The Unix Hardware Buyer HOWTO

Eric Raymond

<esr@thyrsus.com>

Revision History

Revision 3.0	2004−02−21	Revised by: esr

Power−protection stuff moved to UPS HOWTO. DIMM memory is gone. Tape drive don't make sense any more. Lots of the theory from my "Ultimate Linux Box" articles now lives here.

Revision 2.4	2003−02−22	Revised by: esr

URL fixes.

Revision 2.3	2002−08−06	Revised by: esr

Buying at the low end isn't a lose anymore. I recommend Athlons. Nuked the section on video standards, EDID takes care of all that now. Also removed the section on older memory types. And keyboards, as the "ergonomic" ones all vanished along with the 1990s carpal−tunnel scare.

Revision 2.2	2002−08−05	Revised by: esr

New section on DVD drives.

Revision 2.1	2002−07−08	Revised by: esr

Corrected Kingston URL. Various small updates for the last year. This HOWTO is much more stable than it used to be.

Revision 2.0	2001−08−09	Revised by: esr

Major update. Revisions based on Ultimate Linux Box experience. Caches are on−chip now. DDS4 tape drives are here. 486 machines, CD caddies, and most non−DDS backup technologies are gone.

Revision 1.1	2001−06−13	Revised by: esr

Mid−2001 update.

Revision 1.0	2001−02−06	Revised by: esr

Initial revision; but see the history in the introduction.

This is your one−stop resource for information about how to buy and configure x86 hardware for cheap, powerful Unix systems.

Table of Contents

1. Introduction....1
1.1. Purpose of this document....1
1.2. New versions of this document....1
1.3. Feedback and corrections....2
1.4. Related resources....2

2. Overview of the Market....3

3. Buying the Basics....4
3.1. How To Pick Your Processor....4
3.2. Bus Wars....4
3.3. One Disk or Two?....4
3.4. Getting Down to Cases....5
3.5. Power Supplies and Fans....6
3.6. Motherboards....6
3.7. Memory....7
3.8. Monitor and Video....7
3.8.1. Selecting a Monitor....8
3.8.2. Buying a Video Card....10
3.9. Keyboards and Mice....10
3.10. Floppy Drives....10
3.11. CD−ROM Drives....11
3.12. DVD Drives....12
3.13. Sound Cards and Speakers....12
3.14. Backup devices....13
3.15. Modems....14
3.15.1. The Simple Answer....14
3.15.2. Overview of the Modem Market....14
3.15.3. Internal vs. External....14
3.15.4. Pitfalls to Avoid....15
3.15.5. What to Buy....15
3.15.6. Fax Modems....15
3.16. Printers....16
3.17. Power Protection....17
3.18. Radio Frequency Interference....17

4. What To Optimize....19
4.1. First, add more memory....19
4.2. Bus and Disk speeds....19
4.3. Disk Wars: IDE vs. SCSI....19
4.3.1. Enhanced IDE....20
4.3.2. Advantages of SCSI....21
4.3.3. SCSI Terminology....22
4.3.4. Avoiding Pitfalls....23
4.3.5. More Resources....25
4.4. Tuning Your I/O Subsystem....26

Table of Contents

5. But What If I'm Economizing?....30

6. Noise Control and Heat Dissipation....31

7. Special Considerations When Buying Laptops....33

8. How to Buy....34
8.1. When to Buy....34
8.2. Where to Buy....34
8.3. Computer Fairs....34
8.4. Mail Order....35
8.5. Computer Superstores....35
8.6. Other Buying Tips....36

9. Questions You Should Always Ask Your Vendor....37
9.1. Minimum Warranty Provisions....37
9.2. Documentation....37
9.3. A System Quality Checklist....37

10. Things to Check when Buying Mail−Order....38
10.1. Tricks and Traps in Mail−Order Warranties....38
10.2. Special Questions to Ask Mail−Order Vendors Before Buying....38
10.3. Payment Method....39
10.4. Which Clone Vendors to Talk To....39
10.4.1. Some pans....39
10.4.2. Some picks....39

11. After You Take Delivery....40

12. Software to go with your hardware....41

13. Other Resources on Building Linux PCs....42

1. Introduction

1.1. Purpose of this document

The purpose of this document is to give you the background information you need to be a savvy buyer of Intel hardware for running Unix. It is aimed especially at hackers and others with the technical skills and confidence to go to the mail−order channel, but contains plenty of useful advice for people buying store−front retail.

This document is maintained and periodically updated as a service to the net by Eric S. Raymond, who began it for the very best self−interested reason that he was in the market and didn't believe in plonking down several grand without doing his homework first (no, I don't get paid for this, though I have had a bunch of free software and hardware dumped on me as a result of it!). Corrections, updates, and all pertinent information are welcomed at esr@snark.thyrsus.com. The editorial «we reflects the generous contributions of many savvy Internetters.

If you email me questions that address gaps in the FAQ material, you will probably get a reply that says "Sorry, everything I know about this topic is in the HOWTO". If you find out the *answer* to such a question, please share it with me for the HOWTO, so everyone can benefit.

If you end up buying something based on information from this HOWTO, please do yourself and the net a favor; make a point of telling the vendor "The HOWTO sent me" or some equivalent. If we can show vendors that this HOWTO influences a lot of purchasing decisions, we get leverage to change some things that need changing.

Note that in December 1996 I published an introductory article on building and tuning Linux systems summarizing much of the material in this HOWTO. It's available here. In 2001 I published an article on building the Ultimate Linux Box.

This Buyer's Guide actually dates back to 1992, when it was known as the "PC−Clone Unix Hardware Buyer's Guide"; this was before Linux took over my world :−). Before that, portions of it were part of a Unix Buyer's Guide that I maintained back in the 1980s on USENET.

It may be a matter of historical interest that the page count of this guide peaked in mid−2001 and has been declining since. Video, sound, and other functions are migrating onto motherboards. Several bus types have disappeared, as have all the old−school backup technologies that couldn't scale up to match disk capacities, Spec sheets are getting simpler. Accordingly, there are parts that used to have whole sections to hemselves that I barely even write about anymore floppy disks and keyboards, for example, are utterly generic now,

In retrospect, the success of the ATX standard for motherboards in 1998−1999 was probably the turning point. The PC industry has become sufficiently commoditized that your choices are now getting simpler rather than more complicated. This is a Good Thing.

1.2. New versions of this document

New versions of the Unix Hardware Buyer HOWTO will be periodically be uploaded to various Linux WWW and FTP sites, including the LDP home page.

You can view the latest version of this on the World Wide Web via the URL http://www.tldp.org/HOWTO/Unix−Hardware−Buyer−HOWTO/.

1.3. Feedback and corrections

If you have questions or comments about this document, please feel free to mail Eric S. Raymond, at esr@thyrsus.com. I welcome any suggestions or criticisms. If you find a mistake with this document, please let me know so I can correct it in the next version. Thanks.

1.4. Related resources

You may also want to look at the read the Hardware−HOWTO. It lists hardware known to be compatible with Linux, and hardware known to be incompatible. I've also done a series of articles on The Ultimate Linux Box.

2. Overview of the Market

The central fact about clone hardware that conditions every aspect of buying it is this: more than anywhere else in the industry, de−facto hardware standards have created a commodity market with low entry barriers, lots of competitive pressure, and volume high enough to amortize a *lot* of development on the cheap.

The result is that this hardware gives you lots of bang−per−buck, and it's getting both cheaper and better all the time. Furthermore, margins are thin enough that vendors have to be lean, hungry, and *very* responsive to the market to survive. You can take advantage of this, but it does mean that much of the info in the rest of this document will be stale in three months and completely obsolete in six.

One good general piece of advice is that you should avoid the highest−end new−technology systems (those not yet shipping in volume). The problem with the high end is that it usually carries a hefty "prestige" price premium, and may be a bit less reliable on average because the technology hasn't been through a lot of test/improve cycles.

There used to be a real issue with low−end PCs as well, because there used to be a lot of dodgy crap PC components out there going into boxes made by vendors trying to save a few cents. That's not really a problem anymore. Market pressure has been very effective at raising reliability standards for even low−end components as the market has matured. It's actually hard to go wrong even buying at the bottom end of the market these days.

I put together the first version of this guide around 1992; Unix−capable systems are now five to ten times cheaper than they were then. At today's prices, building your own system from parts no longer makes much sense at all so this HOWTO is now more oriented towards helping you configure a whole system from a single vendor.

---

3. Buying the Basics

In this section, we cover things to look out for that are more or less independent of price−performance tradeoffs, part of your minimum system for running Unix.

Issues like your choice of disk, processor, and bus (where there is a strong tradeoff between price and capability) are covered in the section on What To Optimize.

3.1. How To Pick Your Processor

Right now (early 2004), the chips to consider for running Unix are the Pentium IVs and their clone equivalents from AMD or Cyrix or, if your budget will stand it, the AMD Opteron. The Pentium IV is something of a dog (very poor price−performance, actually slower than a III on some benchmarks), and the Itanium isn't out of the starting gate.

Brands don't matter much, so don't feel you need to pay Intel's premiums if you see an attractive Cyrix, AMD or other chip−clone system offered. In the last few years I've become a big fan of the AMD Athlon line faster, cheaper, and better−designed than Intel Pentiums.

To compare the performance of different Intel−based systems with each other and with machines from other manufacturers, you can take a look at the SPECmark Table at ftp://ftp.cdf.toronto.edu/pub/spectable. That document recommends (and I do too) that you read the SPEC FAQ at http://www.specbench.org/spec/specfaq.html to get background before browsing the table.

3.2. Bus Wars

The system bus is what ties all the parts of your machine together. This is an area in which progress has simplified your choices a lot. There used to be no fewer than four competing bus standards out there (ISA, EISA, VESA/VLB, PCI, and PCMCIA). Now there are effectively just two PCI−X for desktop/tower machines and PCMCIA for laptops; even PCI is now legacy technology.

I used to recommend dual−bus PCI/ISA boards, but no longer do. The on−board USB support and PS/2 mouse port now common on motherboards made the difference, it means you no longer need ISA even internally (and PCI cards are cheap these days). For your new desktop machine, go PCI−X only.

In the laptop market everything is PCMCIA. PCMCIA peripherals are about the size of credit cards (85x54mm) and vary in thickness between 5 and 10mm. They have the interesting feature that they can be hot−swapped (unplugged out and plugged in) while the computer is on. However, they are seldom seen in desktop machines. They require a special daemon to handle swapping, which is now standard under Linux.

3.3. One Disk or Two?

I always build with two disks one "system" disk and one "home" disk. There are two good reasons to do this that have nothing to do with the extra capacity. One of them is the performance advantage of being able to interleave commands to different physical spindles that we discussed above. The other is that I am quite a bit less likely to lose two disks at once than I am to trash a single one.

Let's suppose you have a fatal disk crash. If you have only one disk, goodbye Charlie. If you have two, maybe the crashed one was your system disk, in which case you can buy another and mess around with a new Linux installation knowing your personal files are safe. Or maybe it was your home disk; in that case, you can still run and do recovery stuff and basic Net communications until you can buy another home disk and restore it from backups (you *did* keep backups, right?).

Your performance−tuning choice is IDE versus SCSI. We'll have more to say about that in Section 4.

3.4. Getting Down to Cases

I used to say that cases are just bent metal, and that it doesn't much matter who makes those. Unfortunately, this isn't true any more. Processors run so hot these days that fans and airflow are a serious concern. They need to be well designed for proper airflow throughout.

Look for the following quality features:

- Aluminum rather than steel. It's lighter and conducts heat better.
- Unobstructed air intake with at least one fan each (in addition to the power supply and processor fans)
- No sharp metal edges. You doon't want to shred your hands when you're tinkering with things.
- There shouldn't be any hot spots (poor air flow).
- Sturdy card clips. Some poorly−designed cases allow cards to wiggle out of their slots under normal vibration.
- Effective and easy to use mechanisms for attaching hard drives, CD−ROM, CD−R/W, DVDs, etc.

If you're fussy about RFI (Radio−Frequency Interference), it's worth finding out whether the plastic parts of the case have conductive coating on the inside; that will cut down emissions significantly, but a few cheap cases omit it.

Should you buy a desktop or tower case? Our advice is go with tower unless you're building a no−expansions personal system and expect to be using the floppies a lot. Many vendors charge nothing extra for a tower case, and the cost difference will be trivial even if they do. What you get for that is less desktop clutter, more and bigger bays for expansion, and often (perhaps most importantly) a beefed−up power−supply and fan. Putting the box and its fan under a table is good for maybe 5db off the effective noise level, too. Airflow is also an issue; if the peripheral bays are less cramped, you get better cooling. Be prepared to buy extension cables for your keyboard and monitor, though; vendors almost never include enough flex.

The airflow thing is a good argument for a full− or mid−tower rather than the baby tower cases some vendors offer. However, smaller towers are getting more attractive as boards and devices shrink and more functions migrate onto the motherboard. A state of the art system, with all 3" disks, 300W power supply, half−size motherboard, on−board IDE and 64meg of RAM sockets, and half−sized expansion cards, will fit into a baby or midsized tower with ample room for expansion; and the whole thing will fit under a desk and make less noise than a classic tower.

For users with really heavy expansibility requirements, rackmount PC cases do exist (ask prospective vendors). Typically a rackmount case will have pretty much the same functionality as an ordinary PC case. But, you can then buy drive racks (complete with power supply), etc. to expand into. Also, you can buy passive backplanes with up to 20 or so slots. You can either put a CPU card in one of the slots, or connect it to an ordinary motherboard through one of the slots.

3.5. Power Supplies and Fans

A lot of people treat power supplies as a commodity, so many interchangeable silver bricks. We know better cheap power supplies go bad, and when they go bad they have a nasty habit of taking out the delicate electronics they're feeding. Also, the power supply tends to be the noisiest component in your system.

Give preference to supplies with a Underwriter's Laboratories rating. There's some controversy over optimum wattage level. On the one hand, you want enough wattage for expansion. On the other, big supplies are noisier, and if you draw too little current for the rating the delivered voltage can become unstable. And the expected wattage load from peripherals is dropping steadily. On the other hand, processors and their cooling fans eat a lot more power than they used to.

The choice is generally between 200W and 300W. After some years of deprecating 300W−and−up supplies as overkill, I'm now persuaded it's time to go back to them; a modern processor can consume 50−75W by itself, and for the newer dual−processor board the power supply needs to be rated 450W or up.

About that annoying fan noise, ask if the power−supply fan on a target system has a variable speed motor with thermostatic control; this will cut down on noise tremendously. However, be aware that a thermostatic sensor basically measures the temperature *at the sensor* (typically within the power supply box) and makes sure there is enough airflow to keep the power supply from overheating. The sensor does not know a thing about the temperature in certain hot spots likely to develop in a PC case (CPU, between SIMMs, between drives mounted in vertically adjacent bays).

This can be a problem, because in garden variety tower cases there often isn't enough airflow to cool all components effectively even if a single fan is going at full speed. This is especially true if your computer has lots of add−on cards or hard disks (not much airflow between cards or between drives). Note that the fan in the power supply was basically designed to cool the power supply, not the components in the case. Not providing additional fans is a sign of cheapness. On tower PCs with "expensive" engineering (e.g. HP Vectra, Compaq) one will find one to two extra fans besides the one in the power supply.

So the bottom line is, use thermostatic controls if you can to cut noise. But if you want high reliability, use two or more fans. Modern designs normally also have a small auxilliary fan mounted right over the chip.

The noise produced by a fan is not just a function of the speed with which it turns. It also depends on the nature of the airflow produced by the fan blades and the bearings of the rotor. If the blades causes lots of turbulent airflow, the fan produces lots of noise. One brand of fans that is much more silent than most others even if going at full throttle is Papst.

3.6. Motherboards

Provided you exercise a little prudence and stay out of the price basement, motherboards and BIOS chips don't vary much in quality. There are only six or so major brands of motherboard inside all those cases and they're pretty much interchangeable; brand premiums are low to nonexistent and cost is strictly tied to maximum speed and bus type. There are only four major brands of BIOS chip (AMI, Phoenix, Mylex, Award) and not much to choose between 'em but the look of the self−test screens (even the "name" vendors use lightly customized versions of these). One advantage Unix buyers have is that Unixes are built not to rely on the BIOS code (because it can't be used in protected mode without more pain than than it's worth). If your BIOS will boot properly, you're usually going to be OK.

Some good features to look for in a motherboard include:

- Gold−plated contacts in the expansion slots and RAM sockets. Base−metal contacts tend to grow an oxidation layer which can cause intermittent connection faults that look like bad RAM chips or boards. (This is why, if your hardware starts flaking out, one of the first things to do is jiggle or remove the boards and reseat them, and press down on the RAM chips to reseat them as well this may break up the oxidation layer. If this doesn't work, rubbing what contacts you can reach with a soft eraser is a good fast way to remove the oxidation film. Beware, some hard erasers, including many pencil erasers, can strip off the plating, too!)
- The board should be speed−rated as high as your processor, of course. It's good if it's rated higher, so upgrade to a faster processor is just a matter of dropping in the chip and a new crystal.
- Voltage, temperature and fan speed monitoring hardware. This is now common on motherboards based on recent iterations of the Intel support chips, especially those designed for server use. Linux supports drivers that can read this hardware, and monitoring can help you spot incipient board failures.

If you're changing a motherboard, see the Installing a Motherboard page first. This one even has a Linux note.

The dominant form factor now is still ATX, but there is a complete redesign called BTX coming down the pike very shortly (as in, later in 2004).

3.7. Memory

All current PC designs include a cacheing memory controller and some fast on−chip cache that combine to produce higher effective speeds. Judging the cache design used to be one of the trickiest parts of evaluating a motherboard, but that stuff is all baked into the processor itself now. Leading−edge designs like the AMD Opteron even implement the memory controller inside the processor itself, removing another source of latency and design variations.

For current motherboards with 133MHz Memory Bus support, PC133 should be used instead of PC100; it gives 33% greater memory bandwidth at very little additional cost. DDR−SDRAM and RDRAM are faster memory types that retrieve data in chunks and give you faster throughput. So−called `PC266' memory is designed for motherboards that transfer at 133 but double the width of the front−side bus connecting processor and memory.

As the throughput of processor−to−memory buses rises, memory latency (bus cycles required for the first fetch in a chunk) is becoming a more important statistic. Lower numbers are better.

For more technical stuff on memory architectures, see The Ultimate Memory Guide maintained by Kingston Technologies.

3.8. Monitor and Video

The more pixels you can afford to put on screen, the better. There are factors other than resolution and price that have a strong liveability impact, however. A major one is the sheer amount of space big monitors take up. It's not a dream system if the display tube won't fit on your desk!

Today's CRT monitors and LCD flatscrees both top out at 2048x1536 resolution (with a few special and extremely expensive exceptions). As LCDs fall in price they are very close to parity with CRTs, and look like

an increasingly good choice. Their only serious drawback for most uses is slow response time twitch games and video have a tendency to blur just a bit.

Next, buy your card. The major issue here is matching the card to the capacity of your monitor you don't want to buy a card and find it can't drive your monitor at its maximum capability. If (unlike us) you're economizing, you also don't want to pay for more card than your monitor can use.

So once you've specified your monitor, find a video card with a maximum video bandwidth equal to or just slightly higher than the monitor's. That's how you know your video system is properly balanced, with a minimum of wasted capacity.

3.8.1. Selecting a Monitor

I used to carry a lot of material on different video standards, interlacing, and flicker. That stuff is all obsolete now. Nobody makes anything less capable than SVGA 1024x768 at 72 refreshes per second any more, and all new monitors auto−negotiate with your video card to settle on the resolutions they can support. An abbrebiation you may see is "UXGA"; that means 1600x1200.

The only situation in which you might have to do manual tuning is when the monitor's resolution is higher than any of the standard mode line X knows how to support. In February 2004 that's above 1920x1440. If you find youself in this situation, see XFree86 Video Timings HOWTO .

3.8.1.1. What To Look For On The Spec Sheet.

- Dot pitch of 0.28 or smaller on a 12"−15" monitor; 0.30 is acceptable on larger ones, especially 19" to 21" screens (but look extra hard at 0.25 21−inchers like the Viewsonic 21PS or Nokia 445X). Dot pitch is the physical resolution of the screen's phosphor mask. Larger dot pitches mean that small fonts and graphic details will be fuzzy.
- 72Hz or better vertical scan frequency, to cut flicker.
- Does it have a tilt−and−swivel base? Adequate controls, including both horizontal and vertical size and horizontal and vertical centering? A linearity control, a trapezoidal control, and a color−temperature control are all pluses; the last is particularly important if you compose graphics on screen for hardcopy from a printer.

If you can, buy your monitor from someplace that will let you see the same monitor (the very unit you will walk out the door with, not a different or `demo' unit of the same model) that will be on your system. There's a lot of quality variation (even in "premium" monitor brands) even among monitors of the same make and model.

Another good reason to see before you buy, and carry it home yourself, is that a lot of monitors are vulnerable to bumps. The yoke can get twisted, producing a disconcerting tilt in the screen image.

The Caveat Emptor guide has a good section on evaluating monitor specifications. And there's a database of monitor specs at The Big Old Monitor List.

3.8.1.2. Eric Buys A Big Monitor: Smart Shopping Tips

In early 1996 the good folks at O'Reilly Associates dropped several $1000 checks on me in relatively quick succession (payment for fast−turnaround technical reviews). I decided to use the money to treat myself to a really good monitor.

This page tells you how I did it. Specific specs and pricing information will date quickly, but the method should still be good years from now.

My existing monitor wasn't bad a 17−inch Swan 617 that I could drive at a bit above 1024x768. Still, I yearned for more real estate especially vertical real estate, so I could view full PostScript pages using a legible font.

This brings us to our first prescription: *be clear about what you want*. It's easy, and very expensive, to buy more monitor than you'll really use.

I knew I wanted something in the 19−to−21−inch range, with 1280x1024 or higher resolution. I knew this would probably cost me about $2000, and could afford it. I knew I *didn't* need one of the monster projection monitors further upmarket, with screen sizes 24" and up. These will typically cost you $4K or so and are too big for desktop use anyway.

I also knew I didn't need one of the special true−color monitors designed for photo composition, making print separations, and so forth. These creatures (always Trinitrons) have better, denser color than conventional tubes but at a hefty price premium (and usually at some cost in available resolution). If all you're going to do most of the time is 16 or 256−color X screens, you don't need this capability.

Once you've settled on what you need, *gather comparative data*. It was 1996, so I started out by making phone calls to manufacturer 800 numbers. Then I discovered that almost all the manufacturers had Web sites, with technical specs for their monitors on them. Today, you'd go to the Web first.

(This space used to include detailed technical data on what I found " model numbers, resolutions, reviews of manufacturer websites, etc." but I've removed it because it's all five years out of data now.)

This wasn't at all a hard call. The ViewSonic 21PS and Nokia 445X stood out from the pack immediately; their combination of high bandwidth with a 21−inch screen size and ultra−fine .25 dot pitch promised better performance than the general run of .28−pitch monitors.

Nor was the choice between the two very hard. ViewSonic's 21PS is $600 less expensive than Nokia's 445X for very similar performance. And, other things being equal, I'd rather buy a monitor from a specialist monitor manufacturer than a general consumer electronics outfit best known for its cellular phones.

So I determined to order a ViewSonic 21PS.

This left me with a second problem. My ATI Mach 32 can't drive a monitor at higher than 1280x1024 resolution and 94MHz bandwidth. So it wouldn't be able to drive the 21PS at 1600x1200. I wound up buying a Mach 64.

The combination worked wonderfully (two years later I discovered that VA Linux Systems bought the same monitor for its high−end systems). The only problem I have with it is that monitor is *way* bright even dialed down to its dimmest setting. You'll need a strong light in the room where you install it. Also, be aware that the only really convenient way to move one of these monster monitors is with a forklift!

Eight years later, in 2004, all these shopping tips are good and the high−end Viewsonics are actually still among the best monitors around.

3.8.2. Buying a Video Card

Video controllers translate byte values deposited in their video memory by your GUI (usually an X server under Linux) into an analog RGB signal which drives your monitor. The simplest kinds treat their video memory as one big frame buffer, requiring the CPU to do all dot−painting. More sophisticated "accelerated" cards offer operations such as BitBlt so your X server can hack the video memory algorithmically. These days almost all cards even at the low end actually have some acceleration features.

Cards are rated by the maximum number of analog signal changes they can produce per second (video bandwidth). Video bandwidth can be used to buy varying combinations of screen resolution and refresh speed, depending on your monitor's capabilities.

Another important variable of video cards is the size of their on−board video RAM. Increased memory lets you run more colors at higher resolutions. 4MB of video memory, which can drive 24−bit or "true" color (16 million colors) at 1024x768, is pretty much the minimum nowadays; most cards have more.

The card's video RAM size has no effect on its speed. What does affect speed is the *type* of memory on board. VRAM (Video Random Access Memory) is fast but more expensive; it features a dual−ported design allowing two devices (the CRT controller and the CPU) to access the memory at the same time. DRAM (Dynamic Random Access Memory) is is similar to the RAM used in main memories. It is cheaper, more common, and slower (because the CRT controller and the CPU must take turns accessing the video buffer).

Effectively all cards made today use AGP, a special high−speed attachment slot, and even low−end motherboards support it. That's if your video isn't integrated right onto the motherboard, an increasing trend.

3.9. Keyboards and Mice

Keyboards are mostly generic nowadays. One useful piece of advice is to not buy any desktop with "Internet" buttons on it; this is a sure sign of a PC that's an overpriced glitzy toy. The coming thing is USB keyboards; by the end of 2004 new machines probably won't have traditional keyboard ports any more. Modern open−source Unixes handle these just fine.

Mice and trackballs used to be simple; then, thanks to Microsoft, they got complicated. Now they're simple again; all ATX motherboards have a mouse port, and all new mice are made to plug into it. They're going to get simpler; dedicated mouse ports are on the way out, and USB mice will soon dominate. XFree86 autodetects your mouse when it starts up, so configuration is not a big deal any more.

Beware that most clone vendors, being DOS oriented, bundle two−button mice. Thus, you may have to buy your own three−button mouse. Ignore the adspeak about dpi and pick a mouse or trackball that feels good to your hand.

Your humble editor really, really likes the Logitech TrackMarble, an optical trackball that eliminates the chronic roller−fouling problems of the older TrackMan. They're well−supported by XFree86 (type MouseMan), so any Linux or BSD will accept them.

3.10. Floppy Drives

There's not much to be said about floppy drives. They're cheap, they're generic, and the rise of CD−ROM drives as a cheap distribution medium has made them much less important than formerly. You only ever see

the 3.5−inch hard−shell floppies with 1.44MB capacity anymore.

You'll probably never use floppies for anything but first boot of a new operating system. Bootable CD−ROMs, standard of most PCs these days, eliminate even that use. So go ahead and settle for cheap Mitsumi and Teac floppy drives. There are no premium floppy drives anymore. Nobody bothers.

It's possible your system won't even include one. No loss in 2004.

3.11. CD−ROM Drives

You'll need a CD−ROM or DVD−ROM drive (you'll almost certainly be installing your Linux from it!). You have a SCSI system, so get a SCSI CD−ROM. That's pretty much the end of spec, as there are only a few models of SCSI DVD−ROM and SCSI CD−ROMs are a very generic item. The only significant price driver is their speed 8x, 10x, or up (it's hard to find lower speeds anymore). Note that however high a read speed the brochure cites, these drives basically don't get any faster in practice above 12x. Big numbers like 40x are theoretical what you'd get on an uninterrupted sequential read of a perfectly balanced, perfectly clean disk.

Standard CD−ROMs hold about 650 megabytes of read−only data in a format called ISO−9660 (formerly "High Sierra"). All current Unixes now support these devices. In fact, most Unix and Linux software is now distributed on ISO−9660 CD−ROM, a cheaper and better method than the QIC tapes we used to use.

Modern CD−ROM drives may be driven through either SCSI or enhanced IDE (ATAPI). Some used to come with dedidcated interface cards, but those are obsolete. A few external CD−ROMs come with a parallel−port interface. Avoid these; they tend to have very slow transfer rates.

Any CD−ROM you buy should be at least a "double−spin" drive meeting the MPC2 (Multimedia PC) standard of a 300K/sec transfer rate when reading. digital data. The older single−speed drives, which only supported the 150K/sec rate Red Book standard for audio CDs, are obsolete. The lowest speed you can buy these days is 4X (600K/sec). 6X, 8X, 10X, 12X, 24X, 32X, 40X, and 56X are available. Note that however high a read speed the brochure cites, these drives basically don't get any faster in practice above 12x. Big numbers like 40x are theoretical what you'd get on an uninterrupted sequential read of a perfectly balanced, perfectly clean disk.

CD−ROM access times about 280ms for high−end double−speed drives (to put this in perspective, it's about 30 times slower than a typical 9ms hard disk, but considerably faster than a tape). Accordingly, modern 32X drives are about half the speed of a hard drive.

Most CD−ROMS will include a headphone jack so you can play audio CDs on them. Better−quality ones will also include two RCA jacks for use with speakers. Another feature to look for is a drive door or seal that protects the drive head from dust.

CD−ROM formats are still an area of some confusion. A slight enhancement of the original "High Sierra" CD−ROM filesystem format (designed for use with DOS, and limited to DOS's 8+3 file−naming convention) has been standardized as ISO−9660.

There is a de−facto Unix standard called Rock Ridge pioneered by the Sun User's Group shareware CD−ROMs. This is a way of putting an extra layer of indirection on an ISO−9660 layout that preserves Unix's long dual−case filenames. Some Unixes (notably Linux, netBSD, freeBSD and BSD/OS) can mount Rock Ridge filesystems.

More much more detail on CD−ROMs, CD−ROM standards and how to buy drives is available in the alt.cdrom FAQ, available for FTP as cdrom.com:/cdrom/faq. It is also archived in the news.answers tree at rtfm. This FAQ includes comparison tables of numerous drive types, CD−ROM sources, and ordering information.

3.12. DVD Drives

(Most of this section courtesy of James Turinsky.)

Most drives manufactured after January 1st 2000, and some drives prior to that have come installed with something called RPC2.

When a drive is RPC2, it means that it stores the Region code physically within the drive. This means that nothing you do on the software level will be of any help (including using DVD Genie or formatting your hard drive).

The only means of bypassing this Regional Protection Scheme is by using a firmware patch. A firmware patch is a special piece of software written for a specific DVD Drive model. It will only works on that specific model.

The original role of the firmware patch was to fix minor flaws in the drive logic (a piece of programmable software within the drive, also referred to as a "Firmware"). This logic also controls the Region Checks, so some inventive programmers have modified these firmware patches to remove the portion of logic that does the Regional checks, thus making the drive region−free (RPC1 stands for region free).

However, since a firmware patch is specific to one drive model, it's impossible to make a global fix for all drives on the hardware level. This limitation requires programmers to modify each firmware patch that comes out for various drives. And here lies the problem. Modifying a firmware is not a simple task, it requires good understanding of the hardware and some advanced programming skills. To top this off, there are quite a few DVD Drive models out there, and not as many programmers with the skill that can access these drives. So in reality, not all drives have firmware patches that can make the drive region free.

So some foresight is required when buying a new DVD Drive. You should check if someone already released a patched firmware for the model you wish to buy.

For more, see the Firmware Patches site.

3.13. Sound Cards and Speakers

Look for the following features as a minimum in your sound card:

- 16−bit sampling (for 65536 dynamic levels rather than 256).
- Mono and stereo support.
- Full−duplex mode.
- Sampling rates ranging fron 8K/sec (voice−quality) through 11KHz (AM−radio quality), 22KHz (FM−radio quality) and standard audio (44.1KHz).
- MIDI interface via a standard 15−pin D−shell connector.
- RCA output jacks for headphones or speakers.
- A microphone jack for sound input.

Older and cheaper cards use FM synthesis. This synthesis uses a combination of sine waves to imitate the sounds of the different instruments. The result is like the sound tracks of most computer games sold a few years ago; imitation music with an arcade−like sound.

The method used by most modern sound boards is called wave table synthesis. In this method, digitized samples of actual instrument sounds are used as templates for the tones generated by the MIDI commands. Wave table cards vary in the quantity and quality of samples; one figure of merit often quoted is the wave table ROM size (often 4MB or 8MB). Also some boards have wavetable RAM that can store samples loaded from a disk.

Soundcards with DSP (Digital Signal Processing) can perform synthesis effects on board, relieving the CPU for other tasks. Some DSP chips are even software−programmable. Some high−end cards even include 3D sound effects. Whether the system used is SRS (Sound Retrieval System), Q−Sound, or Spatializer, it is designed to improve the perceived stereo effect of your speakers. These 3D effects work by delaying the timing of certain portions of the audio signal so that different frequencies hit your ear at slightly different times. The downside is that some of the cards equipped with 3D sound add a noticeable amount of noise to the card's output.

If you play a lot of computer games, you'll need to pay attention to compatibility. DOS games are written almost exclusively for the Creative Labs specification; you will need a card that is 100% Sound Blaster compatible. Many vendors do not license the Creative Labs specification but claim that their cards are 100% game compatible. This means that the sound will work, but not all sounds that you hear will be the ones that the game programmers intended. If you play many DOS games, it would be best to buy a Sound Blaster and save yourself a migraine.

Lastly, try to avoid sound cards with built−in amplifiers that are more powerful than 4 watts/channel. Sound cards that have more powerful amplifiers are said to have the problem of adding noise to the card's output. Use powered speakers with a 4 watt/channel card to solve this problem. Most cards are equipped with 4 watts/channel anyway. Wavetable cards are so inexpensive these days that it's almost worth their additional cost over a regular FM synthesis card. If you decide to settle for an FM card, make sure that there is a daughterboard made for the card that will let you upgrade to wavetable synthesis. In some cases, however, the wavetable card is cheaper than buying an FM card and then deciding that you want the wavetable upgrade. If you do decide on the wavetable as your card of choice, PC Magazine rated the best MIDI wavetables (MIDI being the most important feature in my opinion) the Media Vision Premium 3−D, Media Vision Pro 3−D, Creative Labs Sound Blaster AWE32, and the Turtle Beach Monterey (although there are value editions of the Sound Blaster 32 that have fewer ROM instrument samples but maintain the superior MIDI wavetable synthesis).

In speakers, look for a magnetically−shielded enclosure with volume, bass and treble controls. Some speakers run off the card's 4−watt signal; others are "self−powered", using batteries or a separate power supply. Your major buying choice is which one of these options to pursue.

One final, important tip: that audio cable from your CD−ROM back to the sound card is used only when you play audio CD−ROMs through your speakers. Software−generated sound goes through the system bus, so you can play "Doom" with sound even if your sound board won't accept the audio cable connector.

3.14. Backup devices

It's good to be able to make backups that you can separate from your system and store off−site in case of disaster. Two years ago, tape drives still seemed like a good idea for personal systems, but I found I seldom

used mine. Today, tape drive with high enough capacity to image today's huge hard disks are too expensive to make sense any more.

For the money you'd spend on a high−capacity tape drive (over $1000) it makes more sense to buy a laptop and a pile of CD−RW media. Sit the laptop on your house Ethernet when you're not traveling, and back up the main machine to it every day, or oftener. Between the efficiency of rsync and the speed of 100−megabit Ethernet, this will be a lot faster than making a tape. Every once in a while, burn a set of backup CD−ROMs.

3.15. Modems

This section will give you a thumbnail sketch of the modem types available out there, one tuned for the typical Unix installation's needs.

3.15.1. The Simple Answer

The modem market has stabilized, with a clear leader at a reasonable price. Demand for modems is dropping as more and more people get broadband Internet through DSL and cable. If you need a modem and can spend $94, get a U.S. Robotics V.92 external. You can then know that you've got the best and skip the rest of this section.

3.15.2. Overview of the Modem Market

The modem market is like consumer electronics (and unlike the computer market as a whole) in that price is a very poor predictor of performance. For ordinary file transfers, some $50 modems are better than some $150 modems. Paying top dollar mainly buys you better tolerance of poor connections and better performance at heavy−duty bi−directional transfers (such as you would generate, for exmaple, using SLIP or PPP over a leased line to an Internet provider).

In today's market, the typical modem does a nominal 56kbps V.90 and V.92 plus V.29 or V.17 fax transmission and reception (over plain old phone lines you won't get more than 53K of that). You don't see much in the way of slow/cheap to fast/expensive product ranges within a single brand, because competition is fierce and for many modem board designs (those featuring DSP (Digital Signal Processor) chips run by a program in ROM) adding a new protocol is basically a software change.

3.15.3. Internal vs. External

Most modems come in two packagings: internal, designed to fit in a PC card slot, and external, with its own case, power supply, and front−panel lights. Typically you'll pay $20 to $30 more for an external modem than you will for the internal equivalent. You'll also need a serial port to connect your external modem to.

Pay that premium being able to see the blinkenlights on the external ones will help you understand and recover from pathological situations. For example, if your Unix system is prone to "screaming−tty" syndrome, you'll quickly learn to recognize the pattern of flickers that goes with it. Punch the hangup/reset button on an external modem and you're done whereas with an internal modem, you have to go root and flounder around killing processes and maybe cold−boot the machine just to reset the card.

See Rick's Rants for extended discussion of this point.

3.15.4. Pitfalls to Avoid

If the abbreviation "RPI" occurs anywhere on the box, don't even consider buying the modem. RPI (Rockwell Protocol Interface) is a proprietary "standard" that allows modem makers to save a few bucks at your expense by using a cheap−jack Rockwell chipset that doesn't do error correction. Instead, it hands the job off to a modem driver which (on a Unix machine) you will not have.

Also avoid anything called a "Windows Modem" or "WinModem", "HCF", or "HSP"; these lobotomized pieces of crap require a Windows DLL to run. They will eat up to 25% of your processor clocks during transfers, and hog high−priority interrupts (causing your machine to stall under Windows even if your processor still has spare cycles).

Multi−user Unix eats enough processor clocks that you want to be sure of good hardware buffering in your UART that is, enough of it to avoid losing characters between modem and PC if the OS is a bit slow responding to an interrupt (V.42bis in hardware won't detect this!). This means you want a 16550A or equivalent UART. If you're using an external modem, this is an issue about your serial−port board(s). If you're using an internal modem, the UART is on the modem card itself. So, when buying internal modems, *ask* what the UART type is. If the vendor says 16540, lose them.

Many fax modems come with bundled MS−DOS fax software that is at best useless under Unix, and at worst a software kluge to cover inadequate hardware. Avoid these bundles and buy a bare modem it's cheaper, and lowers the likelihood that something vital to your communications needs has been left out of the hardware.

Avoid "Class 1" and "Class 2" modems. Look for "Class 2.0" for the full EIA−standard command set.

3.15.5. What to Buy

What you want, these days, is a V.92 modem. V.92, formally, is the ITU−T recommendation for asymmetric data signalling rates of up to 56Kbps in the direction of a digitally connected server to a capable client, and up to 33.6Kbps in the direction of the client to the server.

The technology is based on eliminating restrictions imposed by the conversion of analog signals to digital form in the downstream data path (server −> client). Data flow in the server to client direction does not occur in the form of a modulated carrier, it is instead sent as binary numbers representative of 256 possible voltage levels. The reason for the asymmetrical send/receive rates is because in the direction from the client to the server it is not possible to use a digital coding scheme and make it work as well as V.34 does, thus V.34 is used instead. It isn't possible because the telco's line card has a codec that is a much better digital level changer for the transmit direction than it is for the receive direction. The codec used in the customer's modem is, in that respect, somewhat more sophisticated and was designed to work as a fairly good level changer in the receive direction (which the telco's codec was not designed to do).

Note: Achievable bit rates are limited to less than 56kbps in the United States by FCC regulations that limit power input to the network.

3.15.6. Fax Modems

Fax capability is included with most all modems these days; it's cheap for manufacturers, being basically a pure software add−on. The CCITT also sets fax protocol standards. Terms to know:

V.29

CCITT standard for Group III fax encoding at 9600bps

V.17
CCITT standard for Group III fax encoding at 14400bps

There's a separate series of standards for software control of fax modems over the serial line maintained by the Electronics Industry Association and friends. These are:

Class 1 base EIA standard for fax control as extensions to the Hayes AT command set.

Class 2.0 enhanced EIA standard including compression, error correction, station ID and other features.

Class 2 marketroidian for anything between Class 1 and Class 2.0. Different "Class 2" modems implement different draft subsets of the 2.0 standard, so "Class 2" fax software won't necessarily drive any given "Class 2" modem.

There's also a proprietary Intel "standard" called CAS, Communicating Applications Specification. Ignore it; only Intel products support it.

The GNU toolset includes an open−source fax transmission and reception toolset, Netfax. Look for it at prep.ai.mit.edu:pub/gnu/fax−*. It says it requires a modem conforming to the "Class 2" control standard, but you'd be safest getting a 2.0−conformant modem for reasons explained above. Netfax also requires GNU Ghostscript to do Postscript handling for it.

3.16. Printers

There really isn't all that much to be said about printers; the market is thoroughly commoditized and printer capabilities pretty much independent of the rest of your hardware. The PC−clone magazines will tell you what you need to know about print quality, speed, features, etc. The business users they feed on are obsessed with all these things.

(There used to be a problem with "GDI printers" and "WinPrinters" that only worked with Windows they required special drivers that took over your CPU to do image processing, These were such a bad idea that they have basically disappeared off the market.)

Most popular printers are supported by GhostScript, and so it's easy to make them do PostScript. If you're buying any letter−quality printer (laser or ink−jet), check to see if it's on GhostScript's supported device list otherwise you'll have to pay a premium for Postscript capability! Postscript is still high−end in the MS−DOS market, but it's ubiquitous in the Unix world.

Warning, however: if you're using ghostscript on a non−Postscript printer, printspeed will be slow, especially with a serial printer. A bitmapped 600 dpi page has a *lot* of pixels on it. At today's prices, paying the small premium for Postscript capability makes sense.

If you're buying a printer for home, an inkjet is a good choice because it doesn't use gobs of power and you won't have the toner/ozone/noise/etc mess that you do with a laser. If all you want is plain−ASCII, dot−matrix is cheaper to buy and run if you can find one. Inexpensive ink−jets and lasers have almost driven them off the market.

Inkjets are great in that they're cheap, many of them do color, and there are many kinds which aren't PCL but are understood by Ghostscript anyway. If you print very infrequently (less than weekly, say), you should be

careful to buy a printer whose print head gets replaced with every ink cartrige: infrequent use can lead to the drying of the ink, both in the ink cartrige and in the print head. The print heads you don't replace with the cartrige tend to cost nearly as much as the printer (~$200 for an Epson Stylus 800) once the warranty runs out (the third such repair, just after the warranty expired, totalled one informant's Stylus 800). Be careful, check print head replacement costs ahead of time, and run at least a cleaning cycle if you don't actually print anything in a given week. (Conversely, toner starts out dry, and ribbon ink won't evaporate for years...if you truly print only rarely, but neither a dot matrix nor a laser makes sense, consider buying no printer and taking your PostScript files to a copy shop...)

A parallel interface is a cheap way to make your printer print a lot faster than a serial line, and everyone's got a parallel port in their PC. Nowadays, though, a lot of printers are moving to Universal Serial Bus. Parallel ports may be obsolete soon.

3.17. Power Protection

I strongly recommend that you buy a UPS to protect your hardware and data. MOV−filtered power bars make nice fuses (they're cheap to replace), but they're not enough. I have written a UPS HOWTO that provides more complete coverage of what used to be in this section.

3.18. Radio Frequency Interference

(Thanks to Robert Corbett <Robert.Corbett@Eng.Sun.COM> for contributing much of this section)

Radio Frequency Interference (RFI) is a growing problem with PC−class machines. Today's processor speeds are such that the electromagnetic noise generated by a PC's circuitry in normal operation can degrade or jam radio and TV reception in the neighborhood. Such noise is called Radio Frequency Interference (RFI). Computers, as transmitting devices, are regulated by the Federal Communications Commission (FCC).

FCC regulations recognize two classes of computer:

If a PC is to be used in a home or apartment, it must be certified to be FCC class B. If it is not, neighbors have a legal right to prevent its use. FCC class A equipment is allowed in industrial environments.

Many systems are not FCC class B. Some manufacturers build boxes that are class B and then ship them with class A monitors or external disk drives. Even the cables can be a source of RFI.

It pays to be cautious. For example, the Mag MX17F is FCC class B. There are less expensive versions of the MX17 that are not. The Mag MX17 is a great monitor (I wish I had one). It would be painful to own one and not be allowed to use it.

An upgradeable system poses special problems. A system that is FCC class B with a 33 MHz CPU might not be when the CPU is upgraded to a 50 or 66 MHz CPU. Some upgrades require knockouts in the case to be removed. If a knockout is larger than whatever replaces it, RFI can leak out through the gap. Grounded metal shims can eliminate the leaks.

Even Class B systems don't mix well with wireless phonesets (not cellular phones, but the kind with a base station and antennaed headset). You'll often find a wireless phone hard to use withing 20 feet of a Class B machine.

To cut down on RFI, get a good metal case with tight joints, or at least make sure any plastic one you buy has a conductive lining. You can also strip the painted metal−to−metal contacting parts of paint so that there's good conductive metal contact. Paint's a poor conductor in most cases, so you can get some benefit from this.

4. What To Optimize

4.1. First, add more memory

Max out your memory. Having lots of free memory will improve your virtual−memory performance (and Unix takes advantage of extra memory more effectively than Windows does). Fortunately, with RAM as cheap as it is now, a gigabyte or three is unlikely to bust your budget even if you're economizing.

4.2. Bus and Disk speeds

Most people think of the processor as the most important choice in specifying any kind of personal−computer system. But for typical job loads under Linux, the processor type is nearly a red herring it's far more important to specify a capable system bus and disk I/O subsystem. If you don't believe this, you may find it enlightening to keep top(1) running for a while as you use your machine. Notice how seldom the CPU idle percentage drops below 90%!

It's true that after people upgrade their motherboards they often do report a throughput increase. But this is often more due to other changes that go with the processor upgrade, such as improved cache memory or an increase in the clocking speed of the system's front−side bus (enabling data to get in and out of the processor faster).

If you're buying for Linux on a fixed budget, it makes sense to trade away some excess processor clocks to get a faster bus and disk subsystem. If you're building a monster hot−rod, go ahead and buy that fastest available processor but once you've gotten past that gearhead desire for big numbers, pay careful attention to bus speeds and your disk subsystem, because that's where you'll get the serious performance wins. The gap between processor speed and I/O subsystem throughput has only widened in the last five years.

How does it translate into a recipe in 2004? Like this; if you're building a hot rod,

- *Do* buy a machine with the fastest available "front−side" (e.g. processor−to−memory) bus. (In February 2004 the maximum is still 266 MHz.)
- *Do* get a high−speed SCSI controller and the fastest SCSI disks you can get your hands on.

If you're economizing, you can back down on these. But in trading away SCSI for EIDE your reliability (expected time before failure) will drop. We'll cover this in more detail in the next section.

4.3. Disk Wars: IDE vs. SCSI

For the fastest disks you can find, pay close attention to average seek and latency time. The former is an average time required to seek to any track; the latter is the maximum time required for any sector on a track to come under the heads, and is a function of the disk's rotation speed.

Of these, average seek time is *much* more important. When you're running Linux or any other virtual−memory operating system, a one millisecond faster seek time can make a really substantial difference in system throughput. Back when PC processors were slow enough for the comparison to be possible (and I was running System V Unix), it was easily worth as much as a 30MHz increment in processor speed. Today the corresponding figure would probably be as much as 300MHz!

The manufacturers themselves avoid running up seek time on the larger−capacity drives by stacking platters vertically rather than increasing the platter size. Thus, seek time (which is proportional to the platter radius and head−motion speed) tends to be constant across different capacities in the same product line. This is good because it means you don't have to worry about a capacity−vs.−speed tradeoff.

Disks of less than 40GB capacity simply aren't being manufactured anymore; there's no margin in them. Our spies tell us that all major disk makers retooled their lines a while back to produce 9GB unit platters, which are simply being stacked 2N per spindle to produce ranges of drives with roughly 18GB increments of capacity.

Average drive latency is inversely proportional to the disk's rotational speed. For years, most disks spun at 3600 rpm; most disks now spin at 7,200 or 10,000rpm, and high−end disks at 15,000 rpm. These fast−spin disks run extremely hot; cooling is becoming a critical constraint in drive design.

Another basic decision is IDE vs. SCSI. Either kind of disk costs about the same, but the premium for a SCSI card varies all over the lot, partly because of price differences between VLB and PCI SCSI cards and especially because many motherboard vendors bundle an IDE chipset right on the system board. SCSI gives you better speed and throughput and loads the processor less, a win for larger disks and an especially significant consideration in a multi−user environment; also it's more expandable. You can have at most two IDE devices; four for EIDE. SCSI permits up to 7 (15 for wide SCSI).

Admittedly, the case for SCSI has eroded a bit since 2001; the new generation of IDE drives is very fast, and controller cards now normally feature a channel per drive and DMA (Direct Memory Access), so that some of of the multi−user contention problems that used to dog IDE have diminished. At 10KRPM and below, IDE is as good as SCSI now (a painful admission for an old−time IDE−hater like me), but at the 15KRPM high end SCSI still rules.

Of course, IDE is cheaper. Many motherboards have IDE right on board now; if not, you'll pay maybe $15 for an IDE adapter board, as opposed to $200+ for the leading SCSI controller. Also, the cheap SCSI cabling most vendors ship can be flaky. You have to use expensive high−class cables for consistently good results. See Mark Sutton's horror story.

4.3.1. Enhanced IDE

These days you seldom see plain IDE; souped−up variants are more usual. These are "Enhanced IDE" (E−IDE) and "Fast AT Attachment" (usually ATA for short). ATA is Seagate's subset of E−IDE, excluding some features designed to permit chaining with CD−ROMs and tape drives using the "ATAPI" interface (an E−IDE extension); in practice, ATA and E−IDE are identical.

You'll need to be careful about chaining in CD−ROMs and tape drives when using IDE/ATA. The IDE bus sends all commands to all disks; they're supposed to latch, and each drive then checks to see whether it is the intended target. The problem is that badly−written drivers for CD−ROMs and tapes can collide with the disk command set. It takes expertise to match these peripherals.

Neither ATA nor E−IDE has the sustained throughput capacity of SCSI (they're not designed to) but they are 60−90% faster than plain old IDE. E−IDE's new "mode 3" boosts the IDE transfer rate from IDE's 3.3MB/sec to 13.3MB/sec. The new interface supports up to 4 drives of up to 8.4 gigabytes capacity.

E−IDE and ATA are advertised as being completely compatible with old IDE. Theoretically, you can mix IDE, E−IDE and ATA drives and controllers any way you like, and the worst result you'll get is conventional

IDE performance if the enhancements don't match up (the controller picks the lowest latch speed). In practice, some IDE controllers (notably the BusLogic) choke on enhanced IDE.

Accordingly, I recommend against trying to mix device types an an E−IDE/ATA bus. Unfortunately, this removes much of E−IDE/ATA's usefulness!

E−IDE on drives above 540MB does automatic block mapping to fool the BIOS about the drive geometry (avoiding limits in the BIOS type tables). They don't require special Unix drivers.

Many motherboards now support "dual EIDE" channels, i.e. two separate [E]IDE interfaces each of which can, theoretically, support two IDE disks or ATA−style devices.

4.3.2. Advantages of SCSI

For starters, SCSI is still at least 10%−15% faster than IDE/ATAPI running flat out. Like Windows, IDE is layered over an ancestral design (ST−506) that's antiquated and prone to failure under stress. For example, on the Tyan K7 motherboards, there are known data−corruption problems with the ATA controller in the presence of various DMA−using bus−mastering cards.

SCSI, on the other hand, was designed from the beginning to scale up well to high−speed, high−throughput systems. Because it's perceived as a "professional" choice, SCSI peripherals are generally better engineered than IDE/ATAPI equivalents, and new high−performing drive technologies tend to become available in SCSI first. You'll pay a few dollars more, but for Linux the cost is well repaid in increased throughput and reliability.

The one aspect of SCSI that often gets overlooked is that it is a true multitasking interface, thanks to the "disconnect/reconnect" sequence that almost all SCSI hardware implements. With disconnect/reconnect, if a target device has to perform some kind of time−consuming mechanical operation (e.g., a seek in the case of a disk or a medium position operation in the case of a tape drive) the device will release control of the SCSI bus and allow it to be used for some other operation.IDE/ATAPI has no such capability and is often responsible for a system stall while a disk, CD−drive or tape drive seeks to the desired medium position.

(Incidentally, SCSI performance can sometimes be improved by setting the ID of the most frequently used disk drive as high as possible. The SCSI priority pecking order is such that devices with higher ID's get first crack at the bus when arbitration occurs during the selection phase.)

Rick's comments from 2001 are still apposite: "They call me a SCSI bigot. So be it but my hardware keeps being future−proof, I don't have to run bizarre emulation layers to address CDRs, I never run low on IRQs or resort to IRQ−sharing (on account of 3−4 ATA controllers each needing one, plus special adapters for scanners, etc.), all my hard drives have hardware−level hot−fix, all my hard disk/CD/tape/etc. devices support a stable standard rather than this month's cheap extension kludge, and I don't have to worry about adverse interactions at the hardware or driver levels from mixing ATA and SCSI."

The cutting edge in SCSI devices is ultra wide LVD (low−voltage−differential) SCSI drives with 320MB/sec transfer speed, running over a 68−pin cable (this is twice as fast as the LVD−160 drives we used last time around). Vendors often call LVD drives "SCSI−3", which is incorrect as most of these devices don't have built−in support for the entire SCSI−3 protocol, and it would be overdesign if they did (the extra commands are designed for use with CD and multimedia devices).

Fast ultra LVD is a bit more expensive to support than the older versions of SCSI (for which key words are

"single−ended", describing the electrical interface, and "narrow", describing the width of data transfers over the older−style 50−pin connector). Thus, you're likely to find it only on hard drives that are physically capable of doing high−speed data access off their media; slower devices such as tapes and CD drives are normally still built with the narrow single−ended variant.

The LVD−160 standard defines the SCSI bus, not the drive itself. Therefore, when used with a single hard drive in a lightly loaded system (e.g., a Linux machine supporting only one user) LVD−160 will have only a marginal effect on system performance. This is because a single hard drive running flat out will use only about 15−20 percent of the available bandwidth, as current drive technology can manage no more than about 28−30 MB/sec off the platters, less if a time consuming seek is involved. This rate could be higher, of course, if a read request was pending and the drive had cached the desired data. Where the LVD−160 bandwidth really becomes advantageous is in implementations of multiple drives (e.g., RAID 5) and/or when activities produce the frequent issue of drive access commands. The latter condition would be common in any environment that supports a lot of users.

Current SCSI drives are not quite fast enough to flood more than half the SCSI bus bandwidth, so you can have at least two drives on a single bus pumping full speed without using it up. In reality, you don't keep drives running full speed all the time, so you should be able to have 3−4 drives on a bus before you really start feeling bandwidth crunch.

4.3.3. SCSI Terminology

The following, by Ashok Singhal <ashoks@duckjibe.eng.sun.com> of Sun Microsystems with additions by your humble editor, is a valiant attempt to demystify SCSI terminology.

The terms "SCSI", "SCSI−2", and "SCSI−3" refer to three different specifications. Each specification has a number of options. Many of these options are independent of each other. I like to think of the main options (there are others that I'll skip over because I don't know enough about them to talk about them on the net) by classifying them into five categories:

4.3.3.1. Logical: SCSI−1, SCSI−2, SCSI−3

This refers to the commands that the controllers understand. You'll no longer see SCSI−1 in new hardware. SCSI−3 is a superset of SCSI−2 including commands intended for CD−R and streaming multimedia devices.

4.3.3.2. Electrical Interface

- single−ended (max cable length 6 meters)
- differential (max cable length 25 meters)

This option is independent of command set, speed, and path width. Differential is less common but allows higher transfer speeds, better noise immunity and longer cables. It's rare in SCSI−1 controllers.

You will normally see single−ended SCSI controllers on low−speed devices such as tapes and CD drives, and differential SCSI on hard drives (look for the specification LVD which means "low−voltage differential").

Nowadays most controllers support both electrical interfaces, but if you mix LVD with single−ended on the same chain, the whole chain will fall back to single−ended (and possibly halve the speed of the faster devices).

4.3.3.3. Handshake

- Asynchronous (acknowledge each word (8, 16 or 32 bits) transferred.
- Synchronous (multiple−word transfers permitted between ACKS).

Synchronous is faster. This mode is negotiated between controller and device; modes may be mixed on the same bus. This is independent of command set, data width, and electrical interface.

4.3.3.4. Synchronous Speed (does not apply for asynchronous option)

Normal transfer speed is 5 megabytes/sec. The "fast" option (10 mb/sec) is defined only in SCSI−2 and SCSI−3. Fast−20 (or "Ultra") is 20 mb/sec; Fast−40 (or "Ultra−2") is 40MB/sec. The fast options basically defines shorter timing parameters such as the assertion period and hold time.

The parameters of the synchronous transfer are negotiated between each target and initiator so different speed transfers can occur over the same bus.

4.3.3.5. Path width

The standard SCSI data path is 8 bits wide. The "wide" option exploits a 16− or 32−bit data path (uses 68−pin rather than 50−pin data cables). You also get 4−bit rather than 3−bit device IDs, so you can have up to 16 devices. The wide option doubles or quadruples your transfer rate, so for example a fast−20/wide SCSI link using 16 bits transfers 40mb/sec.

What are those "LUN" numbers you see when you boot up? Think of them as sub−addresses on the SCSI bus. Most SCSI devices have only one "logical" device inside them, thus they're LUN zero. Some SCSI devices can, however, present more than one separate logical unit to the bus master, with different LUNs (0 through 7). The only context in which you'll normally use LUNs is with CD−ROM juke boxes. Some have been marketed that offer up to 7 CD−ROMS with one read head. These use the LUN to differentiate which disk to select.

(There's history behind this. Back in the days of EISA, drives were supposed to be under the control of a separate SCSI controller, which could handle up to 7 such devices (15 for wide SCSI). These drives were to be the Logical Units; hence the LUN, or Logical Unit Number. Then, up to 7 of these SCSI controllers would be run by the controller that we today consider the SCSI controller. In practice, hardware cost dropped so rapidly, and capability increased so rapidly, it became more logical to embed the controller on the drive.)

4.3.4. Avoiding Pitfalls

Here are a couple of rules and heuristics to follow:

Rule 1: Total SCSI cable length (both external and internal devices) must not exceed six meters. For modern Ultra SCSI (with its higher speed) cut that to three feet!

It's probably not a good idea to cable 20MB/s or faster SCSI devices externally at all. If you must, one of our informants advises using a Granite Digital "perfect impedance" teflon cable (or equivalent); these cables basically provide a near−perfect electrical environment for a decent price, and can be ordered in custom configurations if needed.

A common error is to forget the length of the ribbon cable used for internal devices when adding external ones (that is, devices chained to the SCSI board's external connector).

Rule 2: Both ends of the bus have to be electrically terminated.

On older devices this is done with removable resistor packs typically 8−pin−inline widgets, yellow or blue, that are plugged into a plastic connector somewhere near the edge of the PCB board on your device. Peripherals commonly come with resistor packs plugged in; you must *remove* the packs on all devices except the two end ones in the physical chain.

Newer devices advertised as having "internal termination" have a jumper or switch on the PCB board that enables termination. These devices are preferable, because the resistor packs are easy to lose or damage.

Rule 3: No more than seven devices per chain (fifteen for Wide SCSI).

There are eight SCSI IDs per controller. The controller reserves ID 7 or 15, so your devices can use IDs 0 through 6 (or 0 through 14, wide). No two devices can share an ID; if this happens by accident, neither will work.

The conventional ID assignments are: Primary hard disk = ID 0, Secondary hard disk = ID 1, Tape = ID 2. Some Unixes (notably SCO) have these wired in. You select a device's ID with jumpers on the PCB or a thumbwheel.

SCSI IDs are completely independent of physical device chain position.

Heuristic 1: If you're buying narrow SCSI, stick with controllers and devices that use the Centronics−style 50−pin connector. Internally these connectors are physically identical to diskette cables. Externally they use a D50 shell. This "standard" connector is common in the desktop/tower/rackmount−PC world, but you'll find lots of funky DIN and mini−DIN plugs on devices designed for Macintosh boxes and some laptops. Ask in advance and don't get burned.

Heuristic 2: For now, when buying a controller, go with an Adaptec xx42 or one of its clones such as the BusLogic 542. (I like the BusLogic 946 and 956, two particularly fast Adaptec clones well−supported under Linux.) The Adaptec is the card everybody supports and the de−facto standard. Occasional integration problems have been reported with Unix under Future Domain and UltraStor cards, apparently due to command−set incompatibilities. At least, before you buy these, make sure your OS explicitly supports them.

However: Beware the combination of an Adaptec 1542 with a PCI Mach32 video card. Older (1.1) Linux kernels handled it OK, but all current ones choke. Your editor had to replace his 1542 because of this, swearing sulphurously the while.

Heuristic 3: You'll have fewer hassles if all your cables are made by the same outfit. (This is due to impedence reflections from minor mismatches. You can get situations where cable A will work with B, cable B will work with C, but A and C aren't happy together. It's also non−commutative. The fact that `computer to A to B' works doesn't mean that `computer to B to A' will work.

Heuristic 4. Beware Cheap SCSI Cables!

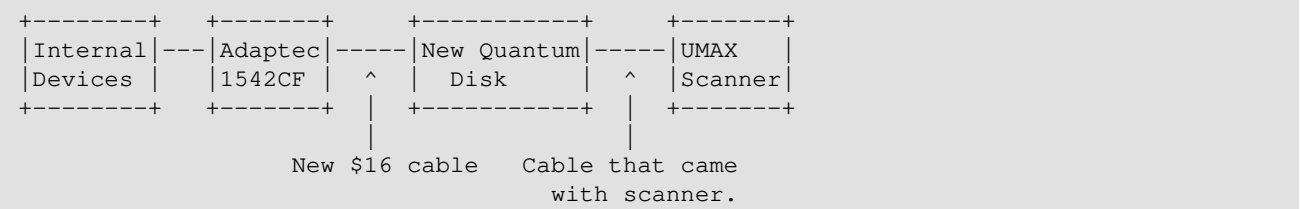
Mark Sutton tells the following instructive horror story in a note dated 5 Apr 1997:

I recently added an additional SCSI hard drive to my home machine. I bought an OEM packaged Quantum Fireball 2 gig SCSI drive (meaning, I bought a drive in shrinkwrap, without so much as mounting hardware or a manual. Thank God for Quantum's web page or I would have had no idea how to disable termination or set the SCSI ID on this sucker. Anyway, I digress...). I stuck the drive in an external mounting kit that I found in a pile of discarded computer parts at work and my that boss said I could have. (All 5 of my internal bays were full of devices.)

Anyway, I had my drive, and my external SCSI mounting kit, I needed a cable.

I went into my friendly local CompUSA in search of a SCSI cable, and side−by−side, on two hooks, were two "identical" SCSI cables. Both were 3 feet. Both had Centronics to Centronics connectors, both were made by the same manufacturer. They had slightly different model numbers. One was $16.00, one was $30.00. Of course, I bought the $16 cable.

Bad, I say, bad *bad* mistake. I hooked this sucker up like so:



Shortly after booting, I found that data all over my old internal hard drive was being hosed. This was happening in DOS as well as in Linux. Any disk access on either disk was hosing data on both disks, attempts to scan were resulting in corrupted scans *and* hosing files on the hard disks. By the time I finished swapping cables around, and checking terminations and settings, I had to restore both Linux and DOS from backups.

I went back to CompUSA, exchanged the $16 cable for the $30 one, hooked it up and had no more problems.

I carefully examined the cables and discovered that the $30 cable contained 24 individual twisted pairs. Each data line was twisted with a ground line. The $16 cable was 24 data wires with one overall grounded shield. Yet, both of these cables (from the same *manufacturer*) were being sold as SCSI cables!

You get what you pay for.

(Another correspondent guesses that the cheap cable probably said "Macintosh" on it. The Mac connector is missing most of its ground pins.)

4.3.5. More Resources

There's a USENET SCSI FAQ. Also see the home page of the T10 committee that writes SCSI standards.

There is a large searchable database of hard disk and controller information at the PC DISK Hardware Database.

4.4. Tuning Your I/O Subsystem

(This section comes to us courtesy of Perry The Cynic, <perry@sutr.cynic.org>; it was written in 1998. My own experience agrees pretty completely with his. I have revised the numbers in it since to reflect more recent developments.)

Building a good I/O subsystem boils down to two major points: *pick matched components* so you don't over−build any piece without benefit, and *construct the whole pipe such that it can feed what your OS/application combo needs*.

It's important to recognize that "balance" is with respect to not only a particular processor/memory subsystem, but also to a particular OS and application mix. A Unix server machine running the whole TCP/IP server suite has radically different I/O requirements than a video−editing workstation. For the "big boys" a good consultant will sample the I/O mix (by reading existing system performance logs or taking new measurements) and figure out how big the I/O system needs to be to satisfy that app mix. This is not something your typical Linux buyer will want to do; for one, the application mix is not static and will change over time. So what you'll do instead is design an I/O subsystem that is internally matched and provides maximum potential I/O performance for the money you're willing to spend. Then you look at the price points and compare them with those for the memory subsystem. That's the most important trade−off inside the box.

So the job now is to design and buy an I/O subsystem that is well matched to provide the best bang for your buck. The two major performance numbers for disk I/O are latency and bandwidth. Latency is how long a program has to wait to get a little piece of random data it asked for. Bandwidth is how much contiguous data can be sent to/from the disk once you've done the first piece. Latency is measured in milliseconds (ms); bandwidth in megabytes per second (MB/s). Obviously, a third number of interest is how big all of your disks are together (how much storage you've got), in Gigabytes (GB).

Within a rather big envelope, minimizing latency is the cat's meow. Every millisecond you shave off effective latency will make your system feel significantly faster. Bandwidth, on the other hand, only helps you if you suck a big chunk of contiguous data off the disk, which happens rarely to most programs. You have to keep bandwidth in mind to avoid mis−matching pieces, because (obviously) the lowest usable bandwidth in a pipe constrains everything else.

I'm going to ignore IDE. IDE is no good for multi−processing systems, period. You may use an IDE CD−ROM if you don't care about its performance, but if you care about your I/O performance, go SCSI. (Beware that if you mix an IDE CD−ROM with SCSI drives under Linux, you'll have to run a SCSI emulation layer that is a bit flaky.)

Let's look at the disks first. Whenever you seriously look at a disk, *get its data sheet*. Every reputable manufacturer has them on their website; just read off the product code and follow the bouncing lights. Beware of numbers (`<12ms fast!') you may see in ads; these folks often look for the lowest/highest numbers on the data sheet and stick them into the ad copy. Not dishonest (usually), but ignorant.

What you need to find out for a disk is:

1. What kind of SCSI interface does it have? Look for "fast", "ultra", and "wide". Ignore disks that say "fiber" (this is a specialty physical layer not appropriate for the insides of small computers). Note that you'll often find the same disk with different interfaces.
2. What is the "typical seek" time (ms)? Make sure you get "typical", not "track−to−track" or "maximum" or some other measure (these don't relate in obvious ways, due to things like

head−settling time).

3. What is the rotational speed? This is typically 4500, 5400, 7200, 10000, or 15000 rpm (rotations per minute). Also look for "rotational latency" (in ms). (In a pinch, average rotational latency is approx. 30000/rpm in milliseconds.)
4. What is the media transfer rate or speed (in MB/s)? Many disks will have a range of numbers (say, 7.2−10.8MB/s). Don't confuse this with the "interface transfer rate" which is always a round number (10 or 20 or 40MB/s) and is the speed of the SCSI bus itself.

These numbers will let you do apple−with−apples comparisons of disks. Beware that they will differ on different−size models of the same disk; typically, bigger disks have slower seek times.

Now what does it all mean? Bandwidth first: the `media transfer rate' is how much data you can, under ideal conditions, get off the disk per second. This is a function mostly of rotation speed; the faster the disk rotates, the more data passes under the heads per time unit. This constrains the sustained bandwidth of *this disk*.

More interestingly, your effective latency is the sum of typical seek time and rotational latency. So for a disk with 8.5ms seek time and 4ms rotational latency, you can expect to spend about 12.5ms between the moment the disk `wants' to read your data and the moment when it actually starts reading it. This is the one number you are trying to make small. Thus, you're looking for a disk with low seek times and high rotation (RPM) rates.

For comparison purposes, the first hard drive I ever bought was a 20MB drive with 65ms seek time and about 3000RPM rotation. A floppy drive has about 100−200ms seek time. A CD−ROM drive can be anywhere between 120ms (fast) and 400ms (slow). The best IDE harddrives have about 10−12ms and 5400 rpm. The best SCSI harddrive I know (the Fujitsu MAM) runs 3.9ms/15000rpm.

Fast, big drives are expensive. Really big drives are very expensive. Really fast drives are pretty expensive. On the other end, really slow, small drives are cheap but not cost effective, because it doesn't cost any less to make the cases, ship the drives, and sell them.

In between is a sweet spot where moving in either direction (cheaper or more expensive) will cost you more than you get out of it. The sweet spot moves (towards better value) with time. Right now (early 2004), it's about at 36GB drives, 6ms, 10000rpm, ultra2 SCSI. If you can make the effort, go to your local computer superstore and write down a dozen or so drives they sell naked . (If they don't sell at least a dozen hard drives naked, find yourself a better store. Use the Web, Luke!) Plot cost against size, seek and rotational speed, and it will usually become pretty obvious which ones to get for your budget.

Do look for specials in stores; many superstores buy overstock from manufacturers. If this is near the sweet spot, it's often surprisingly cheaper than comparable drives. Just make sure you understand the warranty procedures.

Note that if you need a lot of capacity, you may be better off with two (or more) drives than a single, bigger one. Not only can it be cheaper but you end up with two separate head assemblies that move independently, which can cut down on latency quite a bit (see below).

Once you've decided which kind of drive(s) you want, you must decide how to distribute them over one or more SCSI buses. Yes, you *may* want more than one SCSI bus. (My current desktop machine has three.) Essentially, the trick is to make sure that all the disks on one bus, talking at the same time, don't exceed the capacity of that bus. At this time, I can't recommend anything but an Ultra/Wide SCSI controller. This means that the attached SCSI bus can transfer data at up to 40MB/s for an Ultra/Wide disk, 20MB/s for an Ultra/narrow disk, and 10MB/s for a `fast SCSI' disk. These numbers allow you do do your math: an 8MB/s

disk will eat an entire bus on its own if it's fast (10MB/s). Three 6MB/s ultra/narrow disks fit onto one bus (3x6=18MB/s<20MB/s), but just barely. Two ultra/wide Cheetahs (12.8MB/s) will share an (ultra/wide) bus (25.6<40), but they would collide on an ultra/narrow bus, and any one Cheetah would be bandwidth constrained on a (non−ultra) `fast' bus (12.8 > 10).

If you find that you need two SCSI buses, you can go with dual channel versions of many popular SCSI controller cards (including the Adaptec). These are simply two controllers on one card (thus taking only one PCI slot). This is cheaper and more compact than two cards; however, on some motherboards with more than 3 PCI slots, using two cards may be somewhat faster (ask me what a PCI bridge is :−).

SCSI performance can sometimes be improved by setting the ID of the most frequently used disk drive as high as possible. The SCSI priority pecking order is such that devices with higher ID's get first crack at the bus when arbitration occurs during the selection phase.

How do you deal with slow SCSI devices CD−ROMS, scanners, tape drives, etc.? If you stick these onto a SCSI bus with fast disks, they will slow down things a bit. You can either accept that (as in "I hardly ever use my scanner anyway"), or stick them onto a separate SCSI bus off a cheap controller card. Or you can (try to) get an ATA version to stick onto that inevitable IDE interface on your motherboard. The same logic applies to disks you won't normally use, such as removables for data exchange.

If you find yourself at the high end of the bandwidth game, be aware that the theoretical maximum of the PCI bus itself is 132MB/s. That means that a dual ultra/wide SCSI controller (2x40MB/s) can fill more than half of the PCI bus's bandwidth, and it is not advised to add another fast controller to that mix. As it is, your device driver better be well written, or your entire system will melt down (figuratively speaking).

Incidentally, all of the numbers I used are optimal bandwidth numbers. The real scoop is usually somewhere between 50−70% of nominal, but things tend to cancel out the drives don't quite transfer as fast as they might, but the SCSI bus has overhead too, as does the controller card.

Whether you have a single disk or multiple ones, on one or several SCSI buses, you should give careful thought to their partition layout. Given a set of disks and controllers, this is the most crucial performance decision you'll make.

A partition is a contiguous group of sectors on the disk. Partitioning typically starts at the outside and proceeds inwards. All partitions on one disk share a single head assembly. That means that if you try to overlap I/O on the first and last partition of a disk, the heads must move full stroke back and forth over the disk, which can radically increase seek time delays. A partition that is in the middle of a partition stack is likely to have best seek performance, since at worst the heads only have to move half−way to get there (and they're likely to be around the area anyway).

Whenever possible, split partitions that compete onto different disks. For example, /usr and the swap should be on different disks if at all possible (unless you have outrageous amounts of RAM).

Another wrinkle is that most modern disks use `zone sectoring'. The upshot is that outside partitions will have higher bandwidth than inner ones (there is more data under the heads per revolution). So if you need a work area for data streaming (say, a CD−R pre−image to record), it should go on an outside (early numbered) partition of a fast−rotating disk. Conversely, it's a good convention to put rarely−used, performance−uncritical partitions on the inside (last).

Another note concerns SCSI mode pages. Each (modern) SCSI disk has a small part of its disk (or a dedicated EEPROM) reserved for persistent configuration information. These parameters are called mode pages , for

the mechanism (in the SCSI protocol) for accessing them. Mode page parameters determine, among others, how the disk will write−cache, what forms of error recovery it uses, how its RAM cache is organized, etc. Very few configuration utilities allow access to mode page parameters (I use FWB Toolkit on a Mac it's simply the best tool I know for that task), and the settings are usually factory preset for, uh, Windows 95 environments with marginal hardware and single−user operation. Particularly the cache organization and disconnect/reconnect pages can make a tremendous difference in actual performance. Unfortunately there's really no easy lunch here − if you set mode page parameters wrong, you can screw up your data in ways you won't notice until months later, so this is definitely `no playing with the pushebuttons' territory.

Ah yes, caches. There are three major points where you could cache I/O buffers: the OS, the SCSI controller, and the on−disk controller. Intelligent OS caching is by far the biggest win, for many reasons. RAM caches on SCSI controller cards are pretty pointless these days; you shouldn't pay extra for them, and experiment with disabling them if you're into tinkering.

RAM caches on the drives themselves are a mixed bag. At moderate size (1−2MB), they are a potential big win for Windows 95/98, because Windows has stupid VM and I/O drivers. If you run a true multi−tasking OS like Linux, having unified RAM caches on the disk is a significant loss, since the overlapping I/O threads kick each other out of the cache, and the disk ends up performing work for nothing.

Most high−performance disks can be reconfigured (using mode page parameters, see above) to have `segmented' caches (sort of like a set−associative memory cache). With that configured properly, the RAM caches can be a moderate win, not because caching is so great on the disk (it's much better in the OS), but because it allows the disk controller more flexibility to reschedule its I/O request queue. You won't really notice it unless you routinely have >2 I/O requests pending at the SCSI level. The conventional wisdom (try it both ways) applies.

And finally I *do* have to make a disclaimer. Much of the stuff above is shameless simplification. In reality, high−performance SCSI disks are very complicated beasties. They run little mini−operating systems that are most comfortable if they have 10−20 I/O requests pending *at the same time*. Under those circumstances, the amortized global latencies are much reduced, though any single request may experience *longer* latencies than if it were the only one pending. The only really valid analysis are stochastic−process models, which we *really* don't want to get into here. :−)

---

5. But What If I'm Economizing?

If you are economizing, here's a simple rule:

- *Do* buy a CPU/motherboard one or two levels lower than commercial state of the art.

In February 2004, the PC market is all Pentium IV and AMD Athlon chips, speeds ranging from 2.2 to 3.4GHz. For best value, look at the middle of that range.

Why? Because of the way manufacturers' price−performance curves are shaped. The top−of−line system is generally boob bait for corporate executives and other people with more money than sense. Chances are the system design is new and untried if you're at the wrong point in the technology cycle, the chip may even be a pre−production sample, or an early production stepping with undiscovered bugs like the infamous FDIV problem in early Pentiums. You don't need such troubles. Better to go with a chip/motherboard combination that's been out for a while and is known good. It's not like you really need the extra speed, after all.

Besides, if you buy one of these gold−plated systems, you're only going to kick yourself three months later when the price plunges by 30%. Further down the product line there's been more real competition and the manufacturer's margins are already squeezed. There's less room for prices to fall, so you won't watch your new toy lose street value so fast. Its price will still drop, but it won't plummet sickeningly.

Again, bear in mind that the cheapest processor you can buy new today is plenty fast enough for Linux. So if dropping back a speed level or two brings you in under budget, you can do it with no regrets.

Another easy economy measure is looking for repaired or reconditioned parts with a warranty. These are often as good as new, and much cheaper. (This is an especially good tactic for monitors and hard drives.)

Your display is one of the areas where pinching pennies is *not* a good idea. You're going to be looking at that display for hours on end. You are going to be using the screen real estate constantly. Buy the best quality, largest screen you possibly can it will be worth it.

There is a fair amount of price variance among equivalent video cards, so shop aggressively here. We won't do this, but if you're on a budget, one easy thing to trade away is bit depth. Manufacturers like to include 24− and 32−bit "photographic" color as sizzle in their advertisements, but unless you're doing something like specialty photocomposition work or medical graphics you'll probably never use more than 65535 colors. So you can settle for 16−bit color (used to be you could settle for 8−bit, before websites started routinely stepping outside the 216−color "web−safe" palette).

On the other hand, you probably don't need the latest and greatest CD−ROM device. High−speed CD−ROMS are really designed for people playing CD−ROM games or other applications involving image and sound archives. If you're doing the Linux thing, chances are you'll primarily use CD−ROMs that are code archives. Your average transfer size will be small and an apparent speed of 6x or even 4x quite satisfactory. So, if you need to, here's a place to cut costs by buying well behind the leading edge.

---

6. Noise Control and Heat Dissipation

An increasingly critical aspect of machine design is handling the waste heat and acoustic noise of operation. This may seem like a boring subject, but cooling is a centrally important one if you want your ULB to last because thermal stress from the electronics' own waste heat is almost certainly what will kill it. You want that fatal moment to happen later rather than sooner. On the other hand, cooling makes acoustic noise, which human beings don't tolerate well.

This tradeoff bites harder than you think; it's the fundamental reason that, despite our money−is−no−object premise, we're not going to relatively exotic technologies like liquid−cooled overclocking or RAID disk arrays for a performance boost. Sure, they may initially look attractive but overclocked chips and banks of disk drives require massive cooling with lots of moving parts, and it's not good to be trying to do creative work like programming with anything that sounds quite so much like an idling jet engine sitting beside one's desk.

In 2001 we had already reached the point at which the thermal load vs. cooling−noise tradeoff is the effective limiting factor in the performance of personal machines. Ten years ago, even low−end and medium "server" machines differed from personal−PC designs in fairly important ways (different processor and bus types, different speed ranges, etc.) Nowadays specialized server architectures are in retreat at the high end of the market and everything else looks like a PC. And the difference between a "PC" and a "server" is mainly that servers live in server rooms, and are allowed to have monster cases with lots of noisy fans.

So how do we manage this tradeoff for a personal, desktop or desk−side machine? Careful choice of components and being willing to pay some price premium for cool−running and low−noise characteristics can help a lot. Even exceptionally clueful system integrators can't generally afford to do this, because they're under constant competitive pressure to cut price and costs by using generic components. But we aren't economizing here; we get to do it *right*.

Reducing expected noise and heat in a design call for different strategies. It's relatively easy to find decibel figures for the noisemaking parts in a PC design. And, once you know a little basic audiometry and a few basic rules of thumb, it's not hard to form a fair estimate of your design's noisiness. Estimating a design's heat dissipation is harder, partly because the waste−heat emission of a PC's subsystems tends to vary in a more complex way than the acoustic emissions of the mechanical parts. This means that you can and should try to design ahead for low noise, but on the other hand expect to have to monitor for heat−dissipation problems in your prototype and solve them by building in more cooling.

Here's the basic audiometry you need to know to control your design's noise emissions:

Sound is measured in *decibels*, abbreviated dB, relative to the threshold of audibility, "A". (Thus, sound levels above that threshold are written "dBA".) The scale is logarithmic, with every 3dB increment roughly doubling sound intensity.

For sounds that are not phase−related, decibel levels add as a logarithmic sum. Thus if X and Y are uncorrelated sound sources,

dBA(X + Y) = 10 * log(10 ^ (dBA(X)/10) + 10 ^ (dBA(Y)/10))

A consequence of the above formula is that dBA(X + Y) cannot be more than 3dB above the greater of dBA(X) and dBA(Y) for uncorrelated sources (6dB for perfectly correlated ones).

Sound from a point source decays by an inverse−square law, roughly 6dB for each doubling of distance.

Important thresholds on the decibel scale:

0 dBA
: Threshold of hearing

20 dBA
: Rustling leaves, quiet living room

30 dBA
: Quiet office

40 dBA
: Quiet conversation

45 dBA
: Threshold of distraction, according to EPA

50 dBA
: Quiet street, average office noise

60 dBA
: Normal conversation (1 foot distance)

70 dBA
: Inside car

75 dBA
: Loud singing (3 feet)

80 dBA
: Typical home−stereo listening level

The acoustic noise emitted by PCs is normally a combination of white noise produced by airflow, high−frequency noise produced by bearing friction in drive spindles and fans, and the constant frequency "blade passing" noise that all propellers emit (the latter is often more intense than white noise and bearing whine).

The best low−noise ball−bearing case fans emit around 20dBA. Typical sleeve−bearing fans emit 30−50dBA.

According to the indispensable Tom's Hardware site, you can expect to cut at least 5dB off the interior noise level of the computer with a good choice of case. We'll improve on that by adding sound−absorbing material to the interior.

We'll use these facts to estimate the design's noisiness later on.

---

7. Special Considerations When Buying Laptops

Up until about 1999 the laptop market was completely crazy. The technology was in a state of violent flux, with "standards" phasing in and out and prices dropping like rocks. Things are beginning to settle out a bit more now.

One sign of this change is that there are now a couple of laptop lines that are clear best−of−breeds for reasons having as much to do with good industrial design and ergonomics as the technical details of the processor and display.

In lightweight machines, I'm a big fan of the Sony VAIO line. I owned one from early 1999 until it physically disintegrated under the rigors of travel in late 2000, and could hardly imagine switching. They weigh 3.5 pounds, give you an honest 3 hours of life per detachable battery pack, have a very nice 1024x768 display, and are just plain *pretty*. Their only serious drawback is that they're not rugged, and often fall apart after a year or so of use.

If you want a full−power laptop that can compete with or replace your desktop machine, the IBM ThinkPad line is the bomb. Capable, rugged, and nicely designed (though somewhat heavyweight for my taste). I now use a ThinkPad X20, the lightest and smallest machine in the line, and like it a lot.

These machines are not cheap, though. If you're trying to save money by buying a no−name laptop, here are things to look for:

First: despite what you may believe, the most important aspect of any laptop is *not* the CPU, or the disk, or the memory, or the screen, or the battery capacity. It's the keyboard feel, since unlike in a PC, you cannot throw the keyboard away and replace it with another one unless you replace the whole computer. *Never buy any laptop that you have not typed on for a couple hours*. Trying a keyboard for a few minutes is not enough. Keyboards have very subtle properties that can still affect whether they mess up your wrists.

A standard desktop keyboard has keycaps 19mm across with 7.55mm between them. If you plot frequency of typing errors against keycap size, it turns out there's a sharp knee in the curve at 17.8 millimeters. Beware of "kneetop" and "palmtop" machines, which squeeze the keycaps a lot tighter and typically don't have enough oomph for Unix anyway; you're best off with the "notebook" class machines that have full−sized keys.

Second: with present flatscreens, 1024x768 color is the best you're going to do (though that may change soon). If you want more than that (for X, for example) you have to either fall back to a desktop or make sure there's an external−monitor port on the laptop (and many laptops won't support higher resolution than the flatscreen's).

Third: about those vendor−supplied time−between−recharge figures; *don't believe them*. They collect those from a totally quiescent machine, sometimes with the screen or hard disk turned off. Under DOS, you'd be lucky to get half the endurance they quote; under Unix, which hits the disk more often, it may be less yet. Figures from magazine reviews are more reliable.

Fourth: get either a CD−ROM drive or an Ethernet card. Otherwise initial load of your Unix could turn into a serious problem...

---

8. How to Buy

8.1. When to Buy

It used to be that good configurations for Unix were what the market called server machines, with beefed−up I/O subsystems and fast buses. No longer; today's servers are monster boxes with multiple power supplies and processors, gigabytes of memory, and industrial−grade air cooling they're not really suitable as personal machines. A typical SCSI desktop workstation is as much as you'll need.

Prices keep dropping, so there's a temptation to wait forever to buy. A good way to cope with this is to configure your system on paper, get a couple of initial estimates, then set a trigger price, below the lowest one, at what you're willing to pay. Then watch and wait. When the configuration cost hits your trigger price, place your order.

The advantage of this method is that it requires you to settle in your mind, well in advance, what you're willing to pay for what you're getting. That way, you'll buy at the earliest time you should, and won't stress too much out afterwards as it depreciates.

Before you shop, do your homework. Publications like "Computer Shopper" (and their web site at http://www.computershopper.com) are invaluable for helping you get a feel for prices and what clonemakers are doing. Another excellent site is ComputerESP.

8.2. Where to Buy

The most important where−to−buy advice is negative. Do *not* go to a traditional, business−oriented storefront dealership. Their overheads are high. So are their prices.

Especially, run do not walk away from any outfit that trumpets business solutions . This is marketing code for the kind of place that will justify a heavy price premium by promising after−sale service and training which, nine times out of ten, will turn out to be nonexistent or incompetent. Sure, they'll give you plush carpeting and a firm handshake from a guy with too many teeth and an expensive watch but did you really want to pay for that?

There are two major alternatives to storefront dealerships and one minor one. The major ones are mail order and computer superstores. The minor one is computer fairs.

8.3. Computer Fairs

I used to be a big fan of hole−in−the−wall stores run by immigrants from the other side of the International Date Line, but most of those places have been driven out of the regular retail game by the superstores. If you still have one in your neighborhood, you're lucky. The only place you normally find diaspora Chinese and Indians selling cheap PCs over the counter anymore is at computer fairs. (Usually they're doing it to publicize a mail−order business.)

You can find good loss−leader deals on individual parts at these fairs (they're especially good places to buy disk drives cheap). But I call them a minor alternative because it's hard to get a custom SCSI−based configuration tuned for Unix built for you at a fair. So you end up, effectively, back in the mail−order or Web channel.

8.4. Mail Order

Direct−mail or Web buying makes a lot of sense today for anyone with more technical savvy than J. Random Luser in a suit. Even from no−name mail−order houses, parts and system quality tend to be high and consistent, so conventional dealerships don't really have much more to offer than a warm fuzzy feeling. Furthermore, competition has become so intense that even mail−order vendors today have to offer not just lower prices than ever before but warranty and support policies of a depth that would have seemed incredible a few years back. For example, many bundle a year of on−site hardware support with their medium− and high−end "business" configurations for a very low premium over the bare hardware.

Note, however, that assembling a system yourself out of mail−order parts is *not* likely to save you money over dealing with the mail−order systems houses. You can't buy parts at the volume they do; the discounts they command are bigger than the premiums reflected in their prices. The lack of any system warranty or support can also be a problem even if you're expert enough to do the integration yourself because you also assume all the risk of defective parts and integration problems.

Watch out for dealers (Spectrum Trading for one) who charge ridiculous shipping fees. One of our spies reports he bought a hotswappable hard disc drive tray that weighed about 3 lbs. and cost $250 and they charged $25 to ship it UPS groud.

Don't forget that (most places) you can avoid sales tax by buying from an out−of−state mail−order outfit, and save yourself 6−8% depending on where you live. If you live near a state line, buying from a local outfit you can often win, quite legally, by having the stuff shipped to a friend or relative just over it. Best of all is a buddy with a state−registered dealer number; these aren't very hard to get and confer not just exemption from sales tax but (often) whopping discounts from the vendors. Hand him a dollar afterwards to make it legal.

(Note: I have been advised that you shouldn't try the latter tactic in Florida they are notoriously tough on "resale license" holders).

(Note II: The Supreme Court has ruled that states may not tax out−of−state businesses under existing law, but left the way open for Congress to pass enabling legislation. Let's hope the mail−order industry has good lobbyists.)

8.5. Computer Superstores

Big chain superstores like CompUSA and Circuit City give you a reasonable alternative to mail order. And there are good reasons to explore it these stores buy and sell at volumes that allow them to offer prices not far above mail−order. (They make back a lot of their margin on computer games and small accessories like mouse pads, cables, and floppy disks.)

One thing you should not buy mail−order if you can avoid it is a monitor. Monitors are subject to substantial quality variations even within the same make and model. Also, one good bump during shipping can twist the yoke on a monitor so the image is tilted with respect to the bezel. So buy your monitor face−to−face, picking the best out of three or four.

Another good argument for buying at a superstore is that you may have to pay return postage if you ship a mail−order system back. On a big, heavy system, this can eat your initial price savings.

The only major problem with superstores is that the salespeople who staff them aren't very bright or very clueful (it's a sort of Darwinian reverse−selection effect; these are the guys who are fascinated by computer technology but not smart enough to be techies). Most of them don't know from Linux and are likely to push things like two−button mice and (worse!) controllerless modems, that you can't use. Use caution and check your system manifest.

But if you shop carefully and don't fall for one of their name−brand "prestige" systems, you can get prices comparable to mail order with the comfort of knowing there's a trouble desk you can drive back to in a pinch. (Also, you *can* see your monitor before you buy!)

8.6. Other Buying Tips

You can often get out of paying tax just by paying cash, especially at computer shows. You can always say you're going to ship the equipment out of the state.

A lot of vendors bundle DOS or Windows and variable amounts of apps with their hardware. If you tell them to lose all this useless cruft they may shave $50 or $100 off the system price.

9. Questions You Should Always Ask Your Vendor

9.1. Minimum Warranty Provisions

The weakest guarantee you should settle for in the mail−order market should include:

- 72−hour burn−in to avoid that sudden infant death syndrome. (Also, try to find out if they do a power−cycling test and how many repeats they do; this stresses the hardware much more than steady burn−in.)
- 30 day money−back guarantee. Watch out for fine print that weakens this with a restocking fee or limits it with exclusions.
- 1 year parts and labor guarantee (some vendors give 2 years).
- 1 year of 800 number tech support (many vendors give lifetime support).

Additionally, many vendors offer a year of on−site service free. You should find out who they contract the service to. Also be sure the free service coverage area includes your site; some unscrupulous vendors weasel their way out with "some locations pay extra", which translates roughly to "through the nose if you're further away than our parking lot".

If you're buying store−front, find out what they'll guarantee beyond the above. If the answer is "nothing", go somewhere else.

9.2. Documentation

Ask your potential suppliers what kind and volume of documentation they supply with your hardware. You should get, at minimum, operations manuals for the motherboard and each card or peripheral; also an IRQ list. Skimpiness in this area is a valuable clue that they may be using no−name parts from Upper Baluchistan, which is not necessarily a red flag in itself but should prompt you to ask more questions.

9.3. A System Quality Checklist

There are various cost−cutting tactics a vendor can use which bring down the system's overall quality. Here are some good questions to ask:

- If you're buying a factory−configured system, does it have FCC certification? While it's not necessarily the case that a non−certified system is going to spew a lot of radio−frequency interference, certification is legally required and becoming more important as clock frequencies climb. Lack of that sticker may indicate a fly−by−night vendor, or at least one in danger of being raided and shut down! (For further discussion, see the section on Radio Frequency Interference above.)
- Are the internal cable connectors keyed, so they can't be put in upside down? This doesn't matter if you'll never, ever *ever* need to upgrade or service your system. Otherwise, it's pretty important; and, vendors who fluff this detail may be quietly cutting other corners.

---

10. Things to Check when Buying Mail–Order

10.1. Tricks and Traps in Mail–Order Warranties

Reading mail–order warranties is an art in itself. A few tips:

Beware the deadly modifier "manufacturer's" on a warranty; this means you have to go back to the equipment's original manufacturer in case of problems and can't get satisfaction from the mail–order house. Also, manufacturer's warranties run from the date *they* ship; by the time the mail–order house assembles and ships your system, it may have run out!

Watch for the equally deadly "We do not guarantee compatibility". This gotcha on a component vendor's ad means you may not be able to return, say, a video card that fails to work with your motherboard.

Another dangerous phrase is "We reserve the right to substitute equivalent items". This means that instead of getting the high–quality name–brand parts advertised in the configuration you just ordered, you may get those no–name parts from Upper Baluchistan theoretically equivalent according to the spec sheets, but perhaps more likely to die the day after the warranty expires. Substitution can be interpreted as "bait and switch", so most vendors are scared of getting called on this. Very few will hold their position if you press the matter.

Another red flag: "Only warranted in supported environments". This may mean they won't honor a warranty on a non–DOS system at all, or it may mean they'll insist on installing the Unix on disk themselves.

One absolute show–stopper is the phrase "All sales are final". This means you have *no* options if a part doesn't work. Avoid any company with this policy.

10.2. Special Questions to Ask Mail–Order Vendors Before Buying

- Does the vendor have the part or system presently in stock? Mail order companies tend to run with very lean inventories; if they don't have your item in stock, delivery may take longer. Possibly *much* longer.
- Does the vendor pay for shipping? What's the delivery wait?
- If you need to return your system, is there a restocking fee? and will the vendor cover the return freight? Knowing the restocking fee can be particularly important, as they make keep you from getting real satisfaction on a bad major part. Avoid dealing with anyone who quotes more than a 15% restocking fee and it's a good idea, if possible, to avoid any dealer who charges a restocking fee at all.

Warranties are tricky. There are companies whose warranties are invalidated by opening the case. Some of those companies sell upgradeable systems, but only authorized service centers can do upgrades without invalidating the warranty. Sometimes a system is purchased with the warranty already invalidated. There are vendors who buy minimal systems and upgrade them with cheap RAM and/or disk drives. If the vendor is not an authorized service center, the manufacturer's warranty is invalidated. The only recourse in case of a problem is the vendor's warranty. So beware!

10.3. Payment Method

It's a good idea to pay with AmEx or Visa or MasterCard; that way you can stop payment if you get a lemon, and may benefit from a buyer−protection plan using the credit card company's clout (not all cards offer buyer−protection plans, and some that do have restrictions which may be applicable). However, watch for phrases like "Credit card surcharges apply" or "All prices reflect 3% cash discount" which mean you're going to get socked extra if you pay by card.

Note that many credit−card companies have clauses in their standard contracts forbidding such surcharges. You can (and should) report such practices to your credit−card issuer. If you already paid the surcharge, they will usually see to it that it is returned to you. Credit−card companies will often stop dealing with businesses that repeat such behavior.

10.4. Which Clone Vendors to Talk To

10.4.1. Some pans

Gateway: may also be a vendor to avoid. Apparently their newer machines don't have parity bits in their memories; memory is tested only on reboot. This is dubious design even for DOS, and totally unacceptable for Unix.

10.4.2. Some picks

In early August 2001 I designed an `Ultimate Linux Box' with Gary Sandine and John Pearson of Los Alamos Computers; you can read all about it These guys know what they are doing and are fun to work with. If you need a high−end Linux workstation, or your laboratory needs a computer cluster, talk with them.

11. After You Take Delivery

Your configuration is custom and involves slightly unusual hardware. Therefore, keep a copy of the configuration you wrote down, and check it against the invoice and the actual delivered hardware. If there is a problem, calling back your vendor right away will maximize your chances of getting the matter settled quickly.

---

12. Software to go with your hardware

I used to maintain an entire separate FAQ on Unixes for 386/486 and Pentium hardware. Times change, industries evolve, and I can now replace that FAQ with just three words:

Go get Linux!

---

13. Other Resources on Building Linux PCs

The PC Tech Guide offers pretty comprehensive descriptions of PC hardware technologies.

Andrew Comech's The Cheap /Linux/ Box page is a useful guide to building with current hardware that is updated every two weeks. Andrew also maintains a short−cut version.

The Caveat Emptor guide has an especially good section on evaluating monitor specifications.

Dick Perron has a PC Hardware Links page. There is lots and lots of good technical stuff linked to here. Power On Self Test codes, manufacturer address lists, common fixes, hard disk interface primer, etc.

Anthony Olszewski's Assembling A PC is an excellent guide to the perplexed. Not Linux−specific. If you're specifically changing a motherboard, see the Installing a Motherboard page. This one even has a Linux note.

Tom's Hardware Guide covers many hardware issues exhaustively. It is especially good about CPU chips and motherboards. Full of ads and slow−loading graphics, though.

The System Optimization Site has many links to other worthwhile sites for hardware buyers.

Christopher B. Browne has a page on Linux VARs that build systems. He also recommends the Linux VAR HOWTO.

Jeff Moe has a Build Your Own PC page. It's more oriented towards building from parts than this one. Less technical depth in most areas, but better coverage of some including RAM, soundcards and motherboard installation. Features nifty and helpful graphics, one of the better graphics−intensive pages I've seen. However, the hardware−selection advice is out of date.

The Linux Hardware Database provides, among other things (e.g., drivers, specs, links, etc.), user ratings for specific hardware components for use under Linux. Our ratings take a lot of the guess work out of choosing which hardware to buy for a Linux box. The site also provides several product−specific resources (i.e., drivers, workarounds, how−to) that help users get hardware working after they have made a purchase.