THE LXF INTERVIEW

Hack daddy

He has the beard and the attitude to match – Alan Cox is the archetypal kernel hacker, and he's here to talk to you.

His computing career started with a ZX81, a RAM pack and a lot of curiosity. It has led Alan Cox, through various stints at games developers, ISPs and studies at Aberystwyth and Swansea, to be a Red Hat fellow – and probably the second-most revered kernel hacker in the world. He's currently finishing his thesis (on desktop Linux) for a Masters degree. *Linux Format* decided to probe the inveterate tinkerer on steam trains, Japanese contributors – oh, and a version control tool called *BitKeeper*.

Linux Format: **You've been at Red Hat for five years now. What is your present role?**

Alan Cox: I do a mix of things. I do a lot of work in the community, where Red Hat basically says, "Work on whatever you think is needed." We try to take feedback from all places. If you just look at the community you often don't see what a lot of people need, because the community's only a fraction of the user base these days.

I also work on some of the kernel projects where particular expertise is needed, helping out when a customer has a particularly difficult problem. When you're selling high-end support services you have to have people who can actually deliver – not just fix the easy problems, but when something really horrendous happens that requires real expertise.

LXF: **Do you find that your academic work conflicts with your work on Linux?**

AC: Well, Red Hat was very good when I was doing the Masters. I was only working part time, and reasonably flexibly, because the Masters work tended to get more and more frenetic as it came up to each exam and then eased again. The thesis has been partly delayed by Red Hat Enterprise Linux 4, but now that RHEL 4 is out the door Red Hat has been very reasonable about the fact that I'm spending a lot more time on my thesis than on Red Hat products.

LXF: **Have you been surprised by anything you found out during your research?**

AC: I've been surprised by quite a few things. The initial idea was to do some interviews, talk to people and then do some quantitative surveys of organisations about things like Gnome and KDE. And it turns out that the majority of the business community using Linux have no idea that KDE and Gnome even exist. They don't know, and in general they don't care – that was quite a surprise.

Even though I asked them about desktop configuration, the one thing they would immediately talk about was the fact that the SUSE, Red Hat and Mandriva configuration tools are all different. For an in-house person who they've trained up on one of [those distros], although in theory they can mix and match and switch, it's actually very hard, particularly for people used to graphical management systems, because they're all different. And that was what was annoying businesses, not the desktop.

LXF: **That's quite valid. In a lot of cases, when the tools try to abstract the functionality of underlying commands and make it simple they actually make it more complex. I've certainly found that I've been trying to reconfigure boxes using what I consider the proper way only to discover that the config files are being automatically overwritten.**

AC: That's one thing we were very careful to avoid with the Red Hat tools. We knew about that trap because IBM had that problem with the IBM management system, which they called SMIT. There are people who wander around even today with T-shirts saying "SMIT happens". There are times when being able to use the skills to hit the config files directly and diagnose a problem is really important.

LXF: **At all the events I've been to recently, people have been going on about desktop Linux: is this its time, is it really happening now? It reminds me of *The Hitchhiker's Guide To The Galaxy* – I think you need to define the question of who is using it more fully if you want a sensible answer.**

AC: I think that's the case. Linux has been used on the technical desktop for a very long time. There are other areas where Linux is also used, some of them quite surprising, like animation and military applications.

And from the people I've been talking to there is now a large and I think very rapidly growing number of people who are using Linux in locked-down environments: call centres, hotels, the kind of environments where they want to make sure that the users can't install *Doom* or *Quake* on the system, and that when they come back in six months' time it has no viruses on it and it looks the same as the system they left.

Another interesting thing I found out: almost everybody I talked to in my research who had switched and

"I'D LIKE TO REWRITE THE TTY LAYER IN THE KERNEL. IT'S A BIG JOB."

started using a Linux desktop was surprised at how far it had come. So it seems that the desktop is actually better now than people's perceptions of it, which is a good thing.

LXF: **Does it surprise you how Linux has taken off in so many different sectors of computing, everything from embedded devices to mainframes?**

AC: That's really the nature of the development, because everybody who comes along wants to make Linux do something new. So each person comes along and takes something which is roughly what they need and fine-tune it.

And as they're tuning it, other people are saying, "Well, that change you've made there doesn't work for embedded", for example. So the mainframe people say, "Well, this gives us better throughput," and the embedded people go, "Oh my God, but my performance isn't predictable any more." So you've got this continuous cycle where each person is refining it for their own uses.

Now what would happen in the proprietary world, of course, is that each person writes their own operating system. So it's a property of open source development that you've got something that you can run on your Palm Pilot and on a 20-tonne mainframe, and pretty much anything in between.

LXF: **Which technologies in computing are exciting to you?**

AC: I think that virtualisation is one of the big ones. When you deal with the mainframe people for a while you start to realise just how useful virtualisation is. And you talk to people whose idea of an operating system upgrade is to install the new version and migrate services across while the old one is still running. There's also a fantastic amount of security work being done with *Xen*, which is something perhaps people know less about. And ideas like, if somebody launches a Denial Of Service attack on your web server, you migrate it to a different IP address somewhere else in the world or you just create ten of them, or a hundred of them, temporarily, then get rid of them later. And that's what you can do very fast with *Xen*.

There's other security work with *Xen*, because it's very small, which means it's much easier to verify that *Xen* is secure and then to trust *Xen* to separate operating system instances. And that will be incredibly important for things like cheap web hosting. At the moment cheap web hosting means you don't get a root account; you don't manage your own machine. With *Xen*, you have the ability to give everybody their own virtual machine, which is an entire Linux system for them to break, or whatever else it is they do with it, without harming other customers, and even to do things like migrate them from hardware to load balance, or when you're taking physical equipment out to servers.

LXF: **Talking about hardware, the trusted computing platform seems to be an idea that never goes away. What do you think the implications are for Linux?**

AC: Well, trusted computing platforms are already out there – the classic example is the XBox. You buy a PC-class computer, and Microsoft controls which software you run on it. That's the negative side of it. You know, someone sells you a product and then is able to say, "But you can't do this with it."

There is a positive side to that kind of trusted computing system as well, though, because it becomes possible to deploy a system in an environment and be sure it's not being compromised. It gives you a place to store things like credit card data and keys, which is probably a great deal safer than on current systems.

It all really comes down to one thing, which is who controls the cryptographic keys that decide what may and may not run on the system. If

« the user controls those then it could still be abused, but the potential is there to do very useful things with it.

LXF: **Do you think there's a possibility, though, that if business interests become the dominant force in adoption of architectures then you'll see banks, for example, only operating online services with authorised trusted platforms?**

AC: I think the media companies are the ones who are more likely to do that, where you can only play a film with their software on their trusted platform. They're already doing this kind of thing in a limited way with cryptographic techniques on Windows and things like the *Windows Media*-encrypted formats, and it's down to what the customer will accept.

What ultimately has to limit it and what someone has to get right is the legal framework, which has to be able to stop organisations holding others to ransom. For example, if you're dealing with licence keys then all of a sudden the company you depend on for software gains this magical ability to just turn all the software off. In that case, if you get in a legal dispute with them, all of the existing rules and precedents and procedures in court have gone out the window. Because they turned your system off, you have to sue them.

But that may just be something that encourages free software. Because once you've got heavy-handed tactics being used, one of the obvious answers is to include the cost of software auditing in your TCO calculations, look horrified at the numbers, and find a cheaper solution.

LXF: **There seem to be a lot of issues arising over intellectual property at the moment – the issue of patents in Europe being one example. Do you think that IP questions could be raised against Linux?**

AC: I think there are questions to be raised against all sorts of software, proprietary and free. You may have come across **gpl-violations.org**. That was produced to deal with very large software theft. My experience is that the proprietary software industry is a lot less clean than the free software industry because they think you can't see what they're up to. It's very hard to track software licensing violations, particularly of proprietary software, because in theory nobody has seen the other version so nobody can tell.

At least in free software you can look at the application, you can types lines of the code into things like Google and see if they turn up elsewhere. So you've got a chance to follow what happened to that code. And we've pretty good processes for doing that now in Linux.

Patents are much more of a problem. The patents system is essentially a gambling machine for people with no morals – if you file enough dodgy lawsuits, eventually you'll win, and you'll win so much money that it's worth playing the game. So that's a fundamental problem with the patents system, but it's not a problem with the idea of patents themselves, it's a problem with the implementation.

When you come to software you've got all sorts of other problems, because software is a literary work. A long time ago there was an argument about whether software is a machine or a literary work. You can't copyright a machine; you can patent ideas with it, but everybody else can build the machine so long as they've got patent licences. They can look at your machine and say, "I can see how to do this without the patent, or after your patent has expired."

The decision at the time, which is actually written into things like WIPO [the World Intellectual Property Organisation], was that software is a literary work. So patents don't apply to literary works, at least until the Americans got involved, and if you try and apply patents to literary work all sorts of things start to go very, very pear-shaped in the legal framework, because the author of a literary work also has various other protections under WIPO that appear to conflict with software being patentable. Because a third party has no right to extract money essentially from publication of a literary work, and that was something that was done to ensure that governments couldn't bring all sorts of interesting tariffs. It all gets really messy.

LXF: **And the current agreements over patents worldwide means you get companies such as Microsoft being granted various software patents in New Zealand, where they know no one is going to look very carefully...**

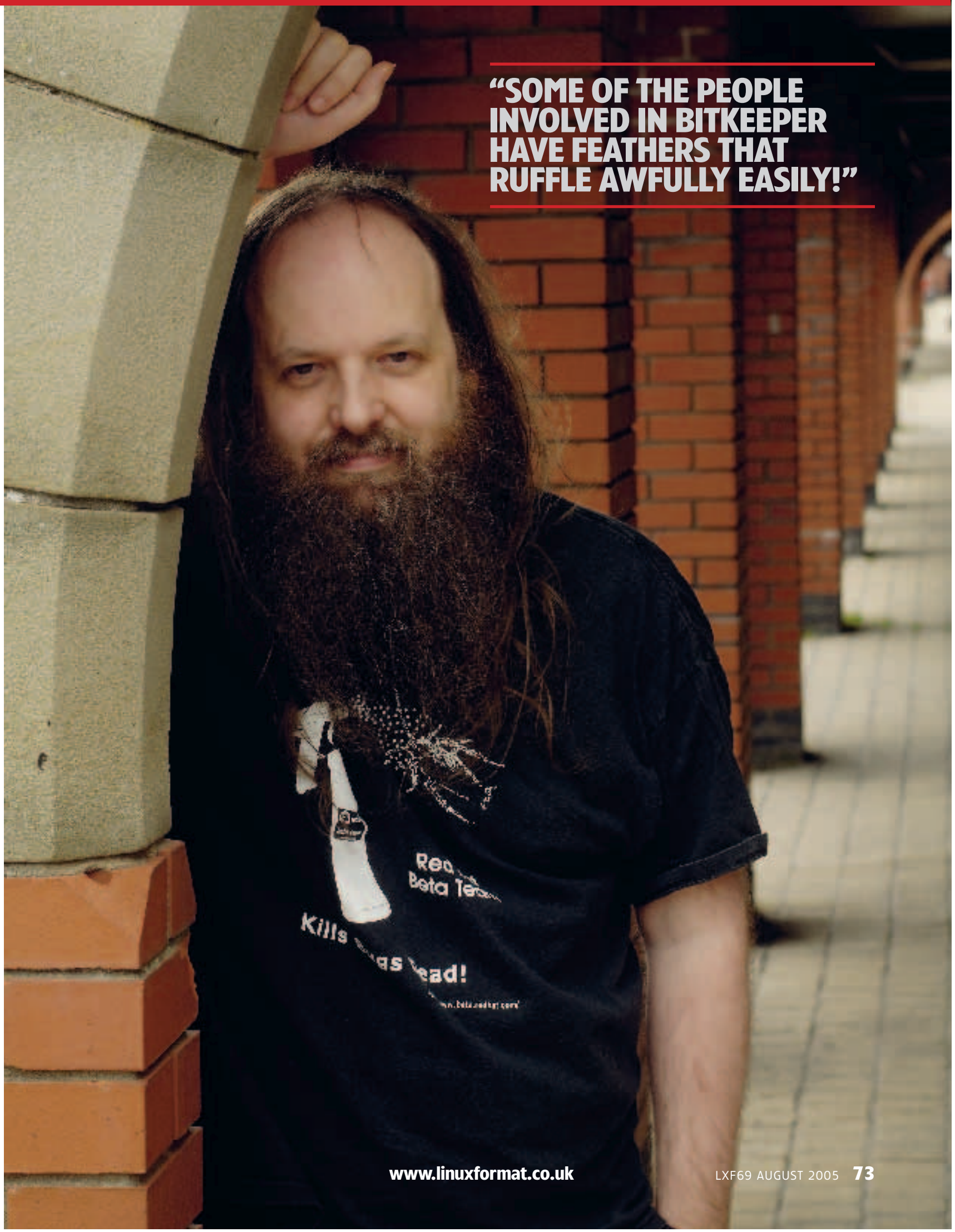
AC: Yes. That's not a software thing. People have been playing games like that for hundreds of years. The whole history of the steam engine was held up because the original creators of the steam engine thought high-pressure steam was a dangerous evil and sort of refused to grant rights to their patents to any of the high-pressure steam people. High-pressure steam was the future, as it turns out, but it was held up for almost 20 years.

The same has happened with IP version 6. You notice that everyone is saying IP version 6 is this, is that, and there's all this research software up there. No one at Cisco is releasing big IPv6 routers. Not because there's no market demand, but because they want 20 years to have elapsed from the publication of the standard before the product comes out – because they know that there will be hundreds of people who've had guesses at where the standard would go and filed patents around it. And it's easier to let things lapse for 20 years than fight the system.

LXF: **You say you've got a fairly free rein to work on what you want. Do you find it difficult to keep yourself disciplined to finish certain projects?**

AC: What often tends to happen in the free software world is that you get a project to a certain state, and it works, it's usable, it's in the mainstream kernel, and then somebody else will come along and start contributing to it who is simply better at that subject than you are. And the best thing you can do is go and find some other project. That happened with things like the networking code, it happened with the multiprocessing... it was great. It wasn't my problem any more. And they »

TOP TRUMPS COURTESY: WINNING MOVES UK LTD



"SOME OF THE PEOPLE INVOLVED IN BITKEEPER HAVE FEATHERS THAT RUFFLE AWFULLY EASILY!"

« worked a lot better than if I had been left in charge of them.

A lot of what I tend to do is cleaning up, fixing and making work really old, horrible grungy code. It was one of the jobs I did in the proprietary software world, which meant I was greatly in demand, and it's one thing I do in the free software world, and I think it's a very important part of my daily job.

LXF: **What are you messing around with at the moment?**
AC: Mostly my thesis!

LXF: **Didn't I read somewhere that you were getting into the *X Windows* and *X.org* stuff?**
AC: I've done little bits – I wrote an accelerator driver for the old voodoo 2 cards, I did a little bit of work on the direct rendering, the 3D stuff, but again, other people with more time and better skills took over from that. The VIA 3D driver is maintained by Thomas Hellström and others, who are doing a far better job that I would ever have time to do.

Somebody will write a piece of software that solves a problem – solves their problem – and then other people will come along and say, "Well, it works for version 1 of the hardware I've got version 2 for: I'll fix this." And somebody else will come along and say, "It doesn't support this feature of the card that I need." So it gradually grows. But if you went to someone with a list of specifications for a Linux operating system – a list of platforms, memory sizes, performance requirements – you could go to some of the biggest software companies in the world and they'd laugh at you and show you the door, because none of them would believe it could be built.

LXF: **By traditional methods I suppose it could *never* have been built. What's your feeling about Extreme Programming – agile programming as it's now been rebranded? Do you think there is any merit in any of these ideas?**
AC: I'm not greatly into buzzwords. The ideas have been around a very long time. The biggest problem I think we have isn't about all these things like agile programming. The biggest problem we have is that nobody understands why there could be a 30-fold or higher difference in the productivity of two programmers. If we could understand why some people are 30 times more productive we would be well on the way. That's what someone has to figure out. Why are some programmers so much better than the average? And having said that, how do we teach everybody else the skills involved?

LXF: **What's your personal take on that? Do you think great programmers are born great?**
AC: There are one or two things that do appear to be noticeable. One clear one is that almost all the people who are really, really good programmers start young. That's the one that stands out above anything else. And that's sort of interesting, because there's a lot of links between programming and ideas of language. And we know that language learning, as opposed to spoken language learning, is something children do very much better when young. So whether there is a correlation there I don't know.

LXF: **I'll have to get my son started on C++.**
AC: If he's in Europe, patent lawyer might be a more useful occupation!

LXF: **One thing that we do have to ask you about: *BitKeeper*. Anything you want to say?**
AC: I think everything I've predicted from the start has come true! As it happens it's worked out quite well; because it's not taken very long to get tools which are open source, and do everything that *BitKeeper* did but that the open source community needs. And now we can improve them to make them do the things we need. It was sort of predictable.

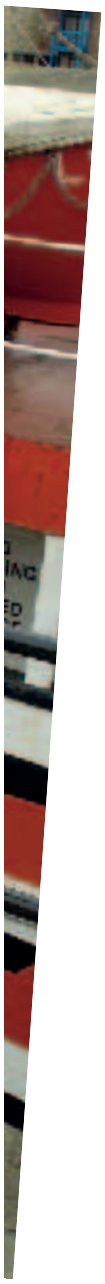
LXF: **Do you think all the ruffled feathers will eventually be smoothed down?**
AC: I don't know – I mean, some of the people involved do have feathers that ruffle awfully easily! I think it's going to be one of those things that some of the people involved just choose never to talk about.

LXF: **It's funny – as someone who is really up there with the big personalities you never seem to come into conflict with anybody. Is that just down to your likeable nature, or...**
AC: I do fall out with people sometimes. But it's normally better to at least try to constructively disagree with someone! Because if you can both walk away and say, "Well, I think your opinion on that is wrong, but whatever," and you have to work with that person again at some point in the future, it's a lot easier than if you've had a screaming match and called each other names.

Some of what you get in open source is also down to culture. People from certain cultures will be very blunt, and people from other cultures don't respect that bluntness. And there are different things that really upset certain people. I've had code from someone in Japan which I was absolutely sure was total garbage but what you don't do to that person is publicly say on the kernel lists "this is crap", which is what they do to a lot of people. Instead I mailed him back with a list of things I thought were problems. I said, "This is an interesting prototype, I look forward to receiving the real thing."

LXF: **And did you ever hear from him again?**
AC: Yeah, we got more code from him. You have be constructively positive when you want to say no to something. So instead of saying "that will never work", say "that's an interesting idea, but how will you deal with such and such", or whatever the problem is.

LXF: **The current 2.6 kernel: there seems to be a lot of work in progress in that stream, or a lot of stuff that maybe in previous iterations would have been in the development kernel rather than in the main stream.**
AC: That was a decision that was taken at the kernel summit. And for



the majority of people, it's better that stuff goes in smaller development cycles. So you get one significant change at a time. Look at Andrew Morton's kernel testing – things fall through into the main kernel. The people who really benefited from the stable kernel series, so to speak, were the big enterprise users, and they tended to buy software that shipped with a version of the kernel that was supported for seven years, so they didn't actually need those guarantees anyway. I don't think it's a big problem. The people who want, for example, that kind of enterprise kernel can grab the source of the RHEL kernel if they want. Or there are various people who actually rebuild all of Enterprise Linux but without the brand support or the services, which is what we are actually selling.

So you grab the CDs of enterprise-stable kernels and do what you want. I don't think it's harmed anybody. It's certainly helped getting things into the kernel more effectively and, I think, a lot of the debugging, because you're no longer wondering, "Is this bug new, or was it introduced by some other random change nine months ago?"

LXF: **We're still dismayed by the poor state of documentation of most of open source software. What can be done about that?**
AC: Well, you need to persuade more people that writing documentation is cool. We simply don't have people who enjoy doing that as part of the software community yet. There are clearly lots of people who enjoy writing documentation; documenting things accurately and precisely. Just look at Wikipedia: it's a huge community around documenting.

It's getting better in some ways because obviously the big vendors are getting requests from their customers saying, "We've got this Linux thing, it's great, it's wonderful and it's faster than their old system, but when we used to run Solaris the manuals were brilliant." And so they're banging on the building saying, "Look, we love your product, but we want the manuals first. You know, the lack of documentation is costing us money and causing us problems; we don't like that."

So there is a commercial pressure as well, which I think will help improve the documentation.

LXF: **So can we see a Red Hat initiative to document everything?**
AC: I don't think anyone's going to do a grand initiative. I think the contributions to companies are growing. There's also the other argument I've heard some people put forward, which is that writing documentation appears to be something the community is bad at, but that's fine, because there are companies who like writing books about products and do a very good job of it.

LXF: **Yes, I'm sure Tim O'Reilly is very happy with the status quo...**
AC: And that may be the answer, if there are different ways of doing things that work better.

LXF: **You started doing your blog in Welsh –**
AC: It's not a blog, it's a diary.

"IF WE COULD UNDERSTAND WHY SOME PROGRAMMERS ARE 30 TIMES MORE PRODUCTIVE WE'D BE WELL ON THE WAY."

LXF: **OK, it's a diary. You pre-dated blogs with your diary.**
AC: I started doing it in Welsh for the practice as much as anything else. I haven't updated it for a long time because I've been doing the Masters, so I shall probably get round to updating it when I've handed in my thesis.

LXF: **As the various translation sites don't have a Translate From Welsh option, doesn't it rather restrict the audience a bit?**
AC: A little. It encourages some of them to learn!

LXF: **You'd better tell us about this train set business. [Alan has, rather (in)famously, begun to fill his house with model railway parts bought from eBay.]**
AC: It's a project to use Linux to control a model railway. That's the whole story! I don't think it's going to be doing very advanced stuff for a very long time. But we wondered how cheaply you could control a railway from a PC, just using cheap model robot components, rather than the fancy, designed-for-the-job stuff. As it turns out it comes in at about £50.

LXF: What do you think you'll do after you finish your thesis?
AC: Get very, very drunk! That's fairly simple. No, I don't know, it'll give me time to get back into some of the projects. I'd like to rewrite the *tty* layer in the kernel because Ted T'so wrote it back around the 1.0, 1.1 kernel era, and so it was designed before multiprocessing, before people would have very high-speed serial links on Linux or very real-time serial and there is a lot in it which kind of works, but it's one of those things where you say, "We really ought to fix that at some stage." It's a big job – it's a hard job – because you're taking something that works. It might need fixing but it's good enough to do the job.

LXF: **What motivates you to keep working on Linux?**
AC: It's fun, basically. It's an interesting job, there's a lot of interesting people, lots of strange things going on. There are lots of different things involved in the job. There's always something new. It's not just the code: it's the people, technology, politics...

LXF: **Can you ever see yourself getting bored?**
AC: I don't know. I'd be surprised. Who knows what I'll be doing in ten years' time? I mean, I started in the gaming world, particularly adventure gaming and multiple users games, and got accidentally diverted into Linux – so presumably something else will divert me again somewhere else. LXF