

ALAN COX

## NUT-CRACKER

Linux Magazine talked to him about his views on kernel development and what might be in store in the future



**Linux Magazine** What drives you to do all this good work?

**Alan Cox** I enjoy it!

**LM** How do you find the time to do it all?

**AC** Large amounts of sleep. I work US hours so sometimes I don't get up until midday.

**LM** What do you consider to be the most vital pieces of software that are missing from Linux?

**AC** Better calendaring software, certain groupware programs. The big ones are now starting to fall into place – office suites like StarOffice and the KOffice work. Especially with StarOffice going to OpenOffice. The Ximian guys are working with Evolution to create a complete clone of Outlook with all the features and then some. A pure mailer program is Sylpheed – It means “Wind Spirit” in Japanese.

**LM** Are the software support models for companies correctly set up?

**AC** Support costs real money. You can pay large amounts of money for complete enterprise-wide support or just the back-end part. It depends on what you negotiate. All the support is there in theory. The Red Hat guys think they are doing a wonderful job but you should really ask the customers.

**LM** Development of the kernel does not use the CVS model. Why not?

**AC** The kernel proper does not use CVS but some developers use some for their parts. The big problem with CVS is that it is not a good way for a single person to have an overview of everything going in and the right kind of quality control and auditing that we require. I believe that Linus is using CVS as he wants to see everything in his CVS tree.

**LM** How does the kernel grow and develop?

**AC** A directed explosion is the best answer. Development goes off at all kinds of angles from a huge number of people for a large number of reasons. Sometimes it is because they see a financial advantage – if we pay someone to do this, then we can sell that. Other times people do it because they like a piece of hardware or they have bought a USB device that is not supported and think that “this is cool and I want to use it under Linux”. Some people do it out of academic interest, researching a given topic to improve a piece of software.

**LM** Does it come about that features are left hanging because everyone wants the feature

but no one gets around to doing it? How is this co-ordinated?

**AC** It's not coordinated. It does sometimes happen, but eventually it irritates someone enough that they fix the problem. It is not uncommon that we have a piece of hardware that someone has written a driver for, but it is not really being maintained or the person who wrote it did not care about certain machines that it did not work on. If there are enough people using it then someone will sit down and say “okay I need to fix this” and then do it. Sometimes it's the vendors, because when they run their QA test there is a problem. Often it is the end users. For example, those people with very old Soundblaster CD drives, if we break something then they still fix the driver.

**LM** Do you get much in the way of requests from users who are unable to contribute directly to the kernel development?

**AC** We certainly get feedback saying it would be nice if Linux did this or if Linux did that. The vendors are very good at getting feedback - “we would install five hundred machines but...”. It can be very useful.

**LM** Do you think that there are good lines of communication from end users to coders?

**AC** Yes. To the extent that what matters is that the end user is prepared to do the work or that they are prepared to pay someone to do the work, and that is how things come around.

**LM** With the continued growth of Linux, do you see any downsides?

**AC** Having a larger mailbox is the biggest potential downside. In the early days the Linux kernel would get two or three messages a day. It has continued to grow with more people becoming involved and more happening in the kernel. I do not know how much longer that will continue to happen - essentially we are running out of things that are important to add to the kernel. Most of the really critical things are now in user space. For example, the world of KDE and GNOME, 3D graphics and all those kind of things.

**LM** What is in the next version of the Linux kernel? Is that the type of question that's even polite to ask?

**AC** It's the type of question we don't know the answer to! There are things people are contributing which look like obvious candidates: An improved input layer; the ability to have multiple monitors and

multiple consoles used more sensibly; plug two mice, two keyboards and two monitors into one Linux box; various pieces of filesystem stuff – XFS, JFS. Compaq are donating clustering code, which is a very interesting and exciting area. You would be able to have a room full of Linux boxes acting as one system, but on top of that, if you lose specific machines then the system continues without a glitch. This is very important for a lot of business applications.

**LM Filesystems, then. Do you have a favourite filesystem?**

**AC** Journaled filesystems are useful for lots of applications. It really depends on what kind of thing we're trying to do. So, we have things like ext2 which migrated into ext3 – a journaled filesystem, which does exactly what every standard generic filesystems has done over a period of time. Reiser FS has done a lot of work on small files, on faster directory handling. IBM's JFS is looking extremely fast – it is interesting for that. We will see, I think, over time, which one will work out for the best. It's a bit muddy at the moment for some of them.

**LM In what way will that muddy water clear? Who will be the victor?**

**AC** I don't actually think there will be a victor. Before, we had competing filesystems – ext fs, xiafs – that was a long time ago, then ext2 came along. Pretty much everyone ended up running the same filesystem after six to nine months, simply because it was the natural one to use. It was the one everyone else used and it seemed to work. I think the vendors will ship the filesystems that work. They do a lot of QA testing on that. In some cases we have specialist filesystems: JFFS for Flash devices is very useful for the iPAQ, but completely useless for your average hard disk user.

**LM What hardware do you recommend and do you get involved with hardware concerns?**

**AC** I try to pick devices with free drivers which work. Like everyone else, I'm trying to build myself a machine that works at a sensible speed or uses as little power of whatever. I play with a fair amount of stuff. I build machines that are all Athlon because that gives me the best build performance. For desktop machines I've been playing with the new VIA C3 chip – it doesn't need a fan and is so much quieter, but it's not a speed demon. I've not really played with the Pentium 4, some benchmarks but that's as far as I've got. It's the first generation of the chip and I think the real question is not how the P4 performs now but in twelve months time.

**LM The support from hardware manufacturers is growing but do you think anything can be done to encourage them more?**

**AC** Most of them make the openness decision based on business risk, or financial reward. So, for example, a lot of small USB vendors have no secrets. Some

hardware vendors like SIS, who are working very closely with the Linux community, are keen to give good support. We also have people like nVidia who are more worried about not giving away secrets to rivals. They are worried that if they release their software technology then someone is going to use it and make the ATI Radeon run even faster than their card. You talk to these people and sometimes it makes sense, sometimes it doesn't, but at least you understand their point of view.

The other thing that has really helped is when people like Compaq and particularly Dell and IBM get involved because when they are building a server they think Linux is going to be one of the supported systems. It means that they go to the hardware vendor they buy from and say "If you have Linux support for this board then we will buy them in large numbers and sell them in our servers". For many more conservative businesses the fact that you can actually say: "Well, if we will do this piece of work for this approximate cost, we will get this money back". It obviously makes it easier for them. Many of them are generally uneasy about giving out documentation to you, as they are more used to a traditional business relationship.

**LM We were talking to someone at the Systems show, they offer the service of a manager/coder interface. Is this common?**

**AC** It's one of the jobs that LinuxCare have been doing. It is to a certain extent, part of what Red Hat offer in doing direct device driver work. We will write you a driver, we will help you commit it to the mainstream kernel, if that's the way you want to go, and we will convert the things the techies say to the things the management need to hear about "time scales and pricing". It's no good asking your average programmer, "How long will this driver take?", the answer is always far too short, so the project will overrun, and they are often far too vague.

**LM Do you use a desktop or are you a console man?**

**AC** Well, I mostly use X, I tend to use XFce for my desktop most of the time, then run mostly GNOME applications, sometimes KDE. You can set both GNOME and KDE up to look the same. In the desktop world, I am very much an end user and as far as I am concerned it all looks the same. OK, some bits are QT, some bits are GTK, but, who cares?

**LM You have a very good line of communication to the community, tell us more about your diary.**

**AC** Well, the diary was originally set up pretty much for that purpose because when I joined Red Hat one of the things Eric wanted, as my then manager, was a monthly summary of what I was doing. So I figured out seeing as this was open source the monthly summary probably ought to be, so that became the diary.

It's no good asking your average programmer, "How long will this driver take?", the answer is always far too short