3.1 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) or hardware, using a Linux compatible 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.

Currently the only supported hardware SCSI RAID controllers are the SmartCache I/III/IV and SmartRAID I/III/IV controller families from DPT. These controllers are supported by the EATA-DMA driver in the standard kernel. This company also has an informative *home page* `<http://www.dpt.com>` 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* `<http://www.i-connect.net/ mike/scsi>` and *DPT* `<http://www.i-connect.net/ mike/scsi/dpt>`.

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 RAID FAQ.

- 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*, *3* and *4* are not so common and is not covered here.
- RAID *5* offers excellent redundancy without wasteful duplication. It is fast in reading but not so fast for writing. It is normally recommended to use at least 3, preferrably more than 5 drives for this level.

There are also hybrids available based on RAID 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.

4.3.2 AFS, Veritas and Other Volume Management Systems

Although multiple partitions and disks have the advantage of making for more space and higher speed and reliability there is a significant snag: if for instance the `/tmp` partition is full you are in trouble even if the news spool is empty, as it is not easy to retransfer quotas across partitions. Volume management is a system that does just this and AFS and Veritas are two of the best known examples. Some also offer other file systems like log file systems and others optimised for reliability or speed. Note that Veritas is not available (yet) for Linux and it is not certain they can sell kernel modules without providing source for their proprietary code, this is just mentioned for information on what is out there. Still, you can check their *home page* `<http://www.veritas.com>` to see how such systems function.

Derek Atkins, of MIT, ported AFS to Linux and has also set up the Linux AFS mailing List for this which is open to the public. Requests to join the list should go to Request and finally bug reports should be directed to Bug Reports.

Important: as AFS uses encryption it is restricted software and cannot easily be exported from the US. AFS is now sold by Transarc and they have set up a www site. The directory structure there has been reorganized recently so I cannot give a more accurate URL than just the *Transarc Home Page* `<http://www.transarc.com>` which lands you in the root of the web site. There you can also find much general information as well as a FAQ.

Volume management is for the time being an area where Linux is lacking. Hot news: someone has just started a virtual partition system project that will reimplement many of the volume management functions found in IBM's AIX system.

4.3.3 Linux `md` Kernel Patch

There is however one kernel project that attempts to do some of this, md, which has been part of the kernel distributions since 1.3.69. Currently providing spanning and RAID it is still in early development and people are reporting varying degrees of success as well as total wipe out. Use with caution.

4.3.4 General File System Consideration

In the Linux world `ext2fs` is well established as a general purpose system. Still for some purposes others can be a better choice. News spools lend themselves to a log file based system whereas high reliability data might need other formats. This is a hotly debated topic and there are currently few choices available but work is underway. Log file systems also have the advantage of very fast file checking. Mail servers in the 100G class can suffer file checks taking several days before becoming operational after rebooting.

The `Minix` file system is the oldest one, used in some rescue disk systems but otherwise very little used these days. At one time the `Xiafs` was a strong contender to the standard for Linux but seems to have fallen behind these days.

Adam Richter from Yggdrasil posted recently 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* `<ftp://ftp.yggdrasil.com/private/adam>` where special patched versions of the kernel can be found. Hopefully this will be rolled into the mainstream kernel in the near future.

There is room for access control lists (ACL) and other unimplemented features in the existing `ext2fs`, stay tuned for future updates. There has been some talk about adding on the fly compression too.

There is also an encrypted file system available but again as this is under export control from the US, make sure you get it from a legal place.

File systems is an active field of academic and industrial research and development, the results of which are quite often freely available. Linux has in many cases been a development tool in such activities so you can expect a lot of continuous work in this field, stay tuned for the latest development.

4.3.5 CD-ROM File Systems

There has been a number of file systems available for use on CD-ROM systems and one of the earliest one was the *High Sierra* format, supposedly named after the hotel where the final agreement took place. This was the precursor to the *ISO 9660* format which is supported by Linux. Later there were the *Rock Ridge* extensions which added file system features such as long filenames, permissions and more.

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

However, once again Microsoft decided it should create another standard and their latest effort here is called *Joliet* and offers some internationalisation features. This is at the time of writing not yet available in the standard kernel releases but exists in beta versions. Hopefully this should soon work its way into the standard kernel.

In a recent Usenet News posting hpa@transmeta.com (H. Peter Anvin) writes the following the following interesting piece of trivia:

```
Actually, Joliet is a city outside Chicago; best known for being the
site of the prison where Elwood 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."
```

4.3.6 Compression

Disk 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 0.8.0a) 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.

Also there is the user file system (`userfs`) that allows FTP based file system and some compression (`arcfs`) plus fast prototyping and many other features.

Recent kernels feature the loop or loopback device which can be used to put a complete file system within a file. There are some possibilities for using this for making new file systems with compression, tarring, encryption etc.

Note that this device is unrelated to the network loopback device.

Very recently a compression package that extends `ext2fs` was announced. It is still under testing and will therefore mainly be of interest for kernel hackers but should soon gain stability for wider use.

4.3.7 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 a 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.

5 Other Operating System

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.

5.1 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
```

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

5.2 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.

The most important thing is the introduction of the new file system `FAT32` which is better suited to large drives. The snag is that there is very little support for this today, not even in NT 4.0 or many drive utility systems. A stable driver for Linux is coming soon but is not yet ready for prime time. Stay tuned for updates.

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.

Very recently someone started a project supporting `ext2fs` support for Win95 which you can read about at this *web site* `<http://www.globalxs.nl/home/p/pvs/>`.

5.3 OS/2

The only special note here is that you can get a file system driver for OS/2 that can read an `ext2fs` partition.

5.4 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* `<http://www.microsoft.com/kb/>` for more. (This affects Linux users, because Linux users have extra partitions)

5.5 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, 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 2.6 beta.

5.5.1 Sun OS 4

This is quite familiar to most Linux users. 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 manpages, 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.

5.5.2 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 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`. This is a kind of souped up RAM disk, and like ordinary RAM disks the contents is lost when the power goes. If space is scarce parts of the pseudo drive is swapped out, so in effect you store temporary files on the swap partition. Linux does not have such a file system; it has been discussed in the past but opinions were mixed. I would be interested in hearing comments on this.

6 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`

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 Beowolf Project* `<http://cesdis.gsfc.nasa.gov/linux/beowulf/beowulf.html>`
- *The Genoa Active Message Machine (GAMMA)* `<http://www.disi.unige.it/project/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* `<http://www.dolphinics.no/>` which is one of the main actors in this field, or you can have a look at *scizzl* `<http://www.scizzl.com/>`.

7 Mounting 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 released. Nobody knows when that will happen, and at the time of writing this a debate of near-religious qualities is taking place on the mailing list. 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      0
+-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       1
+-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*.

8 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.

At the end of this document there is an appendix with a few blank forms that you can fill in to help you decide and design your system. The following few paragraphs will refer to them.

8.1 Selection

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 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)

8.2 Mapping

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.

The tables in the appendices are designed to simplify the mapping process. 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. After that it is recommended to sort this list according to partition numbers into the table in appendix C and to use this when running the partitioning program (`fdisk` or `cfdisk`) and when doing the installation.

8.3 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
```

8.3.1 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.

8.3.2 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 spool and `.overview` files on separate drives.

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.

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 8.5.1 and 8.5.3.

8.4 Usage requirements

When you get a box of 10 or so CD-ROMs with a Linux distribution and the entire contents of the big FTP sites it can be tempting to install as much as your drives can take. Soon, however, one would find that this leaves little room to grow and that it is easy to bite over more than can be chewed, at least in polite company. Therefore I will make a few comments on a few points to keep in mind when you plan out your system. Comments here are actively sought.

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 50MB 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, 500MB 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 2-4GB 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 2GB 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.

8.5 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.

8.5.1 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.

8.5.2 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 50GB disk. Also memory is important for a dedicated FTP server, about 256MB RAM would be sufficient for a very big server, whereas smaller servers can get the job done well with 64MB RAM. Network connections would still be the most important factor.

8.5.3 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 2GB 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* `<http://www.nlanr.net/Squid>` and the one from Netscape.

8.5.4 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.

8.5.5 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 17GB. 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 12GB. Still others, that shall remain nameless, feel 2GB is sufficient to claim ISP status. In this case news expires so fast I feel the spelling IsP is barely justified.

8.5.6 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.

8.6 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.

8.7 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 point and partitions have to be defined before the system installation.

9 Implementation

Having done the layout you should now have a detailled 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.

9.1 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 `./kernel/Documentation/devices.tex`.

Partitions are labelled numerically for each drive `hda1`, `hda2` and so on.

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

9.2 Partitioning

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. Under DOS there are other choices, mainly the version of `fdisk` that is bundled with for

instance DOS, or `fips`. The latter has the unique advantage here that it can repartition a drive without necessarily damaging existing data, unlike all the other partitioning programs.

In order to get the most out of `fips` you should first defragment your drive. This way you can allocate more space to other partitions.

Nevertheless, it is important you do a full backup of all your valued data before partitioning.

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 a `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` and `ext2fs` are the ones you will need to know.

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.

9.3 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`.

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

9.4 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 manpage 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`.

9.5 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.

10 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` 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.

10.1 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.

10.2 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 making a defragmentation tool.

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, delete the original and then untar it back again.

10.3 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 5GB 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. 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 -)
```

11 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.

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.

11.1 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.

11.2 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.

12 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:

News groups

- *Storage* `<news:comp.arch.storage>`.
- *PC storage* `<news:comp.sys.ibm.pc.hardware.storage>`.
- *AFS* `<news:alt.filesystems.afs>`.
- *SCSI* `<news:comp.periphs.scsi>`.
- *Linux setup* `<news:comp.os.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* `<ftp://rtfm.mit.edu>`. The WWW versions can be browsed at *FAQ main archive WWW site* `<http://www.cis.ohio-state.edu/hypertext/faq/usenet/FAQ-List.html>`.

Some FAQs have their own home site, of particular interest here are *SCSI FAQ* `<http://www.paranoia.com/~filipg/HTML/LINK/F_SCSI.html>` and *comp.arch.storage FAQ* `<http://alumni.caltch.edu/~rdv/comp_arch_storage/FAQ-1.html>`.

Mailing lists

linux-raid, linux-scsi, linux-ext2fs ...

HOWTO

Bootdisk, Installation, , SCSI, UMSDOS ...

mini-HOWTO

Backup-With-MSDOS, Diskless, LILO, Linux+DOS+Win95+OS2, Linux+OS2+DOS, Linux+Win95, NFS-Root, Win95+Win+Linux, ZIP Drive ...

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`.

The kernel source is, of course, the ultimate documentation. In other words, *use the source, Luke*.

Much of the work here is based on the Filesystem Structure Standard (FSSTND). It has changed name to File Hierarchy Standard (FHS) and is less Linux specific. The maintainer has set up a *home page* `<http://www.pathname.com/fhs>` which tells you how to join the currently private mailing list, where the development takes place.

Many mailing lists are at `vger.rutgers.edu` but this is notoriously overloaded, so try to find a mirror. There are some lists mirrored at *The Redhat Home Page* `<http://www.redhat.com>`. Many lists are also accessible at *linuxhq* `<http://www.linuxhq.com/lnxlists>`, 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. The lists `linux-raid` and `linux-scsi` are of particular interest.

A few project pages:

- Mike Neuffer, the author of the DPT caching RAID controller drivers, has some interesting pages on *SCSI* `<http://www.i-connect.net/ mike/scsi>` and *DPT* `<http://www.i-connect.net/ mike/scsi/dpt>`.
- Software RAID 1 development information can be found at *RAID 1 development page* `<http://www.nuclecu.unam.mx/ miguel/raid>`.
- Mark D. Roth has information on *VPS* `<http://www.uiuc.edu/ph/www/roth>`
- A similar kind of project on an *Enhanced File System* `<http://www.virtual.net.au/ rjh/enh-fs.html>`
- People who are awaiting support for VFAT32 and Joliet could have a look at the *development page* `<http://bmrc.berkeley.edu/people/chaffee/index.html>` for a preview. Only minor details are missing before it comes into the kernel.

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

Remember you can also use the web search engines and that some, like

- *Altavista* `<http://www.altavista.digital.com>`
- *Excite* `<http://www.excite.com>`

- *Hotbot* <http://www.hotbot.com>

can also search usenet news.

Also remember that *Dejanews* <http://www.dejanews.com> is a dedicated news searcher that keeps a news spool from early 1995 and onwards.

For diagrams and information on all sorts of disk drives, controllers etc. both for current and discontinued lines *The Ref* <http://theref.c3d.rl.af.mil> is the site you need. There is a lot of useful information here, a real treasure trove. You can also download the database using *FTP* <ftp://theref.c3d.rl.af.mil/public>.

If you have to ask for help you are most likely to get help in the `comp.os.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.

13 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.

13.1 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` or `cfdisk`. 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.

Also I hope to get some information from DPT who make the only RAID controller supported by Linux so far. I have contacted them but have yet to hear from them.

There is some minor overlapping with the Linux Filesystem Structure Standard that I hope to integrate better soon, which will probably mean a big reworking of all the tables at the end of this document. When the new version is released there will be a substantial rewrite of some of the sections in this HOWTO but no release date has been announced yet.

When the new standard appear various details such as directory names, sizes and file placements will be changed.

I have made the assumption that the first partition starts at track 0 and that this track is the innermost track. That, however, is looking more and more like an unwarranted assumption, and not only because of the logical re-mapping that takes place. More on this when information becomes available.

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.

13.2 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?
- Leads to file system, volume management and other related software is welcome.
- 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 parallelicity.

13.3 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.

14 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, the section on 4.3.7 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 standard (soon released under the name File Hierarchy Standard) 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 the section on 4.3.1 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. The advantage is that this format is better suited to large drives. Unfortunately there is no stable driver for Linux out *yet* . A test version is out but not yet in the standard kernel.

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

  Until a stable version is available you can avoid this problem by installing Windows95 over an existing FAT16 partition, made for instance by an older installation of DOS. This forces the Windows95 to use FAT16 which *is* supported by Linux.

- Q: I cannot get the disk size and partition sizes to match, something is missing. What has happened? 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 th offending partition on a temporary mounting point. You might find it useful to have "spare" emergency mounting points ready made.

- 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* `<http://www.nyx.net>` which also gives you information on how to get your own free account.

15 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.

15.1 Combining `swap` and `/tmp`

Recently there have been discussions in the various linux related news groups about specialized file systems for temporary storage. This is partly inspired by the `tmpfs` on *BSD* and Solaris, as well as `swapfs` on the NeXT machines.

The rationale is that these are temporary storage that normally does not require much space, yet in normal systems you need to reserve a certain amount of space for these. Elementary statistical knowledge tells you (very simplified) that when you sum a number of variables the relative statistical uncertainty decreases. So combining `swap` and `/tmp` you do not need to reserve as much space as you otherwise would need.

These specialized file system is nothing more than a swappable RAM disk that are swapped out to disk when and only when space is limited, thus effectively putting temporary files on the swap partition.

There is, however, a snag. This scheme prevents you from getting parallel activity on `swap` and `/tmp` drives so under heavy activity the system takes a bigger performance hit. Put another way, you trade speed to get space. Interleaving across multiple drives reduces this somewhat.

15.2 Interleaved `swap` drives.

This is not striping across several drives, instead drives are accessed in a round robin fashion in order to spread the load in a crude fashion. In Linux you additionally have a priority parameter you can adjust for tuning your system, especially useful if your disks differs significantly in speed. Check `man 8 swapon` as well as `man 2 swapon` for more information.

15.3 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 64MB, 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.

15.4 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.

15.5 SCSI id numbers and names

Partitions are labeled in the order they are found, *not* depending on the SCSI id number. This means that if you add a drive with an id number inserted in the previous order of numbers, or change id number in any other way, the partition names will be messed up. This is important if you use removable media. In order to save yourself from some unpleasant experiences, you are recommended to use low numbers for fixed media and reserve the last number(s) for removable media drives.

Many have been bitten by this misfeature and there is a strong call for something to be done about it. Nobody knows how soon this will be fixed so in the meantime it is wise to take this into consideration when you design your system. For instance it may be a good idea to use the lowest SCSI id number for you root disk so that it has the least probability of being renumbered should one drive fail.

15.6 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 but 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.

15.7 Dejanews

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 *Dejanews* `<http://www.dejanews.com>` for more information.

What perhaps is less known, is that they use about 20 Linux SMP computers each of which uses the md module to manage between 4 and 24 Gig of disk space (over 150 Gig altogether) for this service. The system is continuously growing but at the time of writing they use mostly dual Pentium Pro 200MHz systems with 256M RAM.

A production 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 PCI SCSI adapters, usually two to a machine.

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.

15.8 File system structure

There are many file system structures in existence, differing with FSSTND (and soon FHS) to varying degree both in terms of philosophy, strategy and implementation. It is not possible to detail all here,

instead the interested reader should read the relevant manual page, `man hier` which is available on many platforms and implementations.

15.9 Track numbering and optimizing schemes

In the old days the file system used to take advantage of knowing the physical drive parameters in order to optimize transfers, for instance by endeavouring to keep a file within a single track if possible which saves track-to-track seek time. These days with logical drive parameters, drive cache and schemes to map out bad sectors, such optimizations become meaningless and might even cost more than it would gain. As most Linux installations use modern file systems these schemes are not used, however, some other operating systems have retained such schemes.

16 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:
```

17 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 4.3.7.

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

```
Directory         sda     sdb     sdc     hda     hdb     hdc     ___

SCSI ID         |  __   |  __   |  __   |

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:
```

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

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

Total capacity: |       |       |       |       |       |       |

Partition

1               |       |       |       |       |       |       |
2               :       :       :       :       :       :       :
3               |       |       |       |       |       |       |
4               :       :       :       :       :       :       :
5               |       |       |       |       |       |       |
6               :       :       :       :       :       :       :
7               |       |       |       |       |       |       |
8               :       :       :       :       :       :       :
```

```
9               |       |       |       |       |       |       |
10              :       :       :       :       :       :       :
11              |       |       |       |       |       |       |
12              :       :       :       :       :       :       :
13              |       |       |       |       |       |       |
14              :       :       :       :       :       :       :
15              |       |       |       |       |       |       |
16              :       :       :       :       :       :       :
```

19 Appendix D: Example: Multipurpose server

The following table is from the setup of a medium sized multipurpose server where I work. 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 is currently afforded by the `md` patch and a dedicated RAID controller was out of our reach.

20 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
```

21 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  |
```

22 Appendix G: Example: partition placement

This is just to sort the partition numbers in ascending order ready to input to fdisk or cfdisk.

```
        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 :
```

23 Appendix H: Example II

The following is an example of a server setup in an academic setting, and is contributed by `nakano@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 100M or so (since we must use Japanese fonts!).

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

24 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 system it would be better to swap `/tmp` and `/var/tmp` and then more `/var` to drive 1.