

Kissing Cousins, Lovers' Quarrels

So, not long after learning enough about UNIX to play around a little, and even managing to get a job working on a UNIX computer, I found myself faced with a terrible problem: Something was going very wrong on the machine I was charged to look after, and I had to fix it. My boss had locked all the manuals for the machine up in her office in order to end a long-standing feud we'd been having over whether we should turn the computer off at night. I admired her apparent faith in books as the only source of knowledge appropriate to solving the world's (or at least our computer's) problems, even if I was a little put off that she thought I'd give up in the absence of a few technical manuals, or suddenly decide I was wrong after all if I couldn't point to a book to prove my point.

Turner-offers are, to my way of thinking, one of the more lily-livered contingents of the computing world. Why turn off your computer at night when you can buy an uninterruptable power supply and enough surge protection to resist the annual output of Three Mile Island and rack up all the bragging points?

And how are you going to be able to brag about the uptime your single user desktop machine enjoys if it never runs for more than fourteen hours a day?

And how are you going to be able to brag about the uptime your single user desktop machine enjoys if it never runs for more than fourteen hours a day? You can't. Turning off a computer at night is nothing short of depriving yourself of the ability to buy more hardware and brag. It's stupid.

Leave-on-ers are, of course, the only right-minded people in the computer world. In addition to knowing a good opportunity to buy more hardware when they see it, and the sensu-

ous pleasures of saying, "My box has been up for more than three months without going down," leave-on-ers know that turning something on and off wears it out more quickly, or something.

Linux, and UNIX in general, presents an even better argument for leaving a computer on all night: If you don't, and you don't take a little additional care, it messes things up—just a little at first, then in a really, really bad way. Linux, you see, comes awake on many machines at around three in the morning and sets about doing some light housekeeping. It shuffles some log files around, deletes things that don't need to exist anymore, and shakes itself out. Sometimes, if things aren't quite right, it sends mail telling you so. If your machine isn't on during the early morning, it can't ever do any of this stuff. If it can't do this stuff, directories slowly fill with redundant files and scraps of messages the computer sends to itself. The /tmp directory, which you can think of as your computer's appendix, slowly fills up with the nail-bitings and waste products of a computer hard at work. Over time, you run out of space and the computer stops being able to keep the sort of temporary information it needs to do the essentials. That's what ours was doing: quietly strangling on its own rubbish.

There are, of course, ways around this. It wouldn't be any good if you had to live with something as arbitrary as a machine that has to be on at a certain time. At the time, however, I didn't know of any of these workarounds; I just knew my boss was a compulsive turner-offer who'd hidden the only evidence I had to thwart her misguided notions about "wearing out the CPU." So I turned to the Internet and begged for help on a UNIX newsgroup.

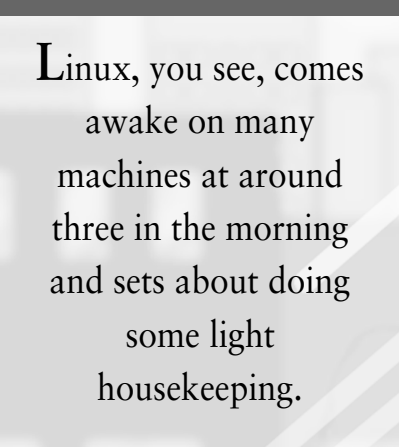
The folks there were initially very helpful. One gentleman in particular took care to answer each of my questions in an e-mail exchange, and the kindness with which he responded made me mist up.

Of course I could make it so the housecleaning took place at a time of day when the boss would allow the machine to be turned on; of course she was crazy to turn the machine off—how was she going to brag about uptime? Of course this was a simple fix—just edit a few files like so...

I reported back to my new mentor the next day, flush with success: All it had taken was a quick session with Emacs and the machine was humming!

Emacs? I'd used *Emacs* to edit a configuration file? Was I some sort of freak!? Did I not know how to use *vi*? Did I drive cement mixers to the drive-in movies and swat flies with hand grenades? Was I not aware of the simple elegance and power of *vi*?

I should have left it alone, apologized for my poor choice of tools, and continued to learn at the feet of this sage. As it was, I wasn't aware I'd set foot in the middle of one of the bloodiest and most protracted battles ever fought in the UNIX world, so I replied in kind. *Vi*, I argued, is for masochists and lickspittles



Linux, you see, comes
awake on many
machines at around
three in the morning
and sets about doing
some light
housekeeping.

on some sort of bizarre kick that causes second-year college students to run away to monastic cults until they get tired of eating porridge and sweeping the floors with rush brooms that are too short. Emacs, on the other hand, is a comfortable tool meant to

be used by people who want a hand in personalizing the text-editing experience, the most important thing a real UNIX user ever does. People who use vi, I posited, are backwards and probably use the word “new-fangled” while they tug on their suspenders.

My ex-guru was careful to forward my thoughtful missive to a few of his friends, who were happy to explain how wrong I was. My inbox stopped smoldering a few weeks later.

ON VIKING MARAUDERS AND RAMPAGING TROLLS

It isn’t hard, looking back on the history of UNIX, to see that Linux is the inheritor of more than a technical

outlook. It’s also the inheritor of a culture. Some of that was covered earlier when we examined where Linux came from and how it came to be where it is, but that’s the boring part. Linux has also inherited three decades of UNIX enthusiasts fighting among themselves, which is much more entertaining. Once you’re done with this chapter, you’ll have the barest essentials required for going out into the world and picking your own fights—and getting stomped by, as a friend of mine puts it, “suspenders-wearing UNIX graybeards,” who have been fighting about some of this stuff since before you were born.

Vi, I argued, is for masochists and lickspittles on some sort of bizarre kick that causes second-year college students to run away to monastic cults until they get tired of eating porridge and sweeping the floors with rush brooms that are too short.

Before going too much further, I should note that there are several types of conflicts within the Linux community, and they bear some examination.

The first is informed very heavily by whatever passed for a school of thought among Viking marauders in the Middle Ages. Although these Linux users don't have long ships to pile into, or ferocious seas to cross, or barbarian enemies to crush, they do have computers, the Internet, and everybody who disagrees with them on anything. They also have a wide variety of fairly good microbrews usually within easy driving distance, which isn't quite as good as a flagon of mead after a long day of smiting the enemy, but it'll do. A fight to these hearty souls is sustenance itself. "A vicious flame war is character building," they'll tell you, if they've decided to take time out from flaming each other long enough to explain what the flaming is all about. At work, they'll tell you, is no less than the Darwinian process of separating the weaklings of the herd from the gene pool, ensuring the survival of good ideas, and maintaining the health of the meritocracy that is Linux.

They await your arrival in the Linux community eagerly.

A second type of conflict is characterized by a slightly more nervous disposition about all the scuffling. They remember with unhappiness the near death of UNIX in the '80s and '90s, which they quickly point out was caused by unconscionable amounts of squabbling and scuffling among whole companies, which led to what is spoken of in hushed and fearful tones as "fragmentation." They're aware of history and quake at the thought of repeating it. They pop up in arguments and remind the combatants that although arguing about the merits of one tool over the

Linux has also inherited three decades of UNIX enthusiasts fighting among themselves, which is much more entertaining.

The Joy of Tech

by Nitrozac and Snaggy



It was a Wild, Wild Kernel, but *someone* had to tame it!

other is all well and good (and that the same argument has been going on for nigh on 30 years), the rest of the world will take it as a sure sign that use of UNIX stimulates testosterone production, territorial marauding, argumentativeness, and, ultimately, bloodshed. Sometimes these kind souls forget themselves and take to calling others names in their zeal to protect Linux from the same fate that befell the rest of the UNIX world, which pleases those of the “Viking” school of thought to no end.

They, too, welcome your arrival in the community, since there’s no point in worrying about fragmentation all the time if new people aren’t coming into the fold.

A third school of thought already bears an acknowledged label: the trolls. Trolls thrive in an environment like the Linux community, which lives so much of its life over the Internet, because they're generally safe from physical assault, and they're not required to keep a straight face while they're up to their trolling. As with many things Linux, the Jargon File provides ready definitions, and I'll relate the first two:

troll¹ (trōl) v. [From the Usenet group alt.folklore.urban]

To utter a posting on Usenet designed to attract predictable responses or flames; or, the post itself. Derives from the phrase “trolling for newbies,” which in turn comes from mainstream “trolling,” a style of fishing in which one trails bait through a likely spot hoping for a bite. The well-constructed troll is a post that induces lots of newbies and flamers to make themselves look even more clueless than they already do, while subtly conveying to the more savvy and experienced that it is in fact a deliberate troll. If you don't fall for the joke, you get to be in on it. See also: YHBT.

troll² (trōl) n. An individual who chronically trolls in sense 1; regularly posts specious arguments, flames, or personal attacks to a newsgroup, discussion list, or in email for no other purpose than to annoy someone or disrupt a discussion. Trolls are recognizable by the fact that they have no real interest in learning about the topic at hand—they simply want to utter flame bait. Like the ugly creatures after which they are named, they exhibit no redeeming characteristics, and as such, they are recognized as a lower form of life on the Net, as in, “Oh, ignore him, he's just a troll.”

Your average troll may well have some emotional investment in a flame war, but you'll probably never know what it is. They're more interested in keeping the entertainment the combatants are providing moving along.

You may have figured out from the definition that trolls can be subtle folk, despite the fireworks they seek to provoke, and that discerning a troll can be a tricky proposition. You're right. Determining whether something that makes your blood pressure rise and inspires you to compose an impassioned response on the spot has been issued by someone who is simply disagreeing or by someone trolling is tough to call. That's why it pays to remember that the newer you are to the Linux world (or any online community), the more likely you are to be easy prey for someone out to keep a fight going.

Trolls, of course, exploit something all newbies have in common: the desperate desire to fit in with their new community. If you think you're going to make points by leaping to the fore and smiting someone with a particularly reprehensible opinion, keep in mind that you may just be painting "newbie" in Day-Glo colors on your own forehead.

The Joy of Tech

by Nitrozac and Snaggy



By the way, trolls also eagerly await your arrival.

So...having covered the Linux community as it relates to conflict, onward with the conflicts themselves. I promise to make no attempt to present any of this fairly.

VI VERSUS EMACS: OF SCALPELS AND CHAINSAWS

Linux is, at root, built around text files. It's configured with text files, and the source code everyone's so happy to get their hands on arrives in text files. Most people who program with Linux use text editors to do their programming; in addition, anyone who's used one of the traditional e-mail programs under Linux such as Elm or PINE has used a text editor to compose their mail. The way a hammer or saw is essential to a carpenter, a text editor is essential to a Linux user.

It makes sense, then, that possibly the most venerable fight in all of Linuxdom—one that predates Linux itself—is over text editors. I'll introduce the contenders.

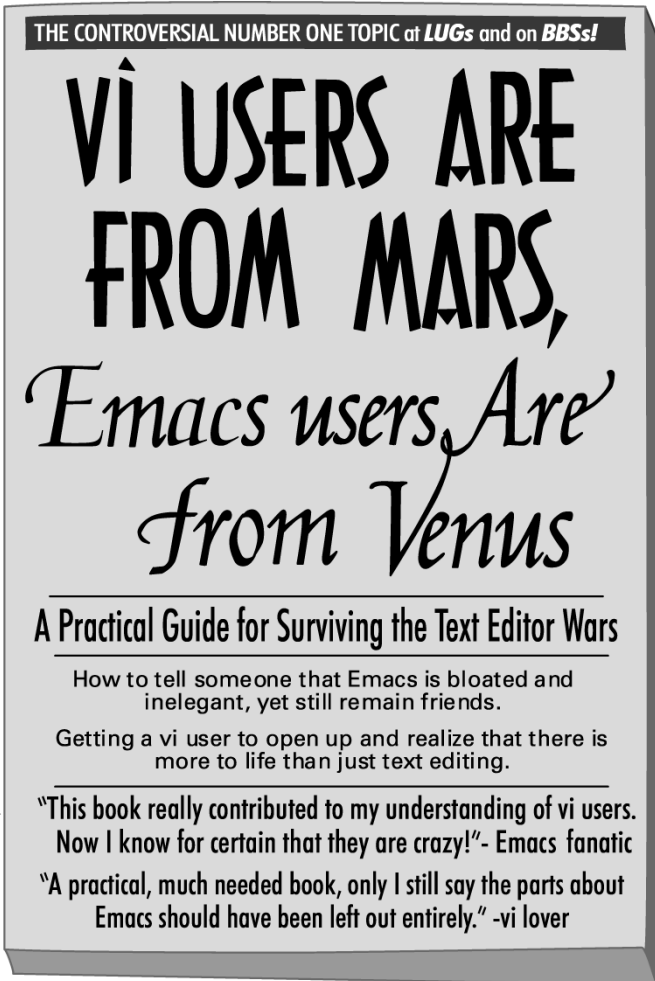
vi: Fast, Efficient, Perplexing

Vi is as close as the UNIX world comes to a standard editor. It turns up everywhere, and it's been around for a long, long time. Words used to describe vi are “elegant,” “powerful,” “light-weight,” “frustrating,” “irritating,” and “inscrutable.”

The elegant and powerful bits are pretty easy to grasp once you spend some time with vi. You can accomplish a lot with very few keys. It's just important that you remember all those possible combinations of “just a few keys,” and prepare to spend weeks training yourself that you can't simply start up vi and start typing. It's a *modal* text editor, which means that you can either be in text-entry mode or command mode. It takes commands to do all those wonderful, powerful, elegant things when it's in command mode, in addition to a few other things, like

The Joy of Tech

by Nitrozac and Snaggy



beeping at you repeatedly and making you sorry to be alive if you thought you were in text-entry mode.

In addition to knowing that vi will frustrate you to no end over the course of learning it, you should also know that you must never, ever, pronounce vi's name like it looks. If you call vi, um, vi, you've once again painted "newbie" in Day-Glo colors on your forehead. Vi is to be pronounced like it looks (the other way it looks): as two distinct letters: "vee eye." Practice this.

Vi is held up by its adherents as the only editor a serious Linux hacker will ever want or need. After all, it's elegant, powerful, lightweight, yada yada yada. It's also got a funny name, it's hard to figure out, and it's singularly unhelpful.

Vi fans are proud of their hard-earned knowledge and mastery. You can find Web pages with buttons that read, "Crafted with the vi editor." There are vi coffee cups with the commands you might need to know printed on the sides. Vi people take a ruthless pleasure in telling newbies to use vi. They know what the newbie's getting into, and they aren't about to warn them. If the newbie works in an adjoining cubicle, vi users sit quietly and listen to the newbie's terminal beep over and over and over while the first lesson of vi is pounded into their skulls: You can't "just start typing" with vi. People who want to do that are morally defective.

Emacs: You Could Settle for Just Editing Text...

Emacs is vi's opposite number. Where vi is lightweight, Emacs is crushing. Where vi is elegant, Emacs is elegant—but in a different, alien sort of way. Where vi is powerful in a manner that involves accomplishing much with little, Emacs is powerful in a manner that leaves text files not only successfully edited, but bruised and flattened.

Not content to simply edit files, Emacs can do a lot more, enabling you to manage your calendar, play games, and read mail, in addition to handling IRC chatting, remembering everything you ever wrote and making recommendations on things that might be pertinent to what you're writing about at the moment. Emacs can also remember the e-mail addresses of anyone who's ever written you, insert messages designed to bother the National Security Agency at the end of every mail you send, or even pretend it's vi.

When it comes down to it, you could fire up Emacs and go live in it all day long. It does that much. Some people make very

firm decisions about what they will and will not do with Emacs once they discover it's taking over their lives.

Since we're dealing with efficiency freaks, that's a wasted tenth of a second, and that's unforgiveable.

All this power requires memorizing a lot of commands, and it comes at the expense of your ability to remember to eat or sleep. On the other hand, once you've mastered Emacs, you've probably reached a level where food and sleep are no longer necessary. You have to take the good with the bad, I suppose.

Vi users, for their part, often look at Emacs users as destroyers of the UNIX tradition.

"Emacs," they argue, "is bloated and inelegant."

They buttress this argument with two points. The first is that even on a bad day, vi takes up about a tenth of the memory that Emacs does. The second is that it usually takes more keys to do things with Emacs. Compare:

To save a file with vi, you type the following:

```
:w
```

Sometimes you need to press the Esc key first, but that's nothing compared to the nightmarish complexity of Emacs. To save a file with Emacs, you must execute the following command:

```
CTRL+X, CTRL+S
```

See?

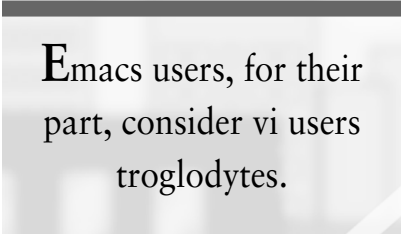
That's a tenth of a second off your life. Since we're dealing with efficiency freaks, that's a wasted tenth of a second, and that's unforgiveable.

In addition, vi lovers argue, vi is everywhere. Since it's so small and minimalistic, it will always be found on most computers, it can be stored on a floppy, and it may be all you have available in the event of a real problem with your system.

Emacs users, for their part, consider vi users troglodytes. After all, using an editor that doesn't do much more than, you know, edit text, is as bad as using flint knives and bearskins.

The other item at the heart of the Emacs enthusiasts' love of Emacs is the fact that the editor is programmable using Lisp. Lisp is a venerable language designed for artificial intelligence applications. "Lisp," its adherents will tell you, "is elegant and powerful." (Like vi, now that I think of it.) Like vi, Lisp has driven better people than you or I to madness, which, by extension, evens Emacs and vi up on the insanity-inducement scale.

A final look at this clash of cultures has to involve the text editors' icons as they appear on the Linux desktop. Emacs has a smiling gnu. The gnu is the mascot of the GNU project, of course, and there's something sort of large and shaggy about Emacs, so it's appropriate. Vi, on the other hand, doesn't have an icon. That would be wasteful. Sometimes, though, it is represented by buttons on Web pages. No point in wasting time figuring out the whole GNU/Emacs/large/shaggy thing, though. They always just say "vi."



Emacs users, for their part, consider vi users troglodytes.

THE DESKTOP WARS: SO MUCH TO FIGHT ABOUT, SO LITTLE TIME

Once upon a time, if you wanted a graphical user interface (GUI) under Linux, you had a limited set of choices, and they were all pretty uncomfortable, especially for someone migrating from Windows or Mac OS.

Some of the early attempts at making Linux "friendlier" were amusing in a half-baked sort of way. One of them made Linux look just enough like Windows 95 (right down to the ugly, teal

wallpaper) that you became sort of homesick for it once you realized you'd been tricked and had to deal with something so un-Windows-like underneath that it seemed to be mocking you.

There are also some very good GUIs that are nothing like anything you've ever seen, which is fun. And there are GUIs that look a lot like things you may have seen at some point or another and enjoyed, but that somehow come up lacking for new users. Enter the "desktop environment."

Desktop environments aren't really very new. In the '80s, the UNIX world lined up behind the "Common Desktop Environment," for instance. Users got a set of applications and tools that all looked and acted much the same. CDE was something of a standard in the UNIX world (and still is in some corners), but it had a few things going against it:

First, it was ugly. No one wants an ugly desktop.

No one wants an ugly desktop.

Second, as businesses will sometimes do when they smell a potential angle, the politics of cooperating with each other over how to best implement CDE slowed it down. To make a long story short, UNIX never did amount to much on the desktops most of us are familiar with, Microsoft rode to glory, and that was that.

So as it became clear that Linux was being taken more and more seriously as a server operating system, people began to look around for the next thing to improve, and everyone agreed that ease of use for newer and non-technical users had to be addressed.

KDE: Better Flamebait through German Engineering

One group of people decided to address the issue via the K Desktop Environment, more commonly known as "KDE." KDE's designers have made it their mission in life to produce software that

makes Linux (and UNIX in general) more palatable for new users. This is partially in response to charges that Linux will never amount to anything as an office operating system without a little more user-friendly polish. No, in case you're wondering, speed and reliability are not factors here. What we're after is making computers more comfortable for brighter-than-average chimpanzees.

So the KDE team set about their work with quiet efficiency, taking bits and pieces of past interfaces and adding bits of their own, questing after the Holy Grail of "usability," which is to say that they wanted to make life easier on any of the chimpanzees feeling left behind during the whole Windows thing.

KDE turned out quite nice, too. Users who downloaded it got a "desktop environment" with all sorts of handy little applications: an address book, an e-mail program, and countless other gimcracks and doo-dads. It even included Solitaire. Admiring users called it "solid" and "smooth," and even "slick," which is high praise, indeed.

Unfortunately, according to some (okay, more than "some," since the whole thing rates coverage as a quarrel of note), KDE had some problems, too.

First, there was an issue with the toolkit upon which KDE was built. Toolkits are a programming thing. They make life easier on programmers by providing a way to write software that looks the same and generally acts the same without having to reprogram all those looks and behaviors into each and every application a programmer writes. If you open a window on your computer's desktop and look at the little buttons and scrollbars and icons the window provides, you're looking at the most visible manifestation of toolkits: the widgets. Those widgets may

What we're after is making computers more comfortable for brighter-than-average chimpanzees.

not look like much, but having a toolkit that defines all those widgets just once and then allows them to be included in programs with much less code than it takes to describe the widget in the first place makes life much easier for programmers.

So the good programmers at KDE, out to make life easier on themselves, picked the Qt toolkit by a company from Norway called TrollTech. They reasoned that Qt was the best toolkit going, and they wanted the best for their software. Qt had a problem, though, in the form of its software license.

Licenses, as you'll recall, are what determine how you can use any software you come across. Most licenses you'll encounter in software you want to run on your Windows or Macintosh computer (and Windows and Mac OS themselves) are licensed in such a way that you can't go around sharing their source code with people, don't have any rights to get at the source code even if you could share it, and may have to give up your firstborn if you make the mistake of coming up with too good an approximation of how their software gets put together and tell anybody else.

TrollTech wasn't that nasty about the whole thing, and they had a license that was fairly friendly to Open Source developers. It had some problems, though, where Free Software and the GNU Copyleft were concerned, because some of its restrictions were a bit too, well, restrictive for it to coexist with Free Software. You may be hankering for more detail than that. After all, Qt's license bears about as much historical weight in the Desktop Wars as the assassin who offed Archduke Franz Ferdinand. Someday, someone will write the book it would take to lay it out meaningfully to you.

As it is, whenever someone writes a "guide to licensing," they're doing it to make a point about their favorite one, and they're usually sniping at the other licenses along the way. We could say something nasty about computer-science types who know enough scholarly language to sound reasonable until well after you realize just

how far they've stuck the knife, but we won't. They gave us Linux, so we forgive them. When the book is written that doesn't involve sniping and back-stabbing, we'll include it in the appendix. If you can find the authors, they may even buy you a copy if you're a little short that day and really need to know more than two years' worth of near-constant flaming across the Internet, meticulously archived everywhere, can provide.

So, to return to the immediate point, Qt wasn't "free" enough (in that sense of free we talked about earlier that has nothing to do with cost) to satisfy everybody.

There was enough of a problem, in fact, that at one point Red Hat announced it wouldn't include it in their distributions. This has been, as the Desktop Wars have played out, attributed to all sorts of sinister motives on Red Hat's part, including a desire to cripple SuSE (which supports KDE by making it their default desktop), nationalism (many of KDE's principal developers are German), and monopolistic intent (which we'll get to).

The second problem, which takes us into yet another battle that predates Linux, and will probably outlive it, was the programming language upon which KDE was to be built: C++. The problem with C++, as any perfectly reasonable person who hates it in favor of say, C, will tell you, is that it isn't C.

"C," a perfectly reasonable person who hates C++ will tell you, "is the language upon which UNIX was built. C is the language of the Linux kernel itself. C is elegant and simple. C is not encumbered by the notions high-foreheaded academicians have about programming languages. C++, on the other hand,

As it is, whenever someone writes a "guide to licensing," they're doing it to make a point about their favorite one, and they're usually sniping at the other licenses along the way.

is the language of ninnies and eggheads with too much time on their hands. C++ is a trainwreck waiting to happen. C++ does not rule.”

At this point, you’ll be asked to perform a simple programming task in C++, presumably because you can’t. The perfectly reasonable person who hates C++ will sit back smugly and laugh. Then your friend who knows more than you will explain how this is actually quite trivial, which will require a disclaimer that there’s something wrong with that approach from your interrogator, which will provoke a nasty jab from your more knowledgeable friend about how C is for Neanderthals. And on and on.

In shorter form: KDE was built on C++. Strike two.

Strike three requires no explanatory text: Some people thought KDE was ugly. These same people are typically quiet

during C/C++ wars, and don’t have very complex opinions about licensing. They’re just very, very concerned that their desktop not look ugly, and they had their say.

C++, on the other hand, is the language of ninnies and eggheads with too much time on their hands.

Now, these problems with KDE would be non-issues if Linux existed in some sort of locked-in state the way Windows does. We all know that the net result of hating anything about

Windows is more of the same the following year, only with a better selection of wallpaper and a new set of things to learn to never do if you don’t want to crash your computer daily. It’s not that the folks at Microsoft heard your complaints and decided to make you suffer for complaining. They’ll make you suffer whether you complain or not.

As we were saying, though, Linux is free in all sorts of ways, including the fact that if you don’t like something, you’re likely

to have the same tools at your disposal as the person who did whatever it is you didn't like.

Enter the GNOME project.

GNOME: Handcrafted With C to Burn out the KDE Heretics

There are several ways to read the beginning of the GNOME project. Some claim it was never motivated by anything other than personal animosity toward KDE's developers. Others buy the "party line," which was that a desktop environment with no licensing problems was required if Linux was to make it as a desktop operating system with its principles intact. Still others just smile and allow that part of the fun of Free Software is the occasional duplication of effort.

There's no denying that GNOME's founder, Miguel de Icaza, had some unpleasant run-ins with KDE developers before deciding to start his own desktop project. There's also no denying that KDE's licensing problems didn't satisfy the strictest members of the Free Software community. And there's no denying that KDE's use of C++ drove some people into a blind rage.

The GNOME desktop environment was announced as an effort to build a "completely free" desktop for Linux. By that, its developers meant that they wouldn't use any toolkits with questionable licenses, or write any software that wasn't completely Free. As a result, they decided to use the GTK toolkit, originally developed for the popular, PhotoShop-like program known as the GIMP.

They also planned to build GNOME on C, which won much applause from everyone bent out of shape over KDE's use of C++. There were some other, more obvious differences to the casual user, as well. Where KDE preferred a somewhat staid look

(reviled by a few as “too Windows-like”), GNOME affected a different appearance (reviled by a few as “too cartoonish”).

GNOME was widely praised throughout the community as a legitimate effort, and widely scorned as a shabby attempt on the part of Miguel to get even with the KDE community for not listening to him. Whatever the motivations, the hackers surrounding GNOME began to hack, and the project slowly took shape.

Red Hat chipped into the effort by housing several GNOME hackers out of the Red Hat Advanced Development Labs, and announced their intent to push GNOME as the de facto Linux desktop, which fanned the flames of those who claimed the distribution makers were out to corner the desktop market by leveraging their considerable name recognition and market share.

As I sit writing this, over two years later, there’s no end in sight to the Desktop Wars.

A few months ago, the GNOME project announced that they were founding the GNOME Foundation. Several major companies like Sun and Compaq got behind the effort, declaring that as far as they were concerned, GNOME was the new standard desktop for not only Linux, but UNIX. Howls went up, as they do whenever companies rear their heads in the Linux world and declare that they’ve seen the light. There was a little irony involved here, too, since the same companies declaring GNOME the Once and Future Desktop had once said the same thing about CDE, and look where it got them.

Not long after the GNOME Foundation was announced, though, TrollTech, the company that makes Qt, the toolkit that caused so many aspersions to be directed against KDE’s licensing purity, decided they’d had enough of the issues their licensing had caused and released their product under the GNU General Public License. This prompted some of the more ardent defenders of KDE to suggest that the GNOME project no longer had any reason to

exist, and ought to just stop all development and call it a day.

Hero of the Free Software Revolution Richard Stallman had something to say about TrollTech's change of license, too, and although he praised them for it, he pointed out a bit of legal esoterica that needed to be dealt with by some of KDE's developers: There were still a few bits

of code floating around here and there that had their licenses infringed during the years KDE was using a less-than-totally free toolkit. These infringements, he pointed out, needed to be forgiven. Worried that the Desktop Wars might almost be over, more howls went up as the word "forgiven" was taken to mean "Apologize for being naughty and defying my will" instead of the more commonly understood "Grant an exception to the license to prevent weakening it if it ever had to be enforced."

You could argue that Mr. Stallman could have chosen his words more carefully, and that he'd shown up at a treaty-signing in a Kaiser Wilhelm helmet brandishing a gun; or you could argue that the Linux community enjoys a good fight and hated the thought of letting one as good as the Desktop Wars go.

DISTRIBUTION WARS: NO, JUST SAYING YOU RUN LINUX ISN'T ENOUGH

So, if you're a religious user of the best editor around and you've got an ethically pure, conceptually superior, attractive desktop, you're still not out of the woods. If you're running Linux, you're running a distribution. Distributions are a way of dealing with what would be a very difficult and messy proposition if there weren't distributions: getting Linux up and running on your own.

Howls went up, as they do whenever companies rear their heads in the Linux world and declare that they've seen the light.

Most of the time, we don't think about what happens when we turn on the computer and wait for it to boot. Linux, of course, isn't the sort of OS to hide behind an attractive screen of clouds on a blue sky—it tells you exactly what it's doing. From loading the kernel to starting up the GUI, you get to read a message about each and every step Linux is taking as it pulls itself up by its bootstraps.

When you send e-mail in Linux, play a CD, compile a program, surf the Web, edit a file (using Emacs, I hope—see how easy it is to pick a fight with just the few tools I've given you already?) or print a picture, you're doing so with one of many, many possible choices and configurations. You could gather up all the software to do these things on your own, and compile it from its source code, and write the configuration files for it, and

hope you got all those hundreds of little programs just so. Or you can just go out and buy, borrow, or download a distribution.

Distributions take the work out of much of running Linux. Their maintainers preselect a collection of applications, do most of the configuration you need to at least boot up and start using your system, and sometimes offer you support for installing and running Linux on your

computer. Some distributions don't cost a dime, others can range up to over \$100.

Because there are so many of them, and because they all have such distinct approaches to presenting the end user with a workable Linux installation, they are naturals for all sorts of good fights.

Here's a quick look at the beef with all of them.

Linux, of course, isn't the sort of OS to hide behind an attractive screen of clouds on a blue sky—it tells you exactly what it's doing.

Red Hat: Easy Installation for the Masses (and That's the Whole Problem...)

Red Hat is probably the best known of all the distributions, thanks to fairly aggressive marketing and what used to be one of the easiest installations going.

That ease of installation, of course, is what earned it early enemies among the angrier elements of the Linux community, many of whom bemoaned the “dumbing down” of Linux because “just anybody” could have a working installation out of the box. Adding insult to injury, Red Hat even had the gall to provide a way to make installing software really, really, easy.

RPM, which stands for “Red Hat Package Manager,” was widely viewed by purists as the worst sort of mollycoddling. Instead of having to build a program from source (which isn't that hard most of the time), users can type a single command to add or remove a program from their computer. RPM isn't the only package manager on the block, but it's the most popular among the more common distributions, and it was a big selling point for Red Hat. People interested in “just running Linux” could be spared one more little chore to try out the newest software.

Some muddleheaded types will try to tell you that RPM, because it seems to work smoothly and allows programs to be packaged pre-compiled, is a “proprietary” program, which raises the ugly specter of a license war, since it would be bad karma indeed if a major Linux distribution were to rely on a closed-source program too much. A quick look at the RPM source code (a tip-off that maybe it isn't that proprietary after all) reveals, though, that it's happily licensed under the GNU General Public License.

Others, who we'll grant their inalienable right to paranoia, insist that Red Hat is out to monopolize Linux and become the next Microsoft. It's hard to disagree with the sentiment that a

Others, who we'll grant their inalienable right to paranoia, insist that Red Hat is out to monopolize Linux and become the next Microsoft.

business's first priority is to make money and grow, but it's even harder to credit the notion that a community of hackers and free-thinkers who all managed to walk away from Microsoft in the first place would quietly line up behind a new monopolist.

Red Hat's popularity, of course, sets them up for all sorts of other problems. With a relatively huge base of new users out and about with their shiny new Red Hat installations and

no sense about how to secure their machines from even rudimentary attacks from the neighborhood script kiddies, there's a lot of squawking about Red Hat (sometimes called "Root Hat" because it's supposedly so easy to "get root" or take control of) and how poor its security is.

Is it that bad? Linux isn't always the most forgiving, no matter which brand name it's being boxed under.

Debian: Live Free or Die, or At Least Don't Run Linux

The Debian Project is another distribution with a distinguished history in the Linux world. Its primary distinction as a community is its insistence on keeping software that isn't free out of its distribution. Many other distributions are happy to include software that isn't free, the Netscape browser being the most notable. Debian, on the other hand, won't. They hedge their bets a little by making Netscape available, of course, but it isn't an "official" part of their distribution, which includes thousands and thousands of programs under the GPL.

Remember the KDE/GNOME flap a section ago? Debian was the last distribution to cave in on that one when it finally admitted

KDE into its distribution after Qt was relicensed. That didn't stop some of Debian's loonier detractors (who hate them for coming off as too consistent with their stated ethics) from accusing them of a nefarious attempt to do a poor packaging job on their release of KDE so as to discredit the project.

And therein lies the source of Debian's special place among Linux distributions: They know what they think is right (free software), and they do what they can to stick to that. So well, in fact, it's hard for some people to deal with rationally.

Debian's other distinction is its nearly glacial production cycle, which is born out of the most meticulous quality-control process going among the major distributions. Where other distributions go through two or three releases, Debian's developers prepare one. The upside to this is that when you install Debian, you're guaranteed a meticulously tested distribution with little that will likely bite you later on. The downside is that you'll celebrate a birthday or two waiting for a new release. Debian's developers are fairly intractable on this point, though, and argue that it's better to have the trailing edge working reliably than the bleeding edge, well, bleeding all over the place.

Debian's other secret weapon is how it manages to get packages of software onto your computer. Through a fairly ingenious tool called "apt-get," Debian's able to download not only a piece of software you want, but every piece of software it might need to work correctly with a single command. The same sort of simplicity is involved with updating any software that may have been changed since you installed it. This appeals to the cardinal virtue all good hackers share: profound laziness. Why painstakingly download package after package when one command does it all?

This appeals to the cardinal virtue all good hackers share: profound laziness.

Periodically a Red Hat or Mandrake or SuSE enthusiast will, in the interests of striking a blow for her favorite distribution, announce a remarkably convoluted Perl script to handle the difficulties RPM presents to anyone with a reasonable splash of laziness. The problem is, no one who wants the ease and simplicity of a packaging system wants to put up with Perl scripts. At this point, I'm supposed to say something very prescient about how if someone doesn't work something out soon, there's going to be a terrible problem. Sadly, having just returned from a convention where no fewer than four companies have announced solutions to take care of this very shortcoming in RPM, I'm reduced to shouting "I told you so!" at the walls.

Slackware: On the Internet, Nobody Knows You're Borrowing Big Sis' Computer

The first distribution I ever kept on a machine for more than a week was Slackware. Because it doesn't exactly have a package-management system like Red Hat or Debian, Slackware is often thought of as the "hard distribution." The fact that I managed to make it work with little more than a book and no connection

to the Internet during my extensive period as the most rank of Linux newbies suggests that this reputation isn't deserved.

Slackware advocates can be pared down into two general categories: good ones and bad ones. Slackware is well-loved by the more "do-it-yourself" element of the Linux commu-

nity in large part because its maintainers haven't made an attempt to hold anybody's hand during the configuration of a freshly installed system. People who like having to do things by hand in

For some people, the level of difficulty they induce on themselves is a bragging point.

the name of finer control over every element of their system are usually too busy to contribute to flame wars, and they're the salt-of-the-earth types, anyhow. When you think "Good Slackware Advocate," think, "too busy working to yammer."

As a result of the difficulties it sometimes poses, Slackware also enjoys an unfortunately sterling reputation among the sort of people who talk about "their boxes" as if they run server farms for Fortune 500 companies when what they really mean is the dual-booting Windows machine their parents let them use when their big sister's away at college. For some people, the level of difficulty they induce on themselves is a bragging point. The "bad" Slackware advocate turns up in any flame fest over distributions convinced that mere mention of his favorite distribution will silence the combatants once they realize how indescribably lame they must be in comparison to our intrepid masochist. When you think "Bad Slackware Advocate," think, "too busy yammering to work." Or think of the kid next door with really baggy pants who makes every word that ends with "s" sound like it's ending with "z," as in "Slackware rulez."

Mandrake: Taking It from Both Sides

At the other end of the spectrum from Slackware is Mandrake. Mandrake was, once upon a time, not much more than a finely tuned version of Red Hat with more goodies, nicer default configurations, and the nifty fact that it was built to accommodate Pentium-based computers specifically. Over the years, Mandrake has slowly grown out of its reputation as a Red Hat taillight-chaser and become its own distribution, and it has a very loyal following.

For the most part, Mandrake users aren't the ones starting fights. They just like being able to buy the latest copy down at the local Wal*Mart (Mandrake is very aggressively marketed where many other distributions fear to tread) and knowing they'll be

getting a fairly well-tuned product out of the box. Lots of people, however, like to start fights with Mandrake users:

“How good can it be if it’s based on Root Hat?” they ask.

“How good can it be if you can buy it at Wal**Mart*?!” they jeer.

“How good can it be if you can buy it at Wal**Mart*?!” they jeer.

“Pentium optimizations introduce only marginal gains in performance that most users will experience only as figments of their overactive imaginations!” they charge.

The average Mandrake user will point out that it really isn’t based on Red Hat so much anymore, and that just because Wal**Mart* sells a lot of copies of “Big Bass Fishin’” alongside “Linux Mandrake Deluxe,” it doesn’t make everything they sell bad, and that they used to think the same thing about Pentium optimizations until they tried them out.

So who’s right?

On the one hand, you have the Red Hat users who enjoy the luxury of being right if you put it to a vote. On the other, you have the Debian users who know that every element of their favorite distribution probably *has* been put to a vote. Then you’ve got the Slackware fans who argue that, like Samurai of old, they’ve forged their installation in the furnace of their own misery, and the Mandrake fans who argue that it’s sheerest lunacy to install something that isn’t pretty good out of the box.

Well, they’re all right. And therein lies another bit of the Joy of Linux: Like some bizarre amalgam of Disneyland and Las Vegas, it has a little something for every taste. The only people who aren’t right, and this goes for just about everything you’ve read about in this chapter, are the people who don’t, in the end, admit that the best judge of which tool is best for you is you.