SCO Forum2003 WORLDWIDE

SCO*source* Update

Chris Sontag, Senior Vice President, SCO*source*
Mark J. Heise, Partner, Boies, Schiller & Flexner, LLP

Agenda

- SCO's ownership of the UNIX® Operating System
- Significant UNIX code is in Linux
- Linux is an unauthorized derivative of UNIX

Contracts, Agreements, and the Law

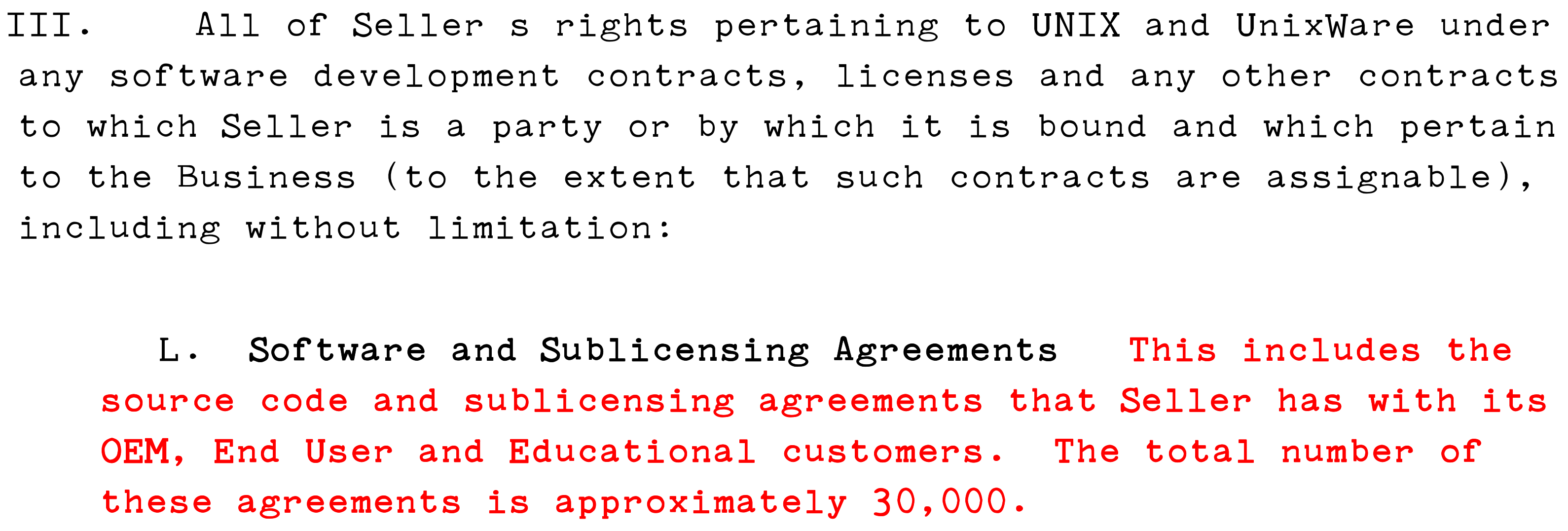
SCO Owns All UNIX System V Source Code

All rights and ownership of UNIX and UnixWare, including but not limited to all versions of UNIX and UnixWare and all copies of UNIX and UnixWare (including revisions and updates in process) and all technical, design, development, installation, operation and maintenance information concerning UNIX and UnixWare, including source code, source documentation, source listings and annotations, appropriate engineering notebooks, test data and test results, as well as all reference manuals and support materials normally distributed by Seller to end-users and potential end-users in connection with the distribution of UNIX and UnixWare, such assets to include without limitations the following:

Schedule 1.1(a) Para. I of Asset Purchase Agreement with Novell

Contracts, Agreements, and the Law

SCO Owns All AT&T Software and Sublicensing Agreements to All UNIX Vendors



III. All of Seller s rights pertaining to UNIX and UnixWare under any software development contracts, licenses and any other contracts to which Seller is a party or by which it is bound and which pertain to the Business (to the extent that such contracts are assignable), including without limitation:

L. **Software and Sublicensing Agreements** This includes the source code and sublicensing agreements that Seller has with its OEM, End User and Educational customers. The total number of these agreements is approximately 30,000.

Schedule 1.1(a), III, L of Asset Purchase Agreement with Novell

Contracts, Agreements, and the Law

SCO Owns All UNIX System V Copyrights

With respect to Schedule 1.1(b) of the agreement, titled Excluded Assets , Section V, Subsection A shall be revised to read:

> All copyrights and trademarks, except for the copyrights and trademarks owned by Novell as of the date of the agreement required for SCO to exercise its rights with respect to the acquisition of UNIX and UnixWare technologies.

Amendment No. 2 of Asset Purchase Agreement with Novell

Contracts, Agreements, and the Law

AT&T Software Agreement Defines Scope of License

- AT&T grants to LICENSEE a personal, nontransferable and nonexclusive right to use in the United States each SOFTWARE PRODUCT solely for LICENSEE s own internal business purposes (¶2.01)
- Such right to use includes the right to modify such SOFTWARE PRODUCT and to prepare derivative works based on such SOFTWARE PRODUCT, provided the resulting materials are treated hereunder as part of the original SOFTWARE PRODUCT. (¶2.01)
- No right granted for use of SOFTWARE PRODUCTS by IBM directly for others, or use by others. (¶2.05)

(AT&T Software Agreement, Section II)

AT&T Software Agreement Defines Scope of License

- LICENSEE agrees that it shall hold all parts of the SOFTWARE PRODUCTS in confidence for SCO (including **methods**) (¶7.06 a)

- **Neither this Agreement nor any rights hereunder shall be assignable or otherwise transferable by LICENSEE and any purported assignment or transfer shall be null and void.** (¶7.09)

- **Nothing in this Agreement grants to LICENSEE the right to sell, lease or otherwise transfer or dispose of a SOFTWARE PRODUCT in whole or in part.** (¶7.10)

Contracts, Agreements, and the Law

SCO UNIX System V Copyright Infringements in Linux®

Literal Copying Line-for-line code copied from System V into Linux kernels 2.4+	**Derivative Works** Modifications of System V created by vendors contributed to Linux kernels 2.4+in violation of contracts
Obfuscation Copying, pasting, removing legal notices, reorganizing the order of the programming structures	**Non-literal transfers** Methods, structures and sequence from System V contributed to Linux kernels 2.4+

Linux is a registered trademark of Linus Torvalds

Literal Copying

Direct System V Code Has Been Copied Into Linux Kernel Releases 2.4x and 2.5x

- Line-for-line copied System V code
- Example being presented is just one of many
- Copied code includes typographical error in comments
- Dun & Bradstreet v. Grace Consulting (3rd Cir. 2002)

Line by Line Copying — One Example of Many

System V Code

```
/* Copyright (c) 1990, 1991 UNIX System
Laboratories, Inc.      */
/* Copyright (c) 1984, 1986, 1987, 1988, 1989,
1990 AT&T  */
/* All Rights Reserved         */

/*    THIS IS UNPUBLISHED
PROPRIETARY SOURCE CODE OF           */
/*    UNIX System Laboratories, Inc.           */
/*    The copyright notice above does not
evidence any  actual or intended publication of
such source code.      */

#ιδεντ      ∀≅(#)υτσ–χομμ:υτιλ/ξξξξξ.χ
1.3∀
#ινχλυδε <υτιλ/παραμ.η>
#ινχλυδε <υτιλ/τψπεσ.η>
#ινχλυδε <σϖχ/σψστμ.η>
…
```

Linux Kernel Code

```
/* This file is subject to the terms and
conditions of the GNU General Public

 * License.  See the file "COPYING" in the
main directory of this archive for more
details.

 * Copyright (C) 1992 - 1997, 2000-2002
xxxxxxxx, Inc. All rights reserved.


 */



#include <linux/types.h>
#include <linux/slab.h>
#include <asm/sn/xxx.h>
#include <asm/sn/addrs.h>
…
```

Line by Line Copying — One Example of Many

System V Code

```
/*
 * Allocate 'size' units from the given map.
 * Return the base of the allocated space.
 * In a map, the addresses are increasing and the
 * list is terminated by a 0 size.
 * The swap map unit is 512 bytes.
 * Algorithm is first-fit.
 *
 * Ασ παρτ οφ τηε κερνελ εϖολυτιον τοωαρδ μοδυλαρ ναμινγ, τηε
 * φυνχτιονσ μαλλοχ ανδ μφρεε αρε βεινγ ρεναμεδ το ρμαλλοχ ανδ ρμφρεε.
 * Χομπατιβιλιτψ ωιλλ βε μαινταινεδ βψ τηε φολλοωινγ ασσεμβλερ χοδε:
 * (αλσο σεε μφρεε/ρμφρεε βελοω)
 */

…
```

Linux Kernel Code

```
/*
 * Allocate 'size' units from the given map.
 * Return the base of the allocated space.
 * In a map, the addresses are increasing and the
 * list is terminated by a 0 size.
 * Algorithm is first-fit.
 */

ulong_t
atealloc(
       struct map *mp,
      size_t size)
{
      register unsigned int a;
      register struct map *bp;
      register unsigned long s;

…
```

Line by Line Copying — One Example of Many

Linux Kernel Code

```
if (size == 0)
        return) ((ulong_t NULL);

    s = mutex_spinlock(maplock(mp));

    for (bp = mapstart(mp); bp->m_size; bp++) {
        if (bp->m_size >= size) {
            a = bp->m_addr;
            bp->m_addr += size;
            if ((bp->m_size -= size) == 0) {
                do {
                    bp++;
                    (bp-1)->m_addr = bp->m_addr;
                } while ((((bp-1)->m_size) = (bp->m_size)));
                mapsize(mp)++;
            }

            ASSERT(bp->m_size < 0x80000000);
…
```

Line by Line Copying — One Example of Many

Linux Kernel Code

```
/*
* Free the previously allocated space a of size units into the specified map.
* Sort ``a'' into map and combine on one or both ends if possible.
* Returns 0 on success, 1 on failure.
*/

void

atefree(struct map *mp, size_t size, ulong_t a)

{

     register struct map *bp;
     register unsigned int t;
     register unsigned long s;

     ASSERT(size >= 0);

…
```

Line by Line Copying — One Example of Many

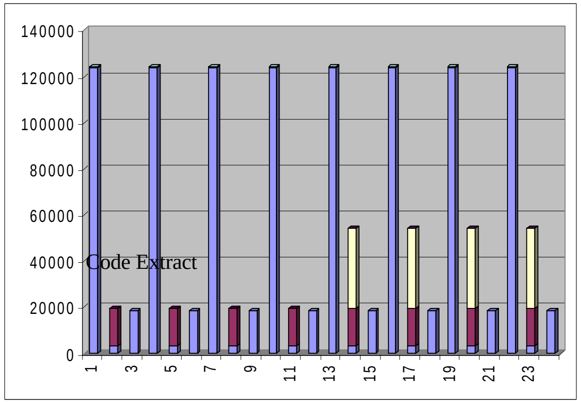
Linux Kernel Code

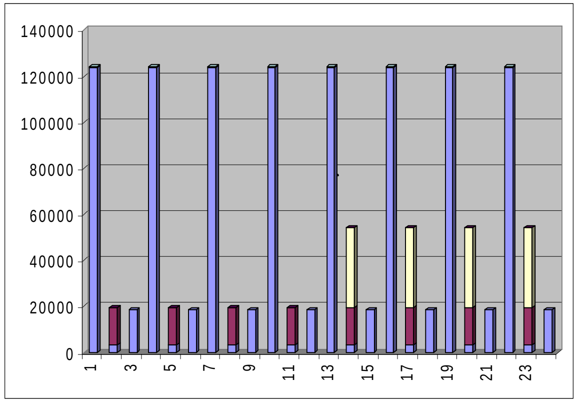
```
if (size == 0)
    return;

bp = mapstart(mp);
s = mutex_spinlock(maplock(mp));

for ( ; bp->m_addr<=a && bp->m_size!=0; bp++)
    ;
if (bp>mapstart(mp) && (bp-1)->m_addr+(bp-1)->m_size == a) {
    (bp-1)->m_size += size;
    if (bp->m_addr) {
        /* m_addr==0 end of map table */
        ASSERT(a+size <= bp->m_addr);
        if (a+size == bp->m_addr) {

            /* compress adjacent map addr entries */
            (bp-1)->m_size += bp->m_size;
            while (bp->m_size) {
```

…

Obfuscated Copying

Obfuscated System V Code Has Been Copied Into Linux Kernel Releases 2.4x and 2.5x





System V Code

```
  πχ += (Α == πχ−>κ) ? πχ−>φτ : πχ−>φφ;
      χοντινυε;
χασε ΒΠΦ_ϑΜΠ|ΒΠΦ_ϑΣΕΤ|ΒΠΦ_Κ:
      πχ += (Α & πχ−>κ) ? πχ−>φτ : πχ−>φφ;
      χοντινυε;
χασε ΒΠΦ_ϑΜΠ|ΒΠΦ_ϑΓΤ|ΒΠΦ_Ξ:
      πχ += (Α > Ξ) ? πχ−>φτ : πχ−>φφ;
      χοντινυε;
χασε ΒΠΦ_ϑΜΠ|ΒΠΦ_ϑΓΕ|ΒΠΦ_Ξ:
      πχ += (Α >= Ξ) ? πχ−>φτ : πχ−>φφ;
      χοντινυε;
χασε ΒΠΦ_ϑΜΠ|ΒΠΦ_ϑΕΘ|ΒΠΦ_Ξ:
      πχ += (Α == Ξ) ? πχ−>φτ : πχ−>φφ;
      χοντινυε;
χασε ΒΠΦ_ϑΜΠ|ΒΠΦ_ϑΣΕΤ|ΒΠΦ_Ξ:
      πχ += (Α & Ξ) ? πχ−>φτ : πχ−>φφ
```

Linux Kernel Code

```
 pc += (A == fentry->k) ? fentry->jt : fentry->jf;
       continue;
case BPF_JMP|BPF_JSET|BPF_K:
       pc += (A & fentry->k) ? fentry->jt : fentry->jf;
       continue;
case BPF_JMP|BPF_JGT|BPF_X:
       pc += (A > X) ? fentry->jt : fentry->jf;
       continue;
case BPF_JMP|BPF_JGE|BPF_X:
       pc += (A >= X) ? fentry->jt : fentry->jf;
       continue;
case BPF_JMP|BPF_JEQ|BPF_X:
       pc += (A == X) ? fentry->jt : fentry->jf;
       continue;
case BPF_JMP|BPF_JSET|BPF_X:
       pc += (A & X) ? fentry->jt : fentry->jf;
```

Derivative Works

SCO has Exclusive Right to Authorize all UNIX System V Derivative Works

17 U.S.C. 106: Exclusive rights in copyrighted works

The owner of copyright under this title has the exclusive rights to do and to authorize any of the following:

to prepare derivative works based upon the copyrighted work;

Derivative Works

Software Agreement Grant of Rights Controls Use of System V Derivative Works

2.01 AT&T grants to Licensee a personal, nontransferable and nonexclusive right to use Software Product solely for Licensee s own internal business purposes and solely on or in conjunction with Designated CPUs for such Software Product. **Such right to use includes the right to modify such software Product and to prepare derivative works based on such Software Product, provided the resulting materials are treated hereunder as part of the original Software Product.**

Derivative Works

Use is Infringing If Scope of License Grant for Derivative Work is Exceeded

Stewart v. Abend (US 1990)

- [U]se is infringing if one who employs the [derivative] work does not have a valid license or assignment for use of the pre-existing work

Gilliam v. ABC (2nd Cir. 1999)

- One who obtains permission to use copyrighted [work] in production of derivative work ***may not exceed*** specific purpose for which permission is granted
- Transfer in excess of license was a ***nullity***

Liu v. Price Waterhouse (7th Cir.)

- Copyright in derivative work used outside scope of license grant reverted to owner of original program

Derivative Works

Examples of Significant Infringing Derivative Works Contributions to Linux 2.4/2.5 Kernels

- NUMA
- RCU
- JFS
- XFS
- Schedulers
- LinuxPPC 32- and 64-bit support
- Enterprise Volume Management System

Derivative Works

- RCU
  - 46 Files
  - 109,688 Lines
- NUMA
  - 101 Files
  - 56,587 Lines
- JFS
  - 44 Files
  - 32,224 Lines
- XFS
  - 173 Files
  - 119,130 Lines
- SMP
  - 1,185 Files
  - 829,393 Lines

TOTAL CURRENTLY KNOWN

- 1,549 Files
- 1,147,022 Lines

Derivative Works

IBM Claimed Copyright Attribution for Transferring Dynix Code to Linux

Copyright (c) International Business Machines Corp., 2001 *This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License* as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA. Author: Dipankar Sarma <dipankar@sequent.com> (Based on a Dynix/ptx implementation by Paul Mckenney <paul.mckenney@us.ibm.com>)

See: http://lse.sourceforge.net/locking/rcu/patches/rclock-2.4.1-01.patch

Non-Literal Transfers – Methods and Concepts

From: Niels Christiansen (nchr@us.ibm.com)

Date: Thu Dec 06 2001 - 11:10:47 EST

Hi Kiran,

Are you concerned with increase in memory used per counter Here? I suppose that must not be that much of an issue for a 16 processor box.... Nope, I'm concerned that if this mechanism is to be used for all counters, **the improvement in cache coherence might be less significant to the point where the additional overhead isn't worth it.**

...which may be true for 4-ways and even 8-ways but when you get to 32-ways and greater, you start seeing cache problems. That was the case on AIX and per-cpu counters was one of the changes that helped get the spectacular scalability on Regatta.

Niels Christiansen
IBM LTC, Kernel Performance

Summary

Improbable Linux Development Path

2.2 Linux kernel

Hobbyist level technology

2-way multi-processor capability

Moderate reliability

2.6 Linux kernel

Enterprise level technology

32-way multi-processor in SMP configurations

128-way multi-processor in NUMA configurations

Higher reliability

Multi-processor capabilities requires extremely high fault tolerances. Multi-processor memories requires "locking" at a fraction of a millisecond. These developments, among others, could not have been accomplished in a compressed time period without direct access to 25 years of UNIX development expertise and use of state-of-the-art Unix development labs.

SCO is Entitled to Copyright Damages Against All Infringers, Together With Contract Damages

504. Remedies for infringement: Damages and profits

- In General.--Except as otherwise provided by this title, an infringer of copyright is liable for
- the copyright owner's **actual damages and any additional profits of the infringer**, as provided by subsection (b).

(b) Actual Damages and Profits.--**The copyright owner is entitled to recover the actual damages suffered by him or her as a result of the infringement, and any profits of the infringer** that are attributable to the infringement and are not taken into account in computing the actual damages. In establishing the infringer's profits, the copyright owner is required to present proof only of the infringer's gross revenue, and the infringer is required to prove his or her deductible expenses and the elements of profit attributable to factors other than the copyrighted work.

Contracts, Agreements, and the Law

SCO is Entitled to Copyright Injunctive Relief Against All Infringers

502. Remedies for infringement: Injunctions

(a) Any court having jurisdiction of a civil action arising under this title may, subject to the provisions of section 1498 of title 28, grant temporary and final injunctions on such terms **as it may deem reasonable to prevent or restrain infringement of a copyright**.

SCO*source* Linux Licensing Program

- License is designed to provide immediate relief to Linux end-users.
- Provides an end-user license for SCO's UnixWare 7.1.3 product for use in conjunction with any Linux offering – structured as a binary, run-only, per CPU license.
- SCO will covenant to hold licensees harmless for running Linux in binary format on any CPU licensed under a valid SCO UnixWare 7.1.3 license.
- Licensing format assures Linux end-users full compliance with SCO's IP rights
- License does not waive any rights SCO has against IBM or others. License does not include source code, modification or distribution rights.
- Pricing is based on past and future Linux usage and number of Linux servers.

SCO

Forum2003

WORLDWIDE

SCO*source* Update

Chris Sontag, Senior Vice President, SCO*source*

Mark J. Heise, Partner, Boies, Schiller & Flexner, LLP