

# “Two years in the trenches”

## Evolution of a free software project

Greg Lehey  
President, AUUG Inc.  
Greg.Lehey@auug.org.au  
Member, FreeBSD Core team  
grog@FreeBSD.org

### *ABSTRACT*

Free software projects have been around for over ten years now, and they've become an important part of the UNIX community. With the example of the FreeBSD project, this paper illustrates some of the changes that have occurred in that time and attempts to guess what the future may bring.

## **The three ages of UNIX**

UNIX® is now a third of a century old. In this time, the face of computing has changed dramatically. It's interesting to consider the evolution of UNIX in three phases of roughly 11 years each:

- From 1969 until about 1980, UNIX was mainly a research project, little known outside AT&T except at some universities.
- From 1980 until about 1991, UNIX developed into a commercial operating system with releases like UNIX System V (remember that?), XENIX, SunOS, Ultrix and friends.
- In the early 90s, efforts at the University of California in Berkeley to produce free UNIX came to fruition. At pretty much exactly the same time, the Linux project started. UNIX was on the way to becoming free.

This paper looks at the last third of this history in more detail.

## **The evolution of free UNIX**

Free software has been around for ever. Until the end of the 60s, coincidentally the time when UNIX evolved, software was almost always free. And why not? Most system software came with the machines and would only run on them, so there was little incentive to charge separately for the software. Applications software was tuned to specific applications, so piracy was hardly worth worrying about.

By the end of the 1960s, that had changed. IBM was faced with competitors who built hardware which was strikingly compatible with the System/360, and IBM's software ran on it. It started to make commercial sense for IBM to unbundle its software. Over the course of the following ten years, more and more compatible hardware became available, and by 1980 just about all software cost money. Vendors had a perception that access to the source code would give their competitors an advantage, so they restricted availability of their software to object form only. The resultant inflexibility was one of the reasons which led to the formation of the Free Software Foundation in the mid-80s.

Towards the end of the 70s, the price of “real”

computers dropped to the point where individuals could own them. People started sharing programs for their CP/M based machines, and later for their Microsoft-based replacements. Most software for small machines continued to cost money, however, and the machines themselves were too small to run UNIX effectively.

This changed round about 1990 with the availability of machines based on the Intel 80386 processor. At the same time, individual access to the Internet became easier, especially for students. These were the background for a number of free software projects which were initially completely independent of the Free Software Foundation and its values.

## The early 90s

Computer hackers<sup>1</sup> have been around as long as computers. They have probably dreamed of having their own computer that long, too, but it really only became even remotely possible in the mid-1970s. People started working on UNIX-like systems early:

- In the late 1970s, a company called Electro-labs started working on a UNIX-like system called OS-2, intended to run on the Z-80 processor. It never passed the beta stage running under CP/M.
- In the early 1980s, Mark Williams Company ported their Coherent system, a clone of the UNIX Seventh edition, to the IBM PC. Without memory protection, and with the memory limitations of the original success, it was not a great success.
- Andy Tanenbaum's Minix operating system ran on a number of processors, including the Intel 8086 family.

None of these projects became very large. The hardware wasn't up to it, and none of them made the source code freely available, though the restrictions on Minix were relatively minor. Two things changed that situation in the late 1980s and early 1990s:

- The release of the Intel 80386 processor and systems based on it, gave the average hacker an affordable machine with virtual memory, capable of running modern UNIX without significant compromises.
- Improved access to the Internet, especially for students, made cooperation on software projects more practicable.

In those days, for most people the attraction was the challenge of running UNIX on one's own computer, not any commercial intent. Indeed, Bill Jolitz staunchly refused to commercialize his 386BSD operating system, and in August 1991, Linus Torvalds wrote in the original announcement of Linux "just a hobby, won't be big and professional like gnu".

I don't believe that religious belief in "free software" was a big thing at the time. The GNU project had been around for a while, and both 386 BSD and Linux took advantage of the software, but initially there was little synergy between the GNU project and the free OS projects.

During this period, relatively few people contributed to the projects. For example, the release notes of FreeBSD 2.0, released in January 1995, mention a total of 55 contributors, including a 15 member core team, who really *were* the project. Most of the other contributors had only loose links with the FreeBSD project.

## The compromise OS

Along with the free operating system projects there was also a "reasonable" operating system project, BSD/386 from Berkeley Software Design Inc., or BSDI (later BSDi). BSD/386 was derived from the same code base as 386BSD, and was in fact quite closely related. The story of that relationship is interesting enough to be the topic for another paper. The difference was that BSD/386 cost money. Including complete source code, it cost \$1,000 US. That may seem to be a lot of money, but in those days source licenses for System V cost at least \$50,000, and a BSD/386 source license was cheaper than a binary license for the System V implementations for the Intel platform. In addition, it was complete—no dozens of additional packages to add, each with their own activation key. Even more surprisingly, BSD/386 was more reliable than System V.

---

1. A hacker is, of course, somebody who can't tell the difference between work and play with computers. See the New Hacker's Dictionary (<http://www.tuxedo.org/~esr/jargon/html/entry/hacker.html>) for more details.

## The mid-90s

In the early days, nobody had very great expectations for the free operating systems. They didn't expect them to be as reliable as UNIX® System V. They didn't even expect them to be as reliable as Microsoft's offerings, though they did expect them to be easier to use.

In a few years, something happened that nobody had expected. Well, very few people: the operating systems did become as reliable as Microsoft's offerings, quite quickly in fact. One of the first results of this recognition came from the massive increase in Internet usage, which created a requirement for low-cost web servers. This was the reason for the foundation of BSDI, who branded their operating system "Internet Server", later "Internet Super Server". Initially they were very successful. As time went on, though, enterprising startups realized that they could save the cost of the software by using free operating systems instead of BSD/386. Thus in 1995, when Yahoo! started up, it was based on FreeBSD.

Linux was slower to reach reliability than the BSD systems: it had to be written from scratch, whereas the free BSD systems were based on a code base at least ten years old, including the most mature of TCP/IP stacks.

By the late 1990s, though, Linux had effectively caught up with the BSDs, and depending on which bigot you ask, either overtook the BSDs or never quite made it.

In the mid-1990s, another thing happened: the general public became aware of the concept of free operating systems. They were still considered very much the realm of geeks, but they were becoming known.

During this time, more and more people became involved with the projects.

## The late 90s

By the end of the 1990s, free software was well enough known in the IT industry for some people to form companies to market it. The result was a further increase in the size of the projects. At the time of writing, the FreeBSD project has 318 committers, in other words developers with write access to the source tree, probably ten times the number as of January 1995.

## Social changes

Clearly the social structure of the FreeBSD project has changed greatly in its nature in the last ten years. By mid-2000, the strain was beginning to show:

- The core team was no longer the group of the most active committers. Their function had effectively become more administrative, but they hadn't recognized the fact yet.
- The architectural direction of the project had become a little vague. The position of chief architect, previously held by David Greenman, had been vacant for some time. In the early stages of the project, most of the work in the project had been to make FreeBSD a stable UNIX-like operating system, but that goal had now been achieved. There was more work to do, but the only clearly defined goal was to improve the SMP performance—the SMPng project, which at the time had just started, and which is still continuing.
- A related problem was the attractiveness to end users. Like other free operating systems, FreeBSD has always been a developer-driven project, but the main source of project funding came from selling CD-ROMs. The sales of the CD-ROMs were obviously dependent on the perception of the purchaser, but nobody in the project was overly concerned with this issue.
- Some developers exploited the ineffectiveness of the core team by doing whatever they wanted. This notably included making changes to the system to match their view of what was needed, possibly breaking parts of the system (those parts which they didn't use), and leaving it to others to clean up.
- This in turn, along with an observed inability on the part of the core team to solve the problem, caused a serious decline in the morale of the project, and a number of people left the project as a result.
- One of the most noticeable rogue developers was a member of the core team. As a result, the lack of activity of the core team was perceived as cronyism.

- The core team had adopted a policy of not communicating the reasons behind its decisions, partly to hide the fact that many members were inactive, and partly to avoid sparking conflicts. Not surprisingly, this gave them a reputation for secrecy.

In November 1999, Nate Williams asked in a mail message:

*Finally, what is the purpose of core? I used to know, but I'm not sure anymore. What determines if someone should become a core member? Is there any way to lose your core member status, in the same manner that you can lose the ability to be considered a maintainer? Do you have to quit in order to not become a core member? (So far that's the case).*

*My \*biggest\* fear is that we will lose active developers simply because we just plod along hoping that everything will work out, and hope that someone will pick up the torch from time to time and take us in some sort of good direction.*

*Lah-de-dah, lah-de-dah. Once upon a time, core members were folks were \*really\* excited and highly motivated to work on this thing, and would spend nights/weekends and all sorts of time on this. But, core is now older, and our real lives get in the way now.*

A couple of people suggested various ways to reform or change the core team, including the possibility of disbanding core altogether and becoming an anarchy, or voting for the core team. A number of people came up with remarkably complicated voting models. Finally, Jordan Hubbard came up with a suggestion and asked the developers to vote on a number of alternatives. Out of 94 votes cast, the most popular were:

- The idea of core is fine, its membership simply needs a shake-up and some mechanism added for voting in new blood. This alternative received 58 votes.
- The idea of core is fine, but some of members simply need to leave. This received 12 votes, most of which identified a single specific member.
- Core needs to be broken up into an oversight/human resources group, leaving architectural decisions to developers. This alternative received 9 votes.

- Don't change anything, core is fine the way it is. Received 7 votes
- Disband core entirely and let committers create a new structure in its place. Received 7 votes

Clearly the majority was for a democratically elected core team. More discussions ensued. Some people were concerned that politics would take over from reason, and the people who would get elected would be those who could drum up enough followers, not those who could do the best jobs. A team of volunteers, consisting of Jonathan Lemon, Warner Losh and Wes Peters, got together and thrashed out the existing voting models and came up with the following proposal, on which we also voted:

- Active committers have made a commit to the tree in the last 12 months.
- Core consists of 9 elected active committers.
- Core elections are held every 2 years, first time September 2000.
- Core members and committers may be ejected by a 2/3 vote of core.
- If the size of core falls below 7, an early election is held.
- A petition of 1/3 of active committers can trigger an early election.
- Elections will be run as follows:
  - Core appoints and announces someone to run the election.
  - 1 week to tally active committers wishing to run for core.
  - 4 weeks for the actual vote
  - 1 week to tally and post the results.
  - Each active committer may vote once in support of up to nine nominees.
  - New core team becomes effective 1 week after the results are posted.
  - Voting ties decided by unambiguously elected new core members.
- These rules can be changed by a 2/3 majority of committers if at least 50% of active committers cast their vote.

These "bylaws" passed by 117 yes votes to 5 no votes, thus also disproving the concern that committers wouldn't be interested enough to vote for the core team.

A couple of these provisions look a little unusual:

- The rationale behind the surprisingly long election period was that, since this is a volunteer project, many people might miss a shorter election period, especially Europeans on multi-week leave.
- We spent a lot of time discussing how to vote. We were concerned that if each voter had only one vote, the majority would vote for the same two or three candidates, effectively leaving the remainder to chance. Even worse, it could lead to less than 9 candidates being voted for at all. Initially, we also discussed a “veto” vote: “Don’t let *<that bloke>* onto core”. You’ll note from Jordan’s poll that the second most popular model was “expel *<that bloke>* from core”. Neither of these suggestions were accepted.

## The election

Nominations for candidacy were accepted between 5 September 2000 and 12 September 2000, after which the election started immediately. It finished on 10 October 2000. The results were announced on 12 October 2000, just in time for the beginning of BSDCon 2000 (<http://www.bsdcn.com>).

A total of 17 candidates registered, surprisingly close to the size of the previous core team. Only 8 of the previous core team stood for election. Campaigning was almost non-existent.

## The morning after

The members of the new team, later to be dubbed *core.2*, were:

- Satoshi Asami, member of the old core team. Guardian of the Ports Collection Japanese.
- David Greenman, one of the founders of the FreeBSD project, and member of the old core team. Kernel hacker and former principal architect of the FreeBSD project. American.
- Jordan Hubbard, one of the founders of the FreeBSD project, and member of the old core team. Release engineer and former president of the FreeBSD project, a position which he had dropped some time before. American.
- Greg Lehey, newly elected. Kernel hacker, author of the Vinum Volume manager. Australian (Adelaide).
- Warner Losh, newly elected. Network hacker. American.
- Doug Rabson, member of the old core team. Kernel hacker, responsible for the port of FreeBSD to the Alpha platform. British.
- Mike Smith, newly elected. Low-level kernel hacker. Australian (Adelaide).
- Robert Watson, newly elected. Network hacker, FreeBSD security officer. British.
- Peter Wemm, member of the old core team. Universal Kernel Hacker. Australian (Perth).

In summary, the new team included five members from the old core team. Two candidates from the old core team were not re-elected. The composition of the team changed in other ways: five members of the old core team had a non-English native language, but only one member of the new team did. Seven members of the old core team lived outside the USA, only two of the new team did. Three members of the new core team were Australians, including myself, compared to two before.

One thing that all members had in common was that they were software developers, not managers. This is not surprising, given the mode of election. I had had some management experience years before, but I believe that I was the only one, and my experience wasn’t much to write home about.

## Into the trenches

The new core team took office at a panel discussion during BSDCon 2000. We had a completely new concept ahead of us, and we certainly weren’t sure how to fulfil our objectives. Worse, we didn’t even know what our objectives were. An association like AUUG has a constitution. The best we had in the FreeBSD project were the “bylaws”, originally intended to define the modality of the elections.

## The first meeting

The second FreeBSD core team had a meeting at the end of the BSDCon in Monterey. It was, in fact, a significant event: in the course of the history of the FreeBSD project, it was the only meeting of the entire core team. Previously, an attempt to pay to bring the core team together (at the FreeBSDCon in Berkeley, the year before) had failed thanks to the efforts of the US Immigration

Department: Andrey Chernov, who lives in Russia, was deemed unsuitable for entry to the USA.

In this meeting, we tried to define what the purpose of the FreeBSD core team was. We discovered a surprising number of differences of opinion, but we finally decided:

- The FreeBSD core team does not define the architectural direction of the project.
- There will be no officers on the core team. Jordan Hubbard had suggested to take the role of spokesman, but the consensus was that people already saw him as exercising too much control on the team, and we suspected this would send the wrong message.
- The FreeBSD project is a volunteer organization, so the core team does not have a mandate to tell anybody to do anything.

That's conceding a lot. So what was left?

- The core team decides who can join the project as a "committer", somebody with commit access to the CVS tree. On request, backed by a *mentor*, who must already be a committer, the core team decides whether to admit him to the project (to "give him a commit bit"). Core voted on each application. A single "no" was sufficient to veto an application, and voting terminated after a week.
- In case of extreme misbehaviour, the core team can expel a committer from the project.
- In case of dispute between two committers, the core team mediates.

A little later we added the concept of a monthly core team report to address the accusations of secrecy made against the previous core team.

Comparing this list with the problems facing the project, a number of issues remained unanswered:

- We still had no architectural direction. The core team's intention here was that a consensus should be formed on the FreeBSD-arch mailing list. If no consensus could be formed, core would mediate.
- Attractiveness to end users. The majority of the members of the core team, being developers themselves, were not very interested in this aspect.

- Rogue developers. We couldn't agree on how to handle this one. One of the issues that was made very clear was that the core team did not have a "big stick". About the only thing that it could do would be to expel a member from the project.
- Project morale. This included behaviour of developers towards each other. Again, core did not come up with a good solution for this problem, though theoretically expulsion from the project would have been a solution.

## Acceptance of core.2

How did we go? Parts of it were excellent.

The biggest problem we found was that core members were *still* unresponsive. Applications for commit bits, our main activity, took up to several months to process instead of the one-week timeout we had set ourselves. One of the problems was the amorphous structure of core: nobody had a particular hat, so there was nobody designated to actually convey the message back to the applicants. In the course of time, that got better. Very few applications were rejected.

Publishing the monthly core reports became very slow. Although the reports for the last months of 2000 were relatively timely, the January 2001 report was released on 29 June 2001. We began to recognize that we needed help, and solicited applications for the position of core secretary, a non-voting position more akin to AUUG's business manager than to the secretary. Initially we let several applicants work on the backed-up reports. The February 2001 report appeared in November 2001, and gradually we caught up with the backlog. We signed up Wilko Bulte as core secretary, a position he still holds, and by May 2002 we had cleared up the backlog.

## Rogue developers

Not surprisingly, problems with rogue developers did not abate. Each occurrence caused a lot of angry discussion with core, which was very wearing on a number of the members. Surprisingly, it also became apparent that many core members saw each occurrence as a separate issue, and personal likes and dislikes were very evident. There was strong resistance to any general policy.

In February 2002, a developer announced his intention to commit some significant changes to the

SMP code. At the time, the most active SMP developer, John Baldwin, was moving house from one coast of the US to the other, and was thus offline. Others who were involved pointed out that these changes were in conflict with changes that John was currently testing and asked the developer to hold off. The developer committed the changes anyway.

The handling of this particular issue became a test of core's authority. For the first time, core decided to revoke the developer's commit privileges if he did not back out the commits. He did so in the nick of time (without knowing about the impending suspension), and asked core to resolve the issue. The resolution was hard, and it looked more like tactics rather than strategy. After a month of discussion involving hundreds of mail messages, core appointed John Baldwin to the position of technical lead for the SMP project, with the power to approve or reject changes.

Based on this relative success, core deliberated and came up with some rules about developer behaviour:

1. *Any committer who commits to the stable branch during a code freeze will have his or her commit bit suspended for 2 days. Any member of core or the re@ team can implement the suspension without the need for a formal vote within core or re@ respectively. The suspension will be published on -developers.*
2. *Any committer who commits to the security branch without approval from the security-officer team will have his or her commit bit suspended for 2 days. Any member of core or the security-officer@ team can implement the suspension without the need for a formal vote within core or security-officer@ respectively. The suspension will be published on -developers.*
3. *When committers engage in a commit war, both parties will have their commit bits suspended for 5 days. Any member of core can do this without the need for a formal vote. The suspension will be published on -developers.*
4. *Any committer observed to act or speak in a way that is in conflict with the normal rules of interpersonal politeness, or in conflict with the best interests of the FreeBSD Project will have his or her commit bit sus-*

*pending for 5 days. Any member of core can implement the suspension without the need for a formal vote within core. The suspension will be published on -developers.*

5. *Core reserves the right to impose harsher penalties for repeat offenders. Harsher penalties include longer suspension terms and the permanent removal of commit privileges and FreeBSD.org accounts. Implementing harsher penalties are subject to a formal 2/3 majority vote in core. The outcome of core's decision will be published on -developers.*

*In all cases where an individual FreeBSD officer takes a personal action he or she will be answerable to core. All cases can be taken for appeal to the core team. The outcome of such an appeal will be published on -developers and in the core monthly report.*

These rules looked rather rigid, but we couldn't come to an agreement to moderate them, so that's the way they remain.

## **The big stick**

In June, core received a formal complaint about the same committer who had caused us so much grief in February. He had committed code in an area on which another developer was working, without discussing the matter with him. This had annoyed the other developer to the point that he relinquished the maintainership of this part of the tree.

We discussed the matter and attempted to decide whether this behaviour was in conflict with the rules we had just published, specifically rule 4. A majority decided that it was, but there were extenuating circumstances. According to the rules, though, we still had to impose the full five day penalty.

The developers reacted in different ways, mainly unfavourably. In the meantime, as described the next section, we were in an election campaign. Some suspected that this punishment was politically motivated, since the developer in question was also an election candidate. It's not clear, however, whether this punishment improved or lessened his chances, but in any case he wasn't elected. The core team decided on a reprieve after two days, and the matter died down relatively quickly.

A few weeks later, two other highly respected committers engaged in a commit war: one committed something that the other didn't like, the other backed it out, the first recommitted it, and so on—a clear violation of rule 3. As always, there were extenuating circumstances. After some deliberation, core decided to suspend the commit bits for 24 hours. Again, this caused a commotion in the mailing lists, but it died down more quickly.

Is this working? It seems to. Core needs to understand how to dose the punishment, but the real issue here is not the temporary loss of commit privileges, it's the open recognition of inappropriate behaviour. It's still too soon to be certain, but maybe people are being more considerate as a result.

## The collapse of core.2

Round May 2001, Satoshi Asami became sick and disappeared from the scene for some time. Even after his return, he did not participate in core discussions, and after several months, we finally decided that he was *de facto* no longer a member of the core team. According to the “bylaws”, we carried on with only eight members.

After the SMP commit war described above, people were feeling tired. Everything seemed to take more effort than necessary. On 29 April 2002, Jordan Hubbard dropped a bombshell: he resigned from core. In his resignation message, he stated:

*... being in core is honestly not what it once was. For an old-timer like myself, who was used to a core team that was far more cohesive and generally on the same page, it's simply a painful experience a lot of the time. Perhaps this is due to overly rose-colored recollections of the old core on my part, and I do certainly recall us having more than our share of disagreement and inefficiency in the past, but on the balance core still feels too much like the pre-WWII Polish Parliament sometimes, where we're fully capable of arguing some issue right up to the point where tanks are rolling through the front door and rendering the whole debate somewhat moot. I'm also not blaming this on the democratic model we've adopted, a stance which would be hypocritical at best since I'm one of the folks who argued strongly in favor of it, but I guess it's going to take a few more iterations before we get it right. It will also probably be a lot easier for truly new people who don't*

*have a lot of preconceived notions of what core is to make that happen.*

*Finally, it also bears noting that while being part of the FreeBSD project is many things, it should always be "fun" to at least some degree for its participants or there's really not much point in being involved. Being in core, where one gets to deal almost solely with conflict resolution and bureaucracy, is not fun in any sense of the word and while being in core constitutes the bulk of my involvement, without any cool development work (which I also haven't had time for) to counter-balance it, it simply leaves me with less and less enthusiasm for FreeBSD.*

Yes, it certainly wasn't fun. Slashdot picked it up with glee, of course, and the usual “FreeBSD is dying” trolls came out again. That's not the point, of course. This has nothing to do with FreeBSD as an operating system, these are simply some interesting project dynamics.

As if that wasn't interesting enough, five days later Mike Smith also resigned from core. In his message, he wrote:

*FreeBSD used to be fun. It used to be about doing things the right way. It used to be something that you could sink your teeth into when the mundane chores of programming for a living got you down. It was something cool and exciting; a way to spend your spare time on an endeavour you loved that was at the same time wholesome and worthwhile.*

*It's not anymore. It's about bylaws and committees and reports and milestones, telling others what to do and doing what you're told. It's about who can rant the longest or shout the loudest or mislead the most people into a bloc in order to legitimise doing what they think is best. Individuals notwithstanding, the project as a whole has lost track of where it's going, and has instead become obsessed with process and mechanics.*

...

*From a technical perspective, the project faces a set of challenges that significantly outstrip our ability to deliver. Some of the resources that we need to address these challenges are tied up in the fruitless metadiscussions that have raged since we made the mistake of electing officers. Others have left in disgust, or been driven out by the culture of abuse and distraction that has grown up since then. More may well remain available to recruitment, but while the project is busy infighting our*



*chances for successful outreach are sorely diminished.*

Does this look familiar? It should do to anybody in commercial software development projects. Coincidentally, both Mike and Jordan work for Apple. They must be involved in project planning there; it's to be expected that it works better at Apple.

### core.3

After Mike Smith's resignation, core only had six members left. According to the "bylaws", this meant that elections had to be held. That, too, caused long discussions. When, how quickly, should we change the "bylaws" first? We did a straw poll which showed that the committers did not want to wait, and they didn't want to change the bylaws. An election schedule was published in accordance with the "bylaws", and then people decided there wasn't enough time for people to declare their candidacy. Accordingly, the period for nominations was extended, and voting ran from 29 May to 25 June.

Despite the perceived tiredness, a record number of nominations were received. In contrast with the first core elections, there was a significant amount of politicking, with some candidates publishing lists of their preferred partners in core.

The results were announced, not as planned on 1 July, but immediately after the elections closed. The third core team consists of the following members:

- John Baldwin, newly elected. FreeBSD SMP technical lead. American.
- Jun Kuriyama, newly elected. Japanese.
- Greg Lehey, member of core.2. Kernel hacker, author of the Vinum Volume manager. Australian (Adelaide).
- Warner Losh, member of core.2. Network hacker. American.
- Mark Murray, newly elected. Security hacker. Zimbabwean.
- Wes Peters, newly elected. Network hacker. American.
- Murray Stokely, newly elected. FreeBSD Release Engineer. American.

- Robert Watson, member of core.2. Network hacker. British.
- Peter Wemm, only member of the original core team left. Universal Kernel Hacker. Australian (Perth).

This time around, all those members of the second core team (four) who stood for election were re-elected.

In many ways, this core team composition is the best we have had. In particular, we have better technical representation (John Baldwin) and the representation of the Release Engineer, Murray Stokely.

The first elected FreeBSD core team got off to an erratic start. As the resignation letters of Jordan Hubbard and Mike Smith show, this was at least partially because of unrealistic expectations of the tasks involved. The new core team looks like it might finally be pointing in the right direction.

## Conclusions

The idea of independent free software projects is still very new. Things are changing rapidly, and it's difficult to guess what will happen in the future. A number of things have become evident, though:

- It's possible to run a small software project without significant administrative overhead. It's impossible to run a large software project without significant administrative overhead.
- A good kernel hacker is not automatically a good manager.
- The problems that large projects face do not differ significantly between volunteer and commercial projects.
- Working on a big software project has almost never been "fun". That was one of the reasons that the free operating system projects started in the first place: here was a chance to work on software free from the constraints of project management. This aspect, sadly, is a thing of the past.

In summary, I suspect that the FreeBSD project, and other similar projects, will gradually become more formalized, more like commercial operating systems. So what will distinguish them? I fear that the distinctions will become less and less as time goes on.