

The Forkin' Fallacy

Brendan Scott¹

OPEN SOURCE LAW

Contact: inquiries@opensourcelaw.biz

Date: December 2004

The Forkin' Problem

One meme which has previously enjoyed some popularity is the idea that Linux is prone to, or likely to, “fork” - that is, to split into a number of separate, possibly competing, branches. Ironically, this argument is based on Linux's great strength – its licensing terms. The argument is that forks can arise because Linux's license at least inhibits, and probably prevents, forced centralized control of strategic direction of the Linux kernel. Having asserted that the licensing permits the creation of forks the argument continues, by reference to the UNIX Wars or the multiplicity of Linux distributions, that this is a bad thing. In this paper we argue that the licensing regime for Linux is its great strength and that, over the long term, it encourages consolidation. We argue further that the creation of forks under these licensing terms promotes innovation and competition and is therefore a good thing both for Linux and for the economy.

The Unix Wars²

“The Unix wars were the struggles between vendors of the Unix computer operating system in the late 1980s and early 1990s to set the standard for Unix henceforth. These battles are commonly held to have harmed the market acceptance of Unix and created a market gap that allowed the rise of Microsoft Windows NT.”

http://en.wikipedia.org/wiki/UNIX_wars

During the 1980s the UNIX operating system split into a variety of different, competing flavors. Despite the market wanting interoperability (as evidenced by numerous attempts at establishing UNIX standards, including SVR4), the vendor of each flavor³ of UNIX had strong incentives to pursue a strategy of product differentiation. In other words, they deliberately made their products incompatible or one way⁴ compatible with other flavors. In this way customers who had installed that flavor would face substantial costs if they acquired a different flavor of UNIX from another vendor. There is the perception that this pursuit of product differentiation did great damage to the UNIX market generally. The UNIX experience therefore stands as a warning of the potential dangers of the multiplicity of forks within an industry.

Licensing

When people mount arguments about UNIX forks, they often overlook the key factor which led to the fragmentation of UNIX – its licensing terms. Some of the characteristics of those licenses are well known. In particular, they permitted licensees to make changes to the software, and to

¹ Brendan is a lawyer. He runs a practice called Open Source Law, devoted to ICT and open source law, based in Sydney, Australia. Brendan speaks frequently on ICT and open source issues and has advised numerous government agencies and private companies on the implications of open source. Orthography is US English (hence eg favor, license etc.)

² For more information on the UNIX Wars see: <http://www.faqs.org/docs/artu/ch02s01.html>

³ I have used the word “flavor” to distinguish the UNIX offered by different vendors. Some people might use “version”, but I have reserved that for use in referring to different versions within a particular flavor – hence “version 2.6 of the Linux kernel”.

⁴ In that it was easy to move to, but not from their flavor.

distribute object code versions of those changes without the source code and without permitting third parties to use the source code. In other words, they supported the creation of closed derivatives. More importantly, they permitted the creator of a fork to prohibit other forks being cross fertilized by changes in their fork. It is in this way that the license terms permitted incompatibility. If the licensing had *required* the option of cross fertilization any incompatibility must necessarily be short lived because the elements causing incompatibility could be incorporated into other forks. The licensing terms therefore included an express permission, almost an invitation, for fragmentation. The same is not true in the case of Linux. The licensing terms for Linux *require* that if the kernel is modified and those modifications are distributed, then those modifications must also be licensed on the same terms as the license for the kernel. In other words cross fertilization is always permitted if the modifications in a fork are distributed. One of the great benefits of this regime is that it is a force for consolidation. If Linux was to fork, any improvements that one fork would be permitted to be incorporated into each other fork.

Closed v Open

“The essence of the bazaar is ... the right to fork. Anyone who doesn't like Linus's decisions about Linux can fork the codebase, start his own effort, and compete for developer and user attention on a legally equal footing.”

Eric Raymond, Open Letter to Jonathon Schwartz, Reported 22 November 2004,
<http://www.onlamp.com/pub/wlg/5961>

Another aspect which is overlooked is the origins of and motivations for the creation of forks. The experience from the UNIX Wars was that the creation of forks was driven more by vendors seeking to lock in customers, and lock out competition than by market demand. Indeed, the licensing conditions permitted vendors to ignore market demand to a greater degree than they would otherwise have been able to in a free software market. Forks in that situation were a one way street – once a fork had occurred, it was very difficult to consolidate that fork with others in the future. The key difference is the kinds of fork being created. I will call these “vendor driven” (or “closed”) forks. There is another kind of fork, which occurs when there is an unmet need in a market and the fork arises in response to that need – I will call these “customer driven” or “open” forks.

Closed Forkin' Bad

Despite the appearance of similarity between closed and open forks, a competition policy analysis shows that the outcomes for the economy are significantly different. The incompatibility between closed forks means that the creation of each closed fork reduces competition and consequently reduces customer choice while raising customer price. Conversely the creation of open forks increases competition, consequently increasing customer choice while lowering customer price. It is no surprise therefore that the forks which resulted from the UNIX Wars were attended by adverse impacts for the market.

Linux Forkin' Good

One possible consequence of an open fork is that, depending on the licensing regime, the open fork may later become a closed fork. Therefore the market outcome, while better than a closed fork, is less than optimal. Where open forks arise off the back of a strong open source license, such as the GPL, then there is a guarantee that any future improvements which arise in the fork will remain open. I will call these “market driven” forks. The optimal market outcome therefore occurs where the forks are market driven. The forks under Linux are specific examples of market driven forks.

As has been mentioned in some media reports, the creation of forks of the Linux kernel is part of the development methodology, with numerous concurrent kernel versions being developed in parallel between major versions (2.2 v 2.4 v 2.6), between development and production versions (2.4 v 2.5 v 2.6) and even between minor versions (2.6.8 v 2.6.10). This permits development to proceed unhindered, while end users adopt the version they feel best maximizes their utility. If they favor newness and innovation, they will use the a more recent version, if they favor stability they will use an older version. Under this approach nobody is left behind because a vendor is breaking or discontinuing backward compatibility – if there is a market for that compatibility it will be back ported into the older versions (as is evidenced by the continued improvement of the 2.4 kernel). Further, developments in newer kernels can, and are, back ported into older versions of the kernel, sometimes for years after the active development of the major version has ceased.

No Forkin' Worries

In practice there are substantial barriers to the creation of a fork, primarily through the existence of embedded knowledge of the people working on the main development branch, and network effects tending to preference the main development branch. As such a fork is only likely to occur where the development path of a software project does not meet the needs of its target community, and that that difference is sufficiently great to overcome these barriers to the creation of a fork. In these circumstances, if anything, the market driven forks must be considered to be a benefit as it ensures that the code development is responsive to the needs of the target community. In any event, the mere presence of a fork does not require the original branch of development be shut down, and it is entirely consistent for there to be two concurrent development streams, each of which cross fertilities the other (as a consequence of the license governing the Linux kernel (ie. the GPL)). If a criterion for evaluating the value of software is that it reflects the ongoing needs of its target market, market driven forks should be considered to be a positive, rather than a negative.

Conclusion

The Linux versions example given above demonstrates in a practical way that a multiplicity of market driven forks gives rise to consumer choice and vigorous competition, without anti-competitive outcomes. Market driven forks allow maximal utilization of software with the added benefit that innovations created by a fork are able to be subsumed into the main trunk. Therefore this "lack of control" in reality drives quality. The only reason a project is forked is if it isn't doing its job. The only reason a fork survives is because it does a better job than the main branch. Further, under the GPL the main branch is able to reintegrate work from the fork anyway, so there has to be something pretty special about a fork for it to survive. Indeed, the lack of ability to fork actually jeopardizes the quality of the software by insulating the main branch from competition. This may may be the main reason why Linux is devastating other members of the UNIX family.

Thus, the proposition that the creation of forks in Linux is bad for Linux, for consumers, or for the economy is simply not true. It is a forkin' fallacy.