

## FAQ: SCO and Linux

*Necessary and important disclaimers: I am not a lawyer, and none of the following constitutes legal advice. At most it contains one layman's understanding of the law. In particular, the legal issues are addressed in the context of the laws of the United States. The rules may differ in other countries. If you want legal advice, ask a lawyer for it*

*I make no pretense of concealing my own biases, but the material presented here is intended to be factual and accurate, though often simplified in the interest of brevity. Despite my best efforts, it is likely that some errors of fact or of law have crept into this document. I invite correction of any such errors. Nevertheless the responsibility for the contents of this document is mine alone.*

### **What is SCO doing about Linux?**

Four things, mainly:

- SCO has sued IBM for allegedly misappropriating SCO's proprietary Unix source code (and "methods," whatever that means) for use in Linux. It has also purportedly terminated IBM's license to use Unix code in IBM's AIX product.
- SCO claims that some of the source code in Linux has been misappropriated from SCO's code, but refuses to identify the alleged infringements.
- SCO has announced plans to invoice Linux end users for licenses to the SCO code allegedly present in Linux, implicitly threatening litigation against those who do not meet its demands.
- SCO has declared that the GNU General Public License (GPL), the license for Linux and much other open source software, is legally invalid.

These actions have outraged the open source community, and perplexed many others in the IT industry. SCO faces a counter suit from IBM and other lawsuits from Red Hat and others.

### **What is intellectual property?**

As the term is generally used, there are four kinds of intellectual property: patents, trademarks, copyrights, and trade secrets. These four kinds of intellectual property are distinct, and different legal rules apply to them. Hence the term "intellectual property" has no very well defined meaning in law. It is best to avoid using the term altogether, unless you want to mislead people by blurring important distinctions.

### **What kinds of intellectual property does SCO claim in Linux?**

SCO owns no relevant patents, nor does it claim to. The Open Group owns the trademarks for "Unix" and "UnixWare".

The complaint filed by SCO makes only a passing reference to alleged copyright infringement, and that reference does not attribute any infringement to IBM specifically. Indeed, SCO has stated that the filing against IBM is about breach of contract, not about copyrights.

What's left is trade secrets – information that one or more parties are under contract not to disclose.

### **Do SCO's versions of Unix contain trade secrets?**

Trade secrecy is like virginity: once lost, it can never be restored. If someone has disclosed a trade secret, even if the disclosure was illegal and improper, the information is no longer a trade secret. Only the one who improperly disclosed it is liable for damages. Others may use the information freely, except of course as that usage may be otherwise restricted by copyright law or other considerations.

It would be difficult for SCO to claim any trade secrets in the Unix code base that it acquired from Novell, and ultimately from AT&T. This source code has circulated widely for years, much of it in book form. Caldera itself has published earlier versions of the source code under a BSD-style license. The principles and architecture of the AT&T-derived Unix kernel are described in books and taught in University classes.

It is possible for SCO to claim trade secret status for additions and modifications to Unix that were made after receiving the ancestral source code. Such changes could have been made by SCO or by its predecessors in interest.

### **Has IBM disclosed any of SCO's trade secrets?**

SCO says it has, and IBM says it hasn't. Until SCO identifies the trade secrets that IBM has allegedly disclosed, it is impossible for an outsider to evaluate SCO's claims.

Likewise it is impossible to evaluate other claims of breach of contract without knowing about the contract.

IBM secured a perpetual and irrevocable Unix license from AT&T. Later SCO took AT&T's place as successor in interest. These contracts are on the public record, but other contracts between the parties may not be – such as the agreements surrounding the ill-fated Project Monterey.

This FAQ will not dwell on the lawsuits between SCO and IBM. IBM can take care of itself.

### **Does SCO own Unix?**

No. SCO's lawsuit asserts repeatedly that SCO owns Unix, but this assertion is demonstrably false.

The trademarks for "Unix" and "UnixWare" belong to the Open Group. Legally, Unix is whatever the Open Group says it is. In practice, the Open Group is a standards body. It permits the Unix trademark to be applied to any operating system that has been shown to comply with the applicable standards, regardless of where the source code came from. By this definition, a version of Unix may be genetically unrelated to the original AT&T Unix.

SCO owns two particular implementations of Unix, namely OpenServer and SCO UnixWare.

### **Does SCO own the copyrights to the ancestral AT&T Unix source code?**

This issue is murky. Evidently, SCO owns the copyrights to whatever *can* be copyrighted in the ancestral Unix code. However, the extent of those rights is unclear.

According to a 1993 court ruling, AT&T had “failed to demonstrate a likelihood that it can successfully defend its copyright in 32V.” The terms of the eventual settlement, though not publicly disclosed, were widely viewed as a crushing defeat for AT&T.

The 1993 ruling would of course not apply to any code written subsequently. Modifications to code that *was* covered by the 1993 ruling are a gray area, and would have to be considered on a case by case basis.

### **But didn't SCO just register its copyrights? Doesn't that establish ownership?**

Registering a copyright only records a claim of ownership. It does not prove that the claim is valid.

Anyone who pays a \$30 fee and fills out the paperwork can register a copyright on anything, including someone else's creation. Such a registration is a formality, a procedural prerequisite for a copyright infringement suit. A registered copyright may be declared invalid if someone demonstrates that the copyright holder is not entitled to it.

## **Does Linux infringe on SCO's copyrights?**

The most that can be said with confidence is that there is no demonstrated infringement.

Let me repeat that: *there is no demonstrated infringement.*

### **Why don't you just say that there is no infringement, period? It looks suspicious to hedge like that.**

Saying that there is no infringement is like saying that there are no unicorns: I can't prove it. No matter how hard I look, maybe there's a unicorn that I missed somewhere.

On the other hand, if you claim that there *are* unicorns, then the burden is on you to show me one. If you point to a goat and a cow as examples of unicorns, your credibility will suffer. If you then claim that you can't show me any unicorns because they're secret, but that I owe you \$699 for every unicorn on my property, don't be surprised if I refer your invoice to the nearest Attorney General.

### **But SCO already showed two examples of infringing code. Even a non-programmer could tell that the Linux code was almost identical to the SCO code. You can't explain that by coincidence.**

Yes, it was similar, and not by coincidence. However, similarity does not establish provenance. From merely comparing the code you can't tell whether the Linux code was derived from the SCO code, or the SCO code from the Linux code, or both from a common ancestor.

To establish provenance you have to trace the history of the code -- who contributed it in the first place, who contributed each modification, and when. For Linux this history is in the public record. Anyone who knows how to look it up can do so. For SCO's code, and for proprietary code in general, that history is not publicly available.

### **What about the first example -- the memory allocation code? Even the variable names are the same.**

This code goes back to at least 1973, and was originally written by either Dennis Ritchie or Ken Thompson when they were both at AT&T. Versions of this code appear in various versions of Unix, including BSD and the AT&T flavors. One version appears in *The C Programming Language* by Kernighan and Ritchie, a widely used introduction to the C programming language.

Because the differences among these versions are very minor, it's difficult to determine from the code alone exactly which was derived from which. Several people in the open source community have analyzed the possibilities and come to somewhat different conclusions. However, it is clear that the Linux version is only trivially different from a version that was released as open source under a BSD-style license by Caldera (now SCO) in 2002.

This code was present in Linux only briefly, and applied only to some SGI hardware that never found more than a handful of customers. It was removed for technical reasons even before SCO publicly identified it as an alleged infringement.

### **What about the second example?**

In the second example, what SCO claims as its own code comes from the Berkeley Packet Filter (BPF), and does not belong to SCO at all. Under the BSD license either SCO or Linux may use this code, though the attribution must be preserved. The fact that SCO claims the code as its own suggests that at least one of the following is true:

- SCO didn't notice the BSD attribution.
- SCO (or one of its predecessors) had improperly removed the BSD attribution.
- SCO is being disingenuous.

The Linux version is not derived from the BSD code. It is a clean room implementation written by Jay Schulist, based on a protocol specification. It looks similar to the BSD code because each version implements the same specification in the most obvious way. It would have been unnatural to implement it in any other way.

SCO originally labeled the Linux version as obfuscated, meaning that the Linux developer had deliberately tried to disguise the origins of the code. Later, SCO backpedaled, insisting that the code samples were shown only to demonstrate SCO's ability to detect obfuscation. You can judge for yourself what this episode demonstrates about SCO.

**If there is infringing code in Linux, who is liable for damages?**

If there is an infringement, whoever contributed the infringing code would be liable. A Linux distributor might be liable if it failed to show due diligence. In practice, senior Linux developers are very careful about copyrights. There are cases on record where contributions have been refused solely because of doubt about their copyright or patent status.

**Would an end user be liable for damages?**

No, because the end user would not be responsible for the infringement.

If Stephen King wrote a novel, and Paramount Studios turned it into a movie without his permission, King could sue Paramount. He could not sue the movie theaters, unless they had somehow colluded with the studio in the infringement. He certainly could not sue the people in the audience, even if it were economically feasible to do so.

Furthermore, in the unlikely event that SCO collects damages from IBM for the alleged trade secret violations, SCO cannot also collect damages from end users for the same infringement.

**If end users are not liable, then how can SCO threaten to sue them?**

Beats me.

**Isn't it possible that someone slipped some SCO code into Linux, and the rest of the Linux team didn't realize it?**

Of course it's possible. There's no way anyone -- either Linux developers or proprietary developers -- can be completely sure that a piece of code wasn't copied from somewhere else, especially if the original was proprietary code, not available for comparison.

However, there's little incentive for anyone to misappropriate code into an Open Source project such as Linux. Since all the code is in the open, infringements cannot be concealed. Anyone caught misappropriating code, especially for such a high-profile project as Linux, would instantly become a pariah, apart from any civil or criminal liability. Such ostracism would be no small penalty in the open source community, where the primary reward for contributions is not money, but bragging rights.

The same is not true of proprietary code. If SCO were to misappropriate Linux code into its own products, the infringement would almost certainly go undetected, unless it were disclosed by an insider or through discovery proceedings.

Indeed, precisely such an infringement has been alleged by a former SCO employee, in the case of the Linux Kernel Personality. These allegations are currently unverifiable because SCO's proprietary code is not available for inspection.

### **Still, suppose there is infringing code in Linux. Can SCO claim damages?**

Even that much is doubtful. To begin with, SCO would have to establish a valid copyright claim -- not a simple task for the AT&T code, due to the 1993 ruling.

Furthermore, the plaintiff in a copyright infringement case must make a good faith effort to mitigate the damages, where possible, by letting the defendant remove the infringing material. SCO has steadfastly refused to do so. It refuses to identify the allegedly infringing code, or even to provide any evidence of infringement, other than the two examples discussed earlier -- both of which were immediately shown to be spurious.

### **But if SCO identified the infringing code, the Linux people would just replace it.**

Yes, they would, assuming of course that they agreed that it infringed. The question is, why does SCO want to prevent them from doing so?

Replacing infringing code would in no way remove the evidence of infringement. The developmental history of Linux is a matter of public record, available from many sources around the globe. The Linux community could not expunge it even if they wanted to.

In fact, one of the spurious claims of infringement (the memory allocator) was based on code that had already been removed from the most current version of Linux.

There are at least two possible reasons why SCO refuses to identify the alleged infringements:

- There is no infringing code. This is the assumption (perhaps an incautious one) of many in the open source community.
- SCO wants the infringing code to remain in Linux.

Why would SCO want infringing code to remain in Linux? Perhaps to extend and preserve the damages that they think they can claim; or perhaps to preserve a basis for demanding that end users purchase additional licenses. In any case, SCO's refusal to mitigate the alleged damages will severely undermine any copyright infringement suit that it may bring to bear.

### **Do I need a license from SCO to run Linux?**

You should ask your lawyer. If you ask me, I say: No.

The GNU General Public License (GPL) gives you all the permission you need to run Linux, as long as you remain in compliance with its provisions. You may also redistribute it, modify it, and redistribute it with your modifications, provided that any redistribution is under the GPL as well. Consult the text of the GPL itself for the full details.

Buying a license from SCO -- or even accepting one for free -- would be an acknowledgement that you need SCO's permission to run Linux. Such a license would grant you no rights that you don't already have. In fact it would forfeit some of the rights you do have, and grant SCO some rights that it does not otherwise have.

### **How would SCO's license interact with the GPL, for the same software?**

They are completely incompatible. SCO's license explicitly denies you rights that the GPL explicitly grants you.

Under SCO's license, you agree not to modify or redistribute SCO's Product, where the "SCO's Product" is defined as "SCO intellectual property in Object Code format." The GPL, on the other hand, expressly grants you the right to do so, subject to certain restrictions. This attempt to alter the terms of the licensing

terminates all of SCO's right under the GPL, including the right to redistribute Linux. Your rights under the GPL as a licensee in good faith are not affected by SCO's forfeiture of its own rights.

Furthermore, since SCO refuses to specify what parts of Linux it claims to own, you have no way to modify or redistribute the parts that SCO does *not* claim to own. Hence the restrictions of the SCO license effectively apply to *all* of Linux, including the parts that belong to others.

Even if SCO identified the parts of Linux kernel that it claims to own, it could not license those parts under a license of its own devising while the rest of the kernel remained under the GPL. The GPL does not permit the commingling of GPL code with code under an incompatible license in the same program.

It would be possible for SCO to use a dual license scheme, licensing its source code under the GPL for use in Linux, but licensing the same code under some other license in other contexts. Some companies have done so. SCO has not.

### **If Linux infringes on SCO's code, don't I need SCO's permission to use it?**

If you received your Linux distribution from SCO (formerly Caldera), then you already have a license from SCO to use that distribution. SCO cannot unilaterally and retroactively change the terms of that license, even for code belonging to SCO, and certainly not for code belonging to others. Even the attempt to do so terminates all of SCO's rights under the GPL.

If you received your Linux distribution from somebody else, then SCO is out of the picture, because it can't require a license for the use of somebody else's product.

### **But SCO says it distributed the infringing code inadvertently, so the GPL doesn't apply.**

This argument would be more interesting if SCO had halted its distribution as soon as it became aware of the alleged infringement. But it didn't. At this writing SCO still offers a version of Linux 2.4 for download by FTP. By distributing allegedly infringing code *after* becoming aware of the alleged infringement, and continuing to do so for months, SCO has placed the allegedly infringing code under the GPL. Therefore it no longer infringes, if indeed it ever did.

### **SCO says that the GPL is invalid, because it is trumped by copyright law.**

According to copyright law, you may make a backup copy of a piece of software, and you may copy it as needed for execution (in the sense that the program is copied, for example, to memory), notwithstanding restrictions on copying imposed by the copyright holder. This provision applies specifically to software, and not (for example) to books, because software is different from books.

SCO's interpretation -- raised in the press but not, so far, in a courtroom -- is that this provision creates a ceiling rather than a floor. In other words, according to SCO, you cannot legally make multiple copies of copyrighted software, even if the copyright owner explicitly gives you permission to do so.

This legal theory is so bizarre and deranged that no one who is not in the pay of SCO takes it seriously. It's doubtful that anyone at SCO takes it seriously either, except as a way to confuse and frighten potential users of open source software. If by some freakish mischance this theory were upheld, Dell and Compaq would not be able to sell you a PC preloaded with Windows, nor with any other copyrighted software that they didn't write themselves.

Despite this novel legal theory, SCO has long redistributed other people's software under the GPL, and continues to do so.

SCO can't have it both ways. It can't redistribute software under the GPL while insisting that no one else may do the same.

### **Has the GPL ever been upheld in court?**

No one has ever dared to challenge it in court.

A case has recently arisen that, if it ever reaches a courtroom, may resolve some of the legal issues. However this case does not question the validity of the GPL itself. Rather, it seeks to clarify the notion of a derivative work, as defined by the GPL. This notion has always been a little fuzzy, because it's hard for a static, generic legal document to anticipate all the twists and turns of technology.

### **But the GPL is about copyright, that is, the right to copy. Don't I also need a run time license?**

No. If you have a legal copy of a program, you may run the program – *unless* you have agreed to accept other conditions. Because the law does not require a run time license, proprietary software vendors typically contrive to extract your agreement to such a license, whether a shrink-wrap license, a click-through license, or a signed contract.

### **Are there any other disadvantages to the SCO license?**

First of all, there is the obvious issue of cost. You would give money to SCO without receiving any additional functionality.

In addition, the SCO license requires you to keep detailed records of your Linux systems and your SCO licenses, and to release that information to SCO at their request. You agree to let SCO audit your systems at any time, at SCO's expense. If the audit determines that your records aren't accurate enough, you must pay for the audit.

### **But at least SCO would promise not to sue me for infringement – right?**

Such a promise would be of doubtful value, since according to normal interpretations of the law, the end user would not be liable for infringement in any case. With or without SCO's license, such a suit by SCO would likely be ruled frivolous.

An SCO license may help you avoid such a lawsuit, which would be expensive and burdensome to fend off, even if frivolous. However, take a close look at the following provision in section 5.0 of the license:

```
SCO may terminate this Agreement, upon reasonable notice and
without judicial or administrative resolution, if Company or any
of Company's employees or consultants breach any term or condition
hereof.
```

As a non-lawyer I'm not sure what this verbiage means. On the face of it, it seems to say that SCO can, at its sole discretion, declare at any time that you have breached the Agreement, and terminate it accordingly.. You would have no right of appeal or other recourse, because you would have already waived all "judicial or administrative resolution." Then SCO could sue you as well as if you had never purchased a license. SCO's attempt to terminate IBM's Unix licenses demonstrates this kind of tactic almost precisely.

A statement issued by SCO has said that "Contracts are what you use against parties you have relationships with."

Do you really want a relationship with a company that has that kind of attitude?

Neither do I.

---

## References

<http://www.sco.com/ibmlawsuit> -- Copies of various filings in the SCO/IBM case, including SCO's original complaint, its amended complaint, and IBM's responses.

<http://www.opensource.org/halloween/halloween9.php> -- A detailed dissection of SCO's filing, by Rob Landley and Bruce Perens.

<http://www.opensource.org/sco-vs-ibm.html> -- OSI Position Paper on the SCO-vs-IBM Complaint, by Eric S. Raymond and Rob Landley.

<http://www.internetweek.com/story/showArticle.jhtml?articleID=9901350> -- SCO threatens to sue end users.

[http://news.com.com/2100-1016\\_3-999371.html](http://news.com.com/2100-1016_3-999371.html) -- News article quoting Darl McBride: "The Linux community would have me publish it now, (so they can have it) laundered by the time we can get to a court hearing."

<http://www.pcworld.com/news/article/0,aid,110904,00.asp> -- News article quoting an SCO statement about the use of contracts.

<http://www.internetweek.com/story/showArticle.jhtml?articleID=13000223> -- SCO announces plans to offer an SCO Intellectual Property License for Linux.

<http://perens.com/Articles/SCO/SCOSlideShow.html> -- Bruce Perens, among others, analyzes what SCO claims to be examples of pirated code.

<http://www.catb.org/~esr/writings/smoking-fizzle.html> -- Eric S. Raymond analyzes what SCO claims to be an example of pirated code.

<http://www.lemis.com/grog/SCO/code-comparison.html> -- Greg Lehey's analysis of what SCO claims to be examples of pirated code.

<http://www.internetwk.com/breakingNews/showArticle.jhtml;jsessionid=MELSVVCOMGMYYQQSNDBC SKHQ?articleID=13900143&pgno=1> -- Chris Sontag of SCO backpedals on the claims of obfuscated BPF code.

<http://www.bayarea.com/mld/mercurynews/6238207.htm> -- An interview with Linus Torvalds.

<http://www.eweek.com/article2/0,3959,1227128,00.asp> -- Another interview with Linus Torvalds, noteworthy for his "smoking crack" remark.

<http://www.nwfusion.com/news/2003/0825scoatta.html> -- Interview with Darl McBride, where he accuses IBM of orchestrating the response of the open source community to SCO's actions.

<http://newsforge.com/newsforge/03/08/22/1746248.shtml?tid=19> -- Eric S. Raymond's angry response to McBride's accusations (above).

<http://www.sco.com/company/openletter/> -- Darl McBride's so-called Open Letter to the Open Source Community, in which he misquotes Bruce Perens, and accuses Linux developers of a "don't ask, don't tell" policy on intellectual property rights.

<http://www.computerworld.com/softwaretopics/os/linux/story/0,10801,84819,00.html> -- An interview with Darl McBride, following his Open Letter.



<http://newsforge.com/newsforge/03/09/10/2321224.shtml?tid=11> -- Linus Torvald's laconic response to McBride's Open Letter.

<http://newsforge.com/newsforge/03/09/09/2355214.shtml?tid=11> -- Eric S. Raymond and Bruce Perens respond to McBride's Open Letter.

<http://www.theinquirer.net/?article=11663> -- Another response to McBride's open letter, from a group of open source advocates.

<http://cm.bell-labs.com/cm/cs/who/dmr/bsdi/930303.ruling.txt> -- A 1993 court ruling denying AT&T injunctive relief on the basis of copyright issues.

<http://www.gnu.org/licenses/gpl.html> -- Required reading: the GNU General Public License.

[http://story.news.yahoo.com/news?tmpl=story&u=/techtarget/20030827/tc\\_techtarget/921474](http://story.news.yahoo.com/news?tmpl=story&u=/techtarget/20030827/tc_techtarget/921474) -- An interview with a lawyer about the intellectual property issues surrounding SCO's claims.

[http://www.osdl.org/docs/osdl\\_eben\\_moglen\\_position\\_paper.pdf](http://www.osdl.org/docs/osdl_eben_moglen_position_paper.pdf) -- Eben Moglen, a Columbia University law professor, discusses SCO's claims.

<http://www.gnu.org/philosophy/sco/sco-preemption.html> -- Eben Moglen again, countering SCO's claim that the GPL is legally invalid.

[http://www.osdl.org/docs/qa\\_re\\_sco\\_vs\\_ibm\\_html.html](http://www.osdl.org/docs/qa_re_sco_vs_ibm_html.html) -- Another lawyer, Lawrence Rosen, discusses the legal issues.

<http://lwn.net/Articles/43085/> --The text of the SCO Intellectual Property License for Linux.

<http://www.opengroup.org/openbrand/register> -- A list of registered products complying with Open Group standards.

<http://www.thejemreport.com/articles/sco.htm> -- Compares various flavors of Unix and Unix-like operating systems, and describes their various degrees of compliance with Open Group standards.

[http://www.eweek.com/print\\_article/0,3668,a=43186,00.asp](http://www.eweek.com/print_article/0,3668,a=43186,00.asp) -- Allegations that SCO has violated the GPL by copying Linux code into the Linux Kernel Personality product.

### **Metareferences**

The following sites include many links to SCO-related resources.

<http://www.groklaw.com/>

<http://groups.yahoo.com/group/no2sco/links>

<http://www.theinquirer.net/?article=11649>

<http://www.onlamp.com/pub/wlg/3702>

Unix and UnixWare are trademarks of the Open Group. AIX is a trademark of IBM. Linux is a trademark of Linus Torvalds. OpenServer is a trademark of the SCO Group.

*Copyright 2003 by Scott McKellar (mck9@swbell.net). You may reproduce this document in its entirety in any medium without restriction, provided that you preserve this notice.*