gaither@
globe.com

Chris Gaither
Boston Globe

SCO's Derived Case Against Linux

Research Note

Gordon Haff
6 August 2003

Legal actions often involve throwing out a wide range of claims and allegations in the hope that at least one will stick. When it comes to SCO's intellectual property claims against Linux, it's been difficult indeed to separate the fundamental arguments from the secondary window dressing. This is especially true when you consider that the only actual lawsuit filed so far is against IBM, and that is for alleged contract breaches rather than direct copyright infringement claims. But as SCO prepares to demand that essentially all enterprise Linux users purchase UnixWare licenses, SCO is extending its arguments from the purely contractual to the copyright realm, in which even those that have had no direct contact with SCO could, in theory, be liable for damages.

We recently met with Chris Sontag, who heads up the company's SCOsource division responsible for IP licensing, as well as Mark J. Heise, an attorney with SCO's law firm, Boies, Schiller & Flexner LLP, to better understand SCO's position. Based in part on that discussion, it's becoming clear that the issues identified earlier in the case—such as line-for-line copying of SCO-owned code into Linux, or the improper use of SCO binary libraries—are not the key. Rather, SCO's case against Linux at large instead rests most heavily on an expansive definition of derivative works—adaptations or extensions to an existing copyrighted work such as a film based on a novel or a new version of a software program.

It's SCO's contention that many of the capabilities built into and on top of the Unix[1] kernel by vendors like HP, IBM, SGI, and Sun over the course of more than two decades were added to Unix at such an intimate level that they are really inseparable from the UNIX System V code to which SCO holds copyright. This extremely broad claim—which has its roots in the copyright law covering derivative works—is the key to SCO's attempt to exert control[2] over even that Unix code that it had no hand in creating.

1. A note on terminology: We use the mixed-case to refer to the family of Unix-like operating systems, whereas UNIX System V refers to the AT&T/USL/SCO software product (from which many, but not all, Unix implementations and capabilities are derived).

Chris Gaither
Boston Globe

Copied Lines: A Red Herring

However, before delving into the minutiae of derivative works, let's consider SCO's two other Linux copyright infringement claims: the literal line-for-line code copying that has been the object of so much attention, and "non-literal transfers" of methods, structure, and sequence from System V UNIX into Linux.

Certainly, SCO will continue its litigation regarding direct copying, if only because it would be the most straightforward claim to prove. But it still won't be trivial. It's not just a matter of pointing to (nearly) identical code—which does indeed appear to be present. For example, the `malloc` and `mfree` routines in Seventh Edition Unix (a precursor of System V) contain many lines of code and comments that are substantially identical to `atealloc` and `atefree` in the Linux 2.4 kernel (where they carry a Silicon Graphics, Inc. copyright notice).[3] The code also appears in `rmalloc` and `rmfree` in 4.3BSD (which are marked copyrighted 1986 by the Regents of the University of California), but apparently not in current BSD versions. Given the wide availability of early Unix source code, [4] it's certainly plausible—even likely—that chunks of ancestral Unix code have made their way into Linux.

But SCO also has to prove that copied code originated in code to which it holds valid copyrights—which is far from a slam dunk. Most notably, in the early nineties, there was a series of complicated copyright claims and counterclaims between USL, BSDI, and CSRG.[5]

The final agreement which paved the way for the creation of a subsequent BSD4.4-lite—the direct ancestor of today's open source BSDs, and a Unix code base purported free and clear of any UNIX System V licensing encumbrances—remains under court seal. However, along the way the US District Court of New Jersey ruled against an injunction sought by USL, and that opinion *is* public. In that ruling, the judge questioned the validity of USL's copyright to UNIX 32V (an offshoot of Seventh Edition UNIX).[6,7] If the reasoning were upheld in the current case, it calls into question whether there are any copyrights to violate if code was copied from *pre*-System V UNIX—as the previous example appears to have been.

What's more, even if SCO can prove that there is directly transferred code that violates valid copy-

2. SCO does not claim to own copyright of the added code; that rests with its creators. What it *does* claim is the right to limit the use of that code under the terms of the various vendors' UNIX System V licenses.
3. When we met with SCO, they showed us the source code printout of several Linux functions including `atealloc` and `atefree` with the lines that allegedly matched lines in UNIX System V source highlighted and the Silicon Graphics copyright notice redacted. SCO offered to show us the corresponding System V UNIX source code under NDA, but we declined. Following our meeting we located a copy of the Linux source code shown to us and compared it to other Unix versions whose source code is available on the Internet. We have little doubt the blocks of code that appear in both Linux and various Unix variants are also present in System V UNIX, although we do not have the System V UNIX code to make a direct and definitive comparison.
4. For example, large chunks of mid-seventies vintage Sixth Edition Unix became widely available in *samizdat* fashion; and was eventually published in a well-known book *Lion's Commentary on UNIX 6th Edition* (though the book was published with the right to use the code for educational purposes only).
5. Respectively: Unix Systems Laboratory, an AT&T offshoot and then-holder of Unix intellectual property rights; Berkeley Software Design, Inc., the sellers of Net2, a version of the Unix-like BSD that originated at the University of California at Berkeley; and the Computer Sciences Research Group of the Regents at Berkeley, the BSD copyright owners.
6. Essentially, the judge held that AT&T had not properly published copyright notices on significant portions of its UNIX code. Under the laws then (but no longer) in effect, such an oversight could invalidate a copyright claim. See Unix System Laboratories, Inc. v. Berkeley Software Design, Inc., US District Court NJ, 1992.

rights, its victory could still be hollow unless the copied code is found to be either pervasive or critical. While SCO hasn't called out a specific number of violations, it does talk about "tens" of files and possibly hundreds of lines of code that show direct copying. Yet in the context of the millions of lines of code in the Linux 2.4 kernel,[8] that's relatively small potatoes on a percentage basis. If a court found the code in question to be critical to the operation of Linux, or determined that it embodied particular trade secrets, case law suggests that even a small quantity would be a significant infringement.[9] But so far we have no indication that's the case.

SCO can certainly try to make a case for damages against any companies that have knowingly taken the copyright notices off UNIX System V files and turned them over to the Open Source community. However, based on the minimal evidence presented so far, the scope and seriousness of the literal copying scarcely seems to justify asking end users and other parties who had no hand in any possible copying to pay fees to SCO. SCO has not yet formally demanded such payment—in the form of UnixWare licenses—from any end users; it has, however, clearly stated its intent to do so.

SCO is similarly placing relatively light emphasis on "non-literal transfers" of UNIX System V methods, structures, and sequences. This is the claim that engineers have transferred techniques and approaches from UNIX System V to Linux, even if they haven't copied code on a line-by-line or literal basis. Given how many structural and conceptual elements of Unix are shared with other operating systems—and how far afield this claim gets from copyright protection's focus on the expression of ideas rather than the ideas themselves—don't expect to hear a lot about this aspect of SCO's claim. SCO is even quite explicit that POSIX-compliant or even "Unix-like" operating systems (*a la* Linux or BSD) don't *inherently* infringe on its intellectual property. Instead, it's the enterprise-level intellectual property transfers from various Unix vendors that it's after, even though they presumably owe less to the methods and structures of System V than do the core operating systems themselves.

Given the weaknesses and/or limitations of its other two claims, SCO appears to now view the third, "derivative works" leg of its case as the key to extracting value (read "money") from Linux. If not quite a claim to all Unix goodness delivered by anyone during the last decade or so, it's still enormously broad. Various legal side issues notwithstanding, this case will likely turn on the validity of SCO's expansive definition of derivative works as it applies to Unix. So let's examine this claim—and the hurdles it faces—in detail.

The Crux of the Case

The most common types of derivative works are those that have been adapted or transformed from a work in a different medium. A movie adapted from a novel is a derivative work; the novel's author must explicitly grant a moviemaker the right to create it. In software, derivative work typically refers to a new version of a program which, in spite of new content, is usually clearly adapted from the previous version.

SCO maintains that, in the case of its UNIX System V code, essentially all extensions to the kernel are derivative works. This includes new file systems, schedulers, clustering, NUMA support, SMP scal-

7. If you are, like most people, already dazed and confused by the numerous twists and turns of Unix's heritage, then *by no means* should you examine Éric Lévénez's fabulous but dizzying Unix historic timeline. Otherwise, you may view it at http://www.levenez.com/unix/. For a condensed Unix history that focuses on the versions most relevant to the SCO-IBM case, see Eric Raymond's and Rob Landley's "OSI Position Paper on the SCO-vs. IBM Complaint" (http://www.opensource.org/sco-vs-ibm.html).
8. Estimates vary from 1 million to 3.4 million, depending on who's counting and what they're counting. A full Linux product such as Red Hat Linux that includes shells, utilities, services, GUIs and a set of user applications runs over 30 million lines of code.
9. "A *de minimis* defense does not apply where the qualitative value of the copying is material." (Dun & Bradstreet Software Services v. Grace Consulting, US Third Circuit Court of Appeals, 2002)

ability improvements, and so on, *ad nauseum*. SCO concedes that there might be some argument over which pieces are inextricably part and parcel of the Unix kernel—and are therefore most derivative—and which are not. But it really only gives a pass to software that talks to the OS at the application API boundary. Extend any deeper into the kernel, and you've somehow inextricably attached yourself to SCO's intellectual property. As a result, you can now only distribute your code as allowed for by your UNIX System V license with SCO or its predecessors—which usually means either (i) in binary form, or (ii) only to those who have valid UNIX System V licenses (which the Linux/open source community clearly lacks).

SCO's central tenet extends beyond a contention that UNIX System V code *plus* extensions to the kernel are derivative works. They quite clearly are. Few would seriously dispute that a complete Unix (e.g. AIX, HP-UX, or Solaris) built atop ancestral UNIX System V source code is derived work. But the heart of SCO's case rests on a much more exuberant claim: That not only is a complete UNIX System V-based Unix a derivative work, *but so is the added code taken alone*. And because UNIX System V licensees generally have a contractual obligation to protect UNIX System V derived works under the same terms as the UNIX System V source code itself, giveaways of this added code to the open source community are strictly *verboten*.

Open source advocates and many others have summarily dismissed as ludicrous such claims to control over intellectual property that SCO had no hand in creating. This is a mistake. The claims are certainly enormously broad and appear to extend beyond what have been considered derivative works in other Federal cases. But all derivative works *by definition* include original content, and yet are constrained by whatever rights were granted by the creator of the original work. The question here isn't whether SCO has rights around Unixes containing large quantities of unique non-SCO IP. It does. Rather, it's whether the IP that has been carved out of various Unixes and handed over to the open source community is sufficiently independent of SCO's copyrighted System V UNIX code to be considered non-derivative.

How Derivative is Derivative?

Ironically, considering that IBM is both the strongest major-label promoter of Linux and the only company against which SCO has so far actually filed suit, the derived works claim may not extend to IBM itself. SCO clearly thinks it does, but in 1985 AT&T signed a "side letter" to IBM's original license agreement, which stated in part "we agree that modifications and derivative works prepared by or for you [IBM] are owned by you." It goes on to state that "ownership of any portion or portions of SOFTWARE PRODUCTS [UNIX] included in any such modification or derivative work remains with us." That seems fair—and it seems fairly clear that IBM extensions like JFS and LVM, at least, would be IBM's to do with as it sees fit, including giving it away to Linux developers. SCO says not. SCO claims that this clause does not give IBM any special rights. Yet to our laymen eyes it raises the ironic possibility that IBM may, perhaps uniquely, have the right to distribute derived Unix code that it created.[10]

But the side issue of the side letter aside, the breadth of SCO's "derivative works" claim is *breathtaking*. Linux is the quintessential son of a thousand fathers. Much of the code donated to it by vendors came down from ancestral mainframe or minicomputer operating systems. For example, no Unix had a journaling file system before IBM's JFS emerged in 1990. JFS, while loosely based on BSD's Fast File System, was designed primarily around IBM's experience with database transactional management. Similarly, IBM's Logical Volume Manager derived from IBM's large systems experience—to a degree that many Unix users of the time were as likely to deride it as "not really Unix" as to

10. At least for AIX. This may or may not protect code derived from Sequent's Dynix/ptx that made its way to Linux by way of IBM, especially if they flowed through AIX first. According to SCO, Sequent did not have an equivalent side letter, nor do other vendors like Sun.

appreciate its manageability advances. Even looking into the kernel, great swaths of the organization, facilities, semantics, and codebases of AIX, DG/UX, HP-UX, and Solaris have had nothing to do with UNIX System V code, regardless of how closely they attempted to match the external behavior of UNIX System V (and later POSIX, XPG, FIPS, and Open Group standards).

Case law suggests that the courts are likely to be skeptical about SCO's expansive interpretation of derivative works copyright law. Take, for example, a 1992 ruling holding that a "Game Genie" device, inserted between a game cartridge and the Nintendo Entertainment System for the purpose of altering features of a Nintendo game, did not violate Nintendo copyrights.[11] The Game Genie certainly interacted with the Nintendo components at an intimate level—it blocked the value of a single data byte sent by the game cartridge to the CPU unit in the Nintendo and replaced it with a new value. However, the court noted that "the examples of derivative works provided by the [Copyright] Act all physically incorporate the underlying work or works. The act's legislative history similarly indicates that 'the infringing work must incorporate a portion of the copyrighted work in some form.' "

SCO's lawyers will doubtless argue technical differences between their case and cases such as this one, but SCO certainly seems to be arguing for broader control over others' work than the courts and Congress have historically been inclined to grant.

Today, IBM; Tomorrow, the World

Perhaps in an effort to avoid simultaneous tussles with essentially all the large system and software companies, SCO is downplaying the breadth of its claims. It singles out IBM for approbation, but there's hardly a major vendor out there who isn't potentially touched by SCO's broadest claims. This includes those like Sun who are currently using the contretemps to their own competitive advantage.

11. Lewis Galoob Toys, Inc. v. Nintendo of America, Inc., US Court of Appeals for the Ninth Circuit , 1992.

If IBM's JFS is infringing, then surely so is SGI's high performance XFS file system, which the company donated to open source. Indeed, why wouldn't SCO consider the Veritas File System—sold as an add-on to Linux and several Unixes—a Unix derivative work that required some form of explicit license from SCO in order to be sold?

Sun and SCO now appear buddy-buddy, but surely Sun's release of source code for internationalization technologies to X.org counts as a violation under SCO's expansive definition of derivative works. This code builds, after all, on Unix's underlying character-oriented subsystems, including ones operating at the kernel level. For its part, HP has made many contributions to Linux, not least in areas related to Itanium support and scalability, as well as around hardening for telco operations. SCO suggests that HP is probably in the clear because its Unix efforts are completely separate from its Linux efforts. But this implies that HP separates its Unix and Linux development with Chinese-wall or clean-room development procedures. It doesn't. There are many instances of engineers who've worked on HP-UX or Tru64 UNIX by day and Linux by night, or who've even been assigned from Unix development to Linux development.

If SCO were to prevail in its claims around IBM-donated technology, it's hard to believe that it wouldn't soon come knocking on a number of other vendors' doors, looking for more tribute.

SCO's Stretch

SCO is using its derivative works claim to exert control over a body of intellectual property that is vastly broader than that which it controls directly. Indeed, SCO's claim primarily concerns enterprise capabilities like clustering, advanced filesystems, and scalable SMP—none of which were in the UNIX System V code base that was historically provided to OEMs. All these enhancements were built by the OEMs from scratch, or otherwise based on IP and techniques the vendors had learned on their minicomputer and mainframe OSs. Such enhancements were then transferred to, or built

into, Linux starting with the 2.4 kernel.[12] SCO contends that these capabilities were added to UNIX at such an intimate level that they are really inseparable from the UNIX System V code to which SCO holds copyright. What makes SCO's claim such a stretch is that so much of the technology that SCO claims control over has roots in operating systems other than Unix. Much of it is not even tied directly to Unix.

When they recently visited, we found Messrs. Sontag and Heise to be calm, rational, and indeed quite personable; they are not the rabid "psycho killers" imagined by much of the Linux and open source community. We do not, however, agree with their assertions that the enhancements added to UNIX System V are as much derivative works as UNIX System V plus those enhancements would be as a whole.

This case continues to look like a "Hail Mary" pass for SCO. This is a company that was failing; it has been losing market share and relevance for years. Now it's begun to make a case against Linux in general, in addition to the one against IBM, based on an enormously expansive interpretation of copyright law. SCO, along with Boies, Schiller & Flexner, probably figure there's enough uncertainty and ambiguity, and little enough definitive case law around software copyrights, that they have at least a shot at making at least some of their claims stick. Stranger things have happened in courts, but SCO will have major hurdles of proof and precedent to overcome—not to mention IBM's quite seasoned and well-resourced legal team. Linux users and open source developers should not blithely ignore SCO's doings—but neither should they assume the worst and take precipitous, expensive, or disruptive actions as the result of this just-begun soap opera.

12. For now, SCO's case is concerned with the Linux 2.4 kernel and its successors; it does not consider Linux 2.2 to be in violation for derivative works. SCO won't take a position on whether there are copyright violations in current open source BSD versions (which are based on 4.4BSD-lite, the revision of BSD assembled from code supposedly "cleansed" of USL code and IP).