Views of Free Software History

- Free software has a fascinating, intricate history.
- This presentation can only skim the surface.

Disjointed outline of free software history and remarks

- Paul Elliott
- pelliott@BlackPatchPanel.com
- Not a academic history.
- I don't limit myself to primary sources.
- If you want a perfect history, get a real historian!
- This is a “philosophical history”; I will give you some opinions and remarks as we go along.
- `http://www.free.blackpatchpanel.com/pme/linux/history-bookmarks.html`
- `http://www.free.blackpatchpanel.com/pme/linux/history.pdf`

What is Free Software?

- Free Software is free as in “freedom”.
- As defined by the GNU project, Free Software includes the following freedoms:
  - “The freedom to run the program.”
  - “The freedom to study how the program works, and adapt it to your needs.”
  - “The freedom to redistribute copies so you can help your neighbor.”
  - “The freedom to improve the program, and release your improvements to the public.”

What is “source code”?

Use the Source Luke!

- “Source Code” is text that is written in a “computer language” such as COBOL, FORTRAN, C, or Java. Such code is designed to be readable and writable by trained human beings, any yet computer translatable into a form that can be “executed” (i.e. run) by a computer.

What is a Fork?

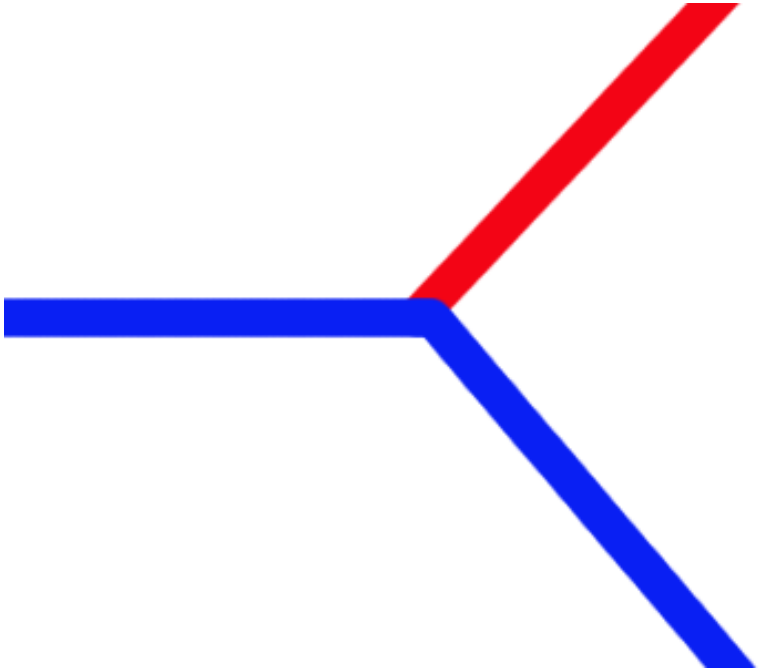
- A project fork is the creation of a separate software project from another software project, usually controlled by a new set of developers.

The Right to Fork is the Right to be Free!

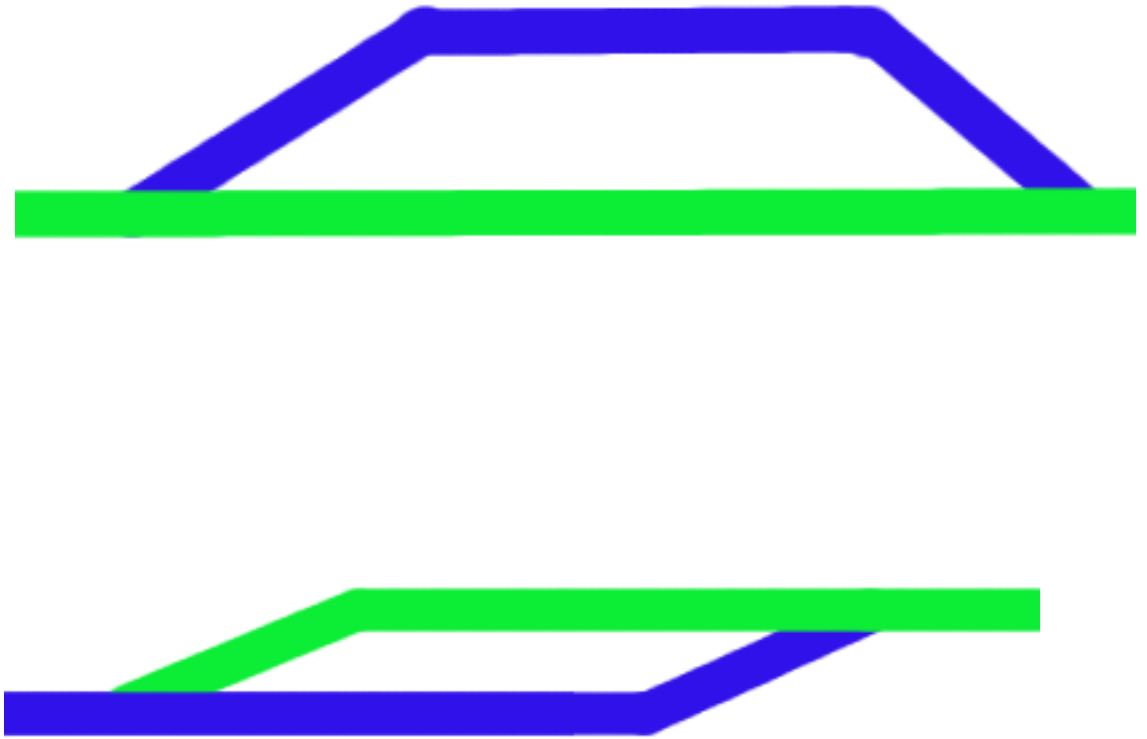
- apologies to A. E. Van Vogt.
- The right to fork is the primary right established by free software licenses.
- The threat of forks is what keeps developers from exploiting software users.

Forks are rare.

- Like the filibuster in the U.S. Senate, everyone knows the possibility is there, so forks only happen rarely.
- When a fork happens, the fork leaders are usually extremely apologetic, justifying themselves and explaining why the fork was necessary.



Forks sometimes heal!

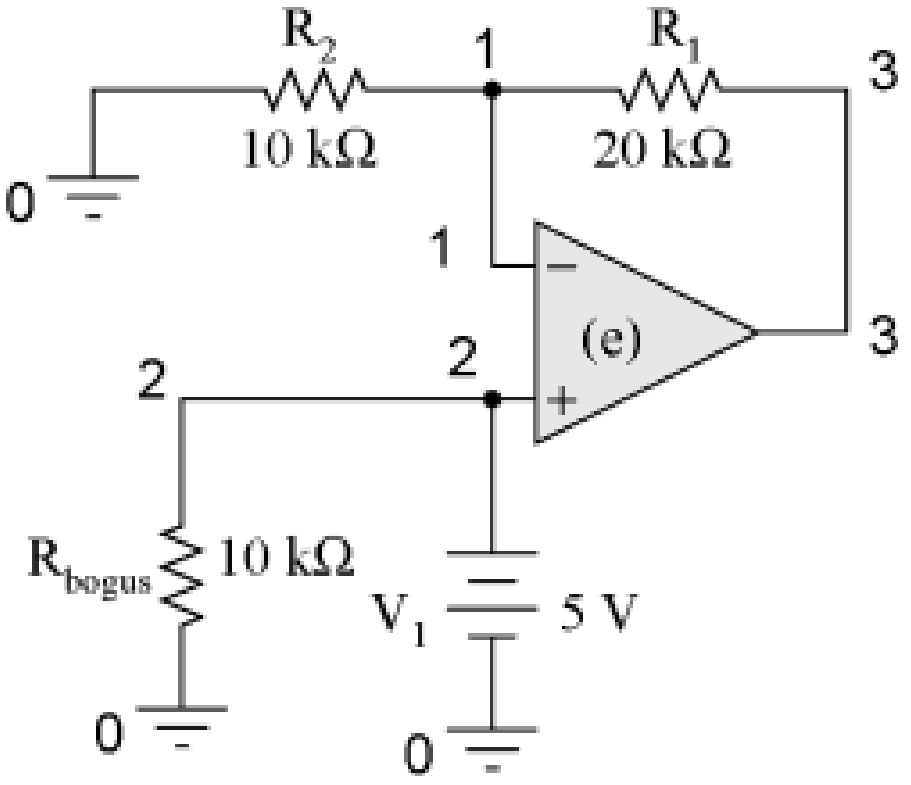


Forks and history.

- Just as in regular history, wars are the landmarks, in free software history, forks are the landmarks.
- This may be an artifact of a human defect which causes humans to focus on conflict!

Spice may have been first free software project.

- Circuit simulation program.
- In 1972, Nagel and Pederson released SPICE1 (a Simulation Program with IC Emphasis) into the public domain.
- Variants are still used today.

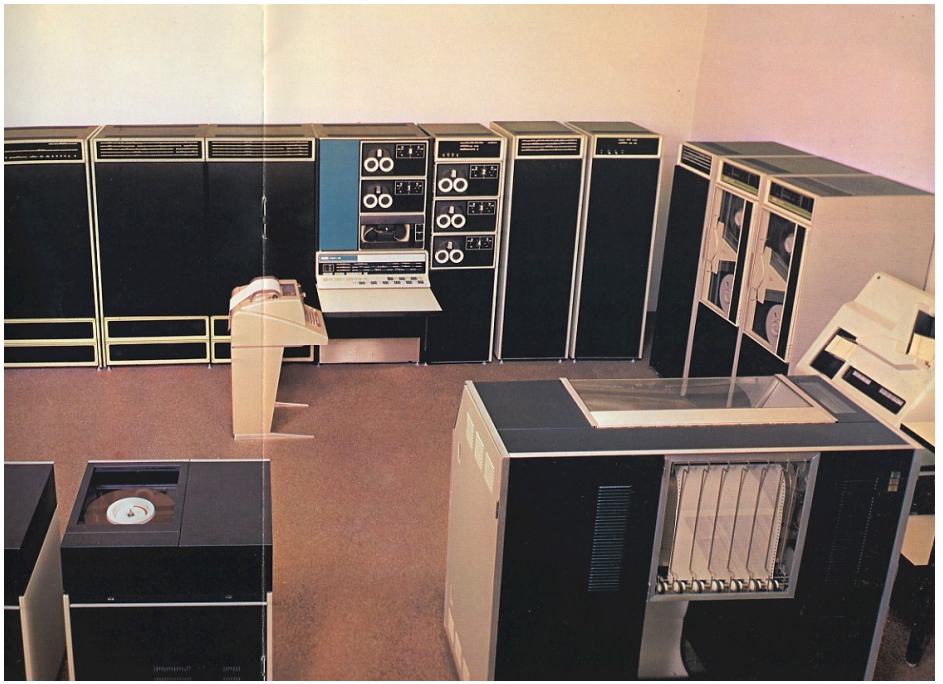


Donald E. Knuth created TeX.

- 1978
- Public domain except you cannot name your fork TeX, Megafont, or Computer Modern.
- Guy Steele ported to ITS!
- Now ported everywhere!

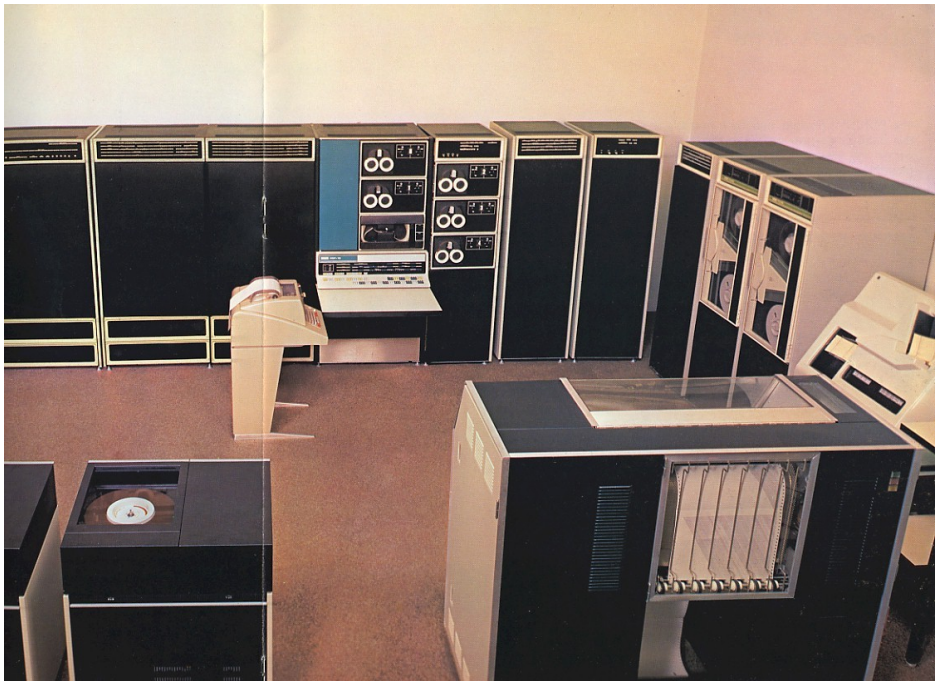
Incompatible timesharing system.

- Ran on PDP10.
- community formed around ITS at MIT artificial intelligence LAB
- programs were freely passed around, but few thought of public domain or licensing.
- RMS was there!



Incompatible timesharing system at MIT.

- was a hot bed of innovative software development activity.
- Many important projects, ideas and people got their start working with ITS.
- “ITS boasted features most commercial OSs would not offer for years.”



Experiences at MIT artificial intelligence LAB made rms into the rms we know today.

geek syndrome?

- Some, including rms, have suspected that Stallman may be a victim of high functioning autism or Aspeger syndrome.

The Xerox printer incident

- rms wanted to fix xerox printer driver to warn users of jammed printer.
- He had done this before with another printer.
- Driver was closed sourced by Xerox.

- Robert Sproull refused request for printer driver. Non-disclosure agreement.

community at MIT AI Lab was destroyed.

- LAB was raided by Lisp machines Inc. (LMI) and Symbolics.
- responsibility for the destruction goes to MIT administration, who should have upheld academic values rather than commercial values.

Where are they now?

- Where are Lisp machines Inc. (LMI) and Symbolics?
- What are their contributions to humanity?
- Compare with ITS community.

Light

- “Neither do men light a candle, and put it under a bushel, but on a candlestick; and it giveth light unto all that are in the house.”
- Christian religious writings.

rms fights the Symbolics war.

- Symbolics closed sourced the software used by MIT lab, hoping to lock out LMI.
- rms disconnects symbolics microwave communications link to MIT's lab.
- rms single handedly creates free versions of all lisp enhancements created by the symbolics team using only the documentation as a guide.
- rms duplicates the work of an entire team of programmers.

But all's too weak.

- “One day, while taking a break from writing code, Stallman experienced a traumatic moment passing through the lab's equipment room. There, Stallman encountered the hulking, unused frame of the PDP-10 machine. Startled by the dormant lights, lights that once actively blinked out a silent code indicating the status of the internal program, Stallman says the emotional impact was not unlike coming across a beloved family member's well-preserved corpse.”
- From “*Free as in Freedom*”

The Center cannot hold.

- "'I started crying right there in the equipment room,' he says. 'Seeing the machine there, dead, with nobody left to fix it, it all drove home how completely my community had been destroyed.'"
- from "*Free as in Freedom*"

Stallman resigns from MIT to begin the noble quest to create GNU.

- had to quit MIT so that MIT could not claim GNU as a work for hire.
- GNU project to include complete OS and suite of utilities including C compiler.
- GNU manifesto written as declaration of intent.

Scratching an itch?

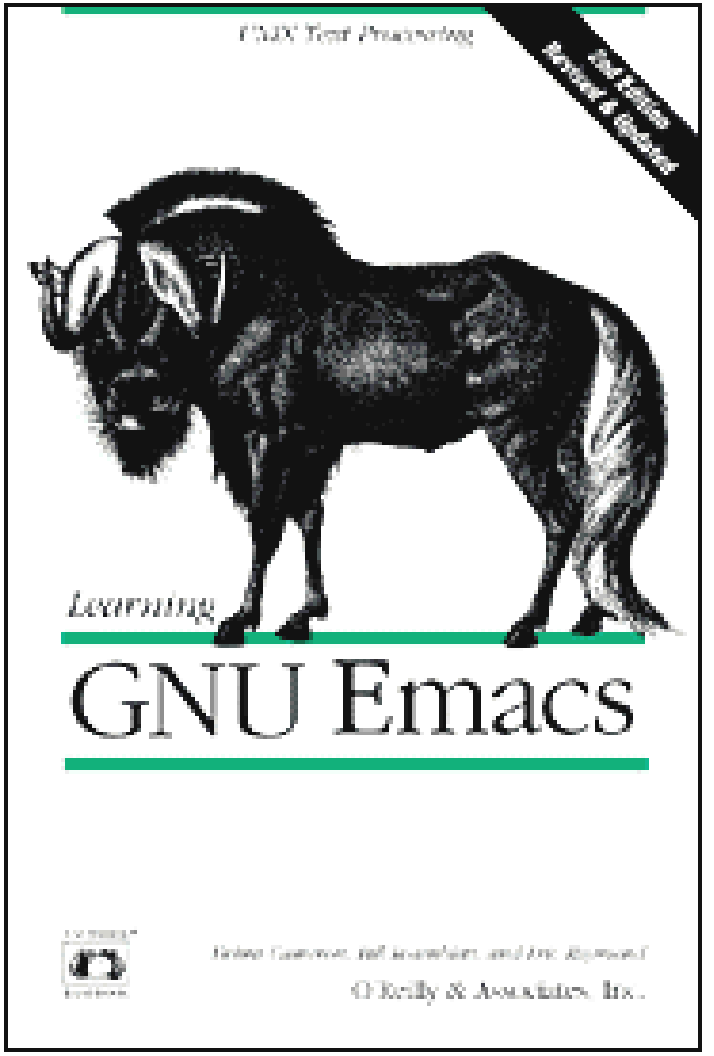
- “Eric Raymond says that 'Every good work of software starts by scratching a developer's personal itch.' Maybe that happens sometimes, but many essential pieces of GNU software were developed in order to have a complete free operating system. They come from a vision and a plan, not from impulse.”
- `http://www.gnu.org/gnu/thegnuproject.html`

rms begins the GNU project

- unsuccessfully searches for existing free C compiler.
- wrote GNU emacs.
- Founded the Free Software Foundation (FSF) to help with funding.

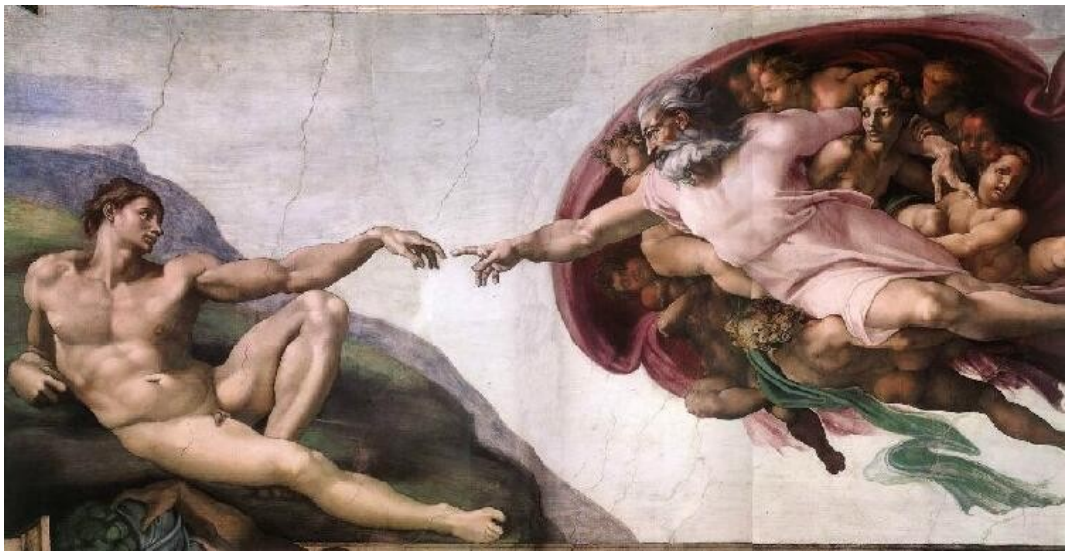
Emacs

- Replaced proprietary Gosling emacs.
- created new lisp interpreter as a foundation.
- version 15 of emacs was released under “the GNU emacs license”.



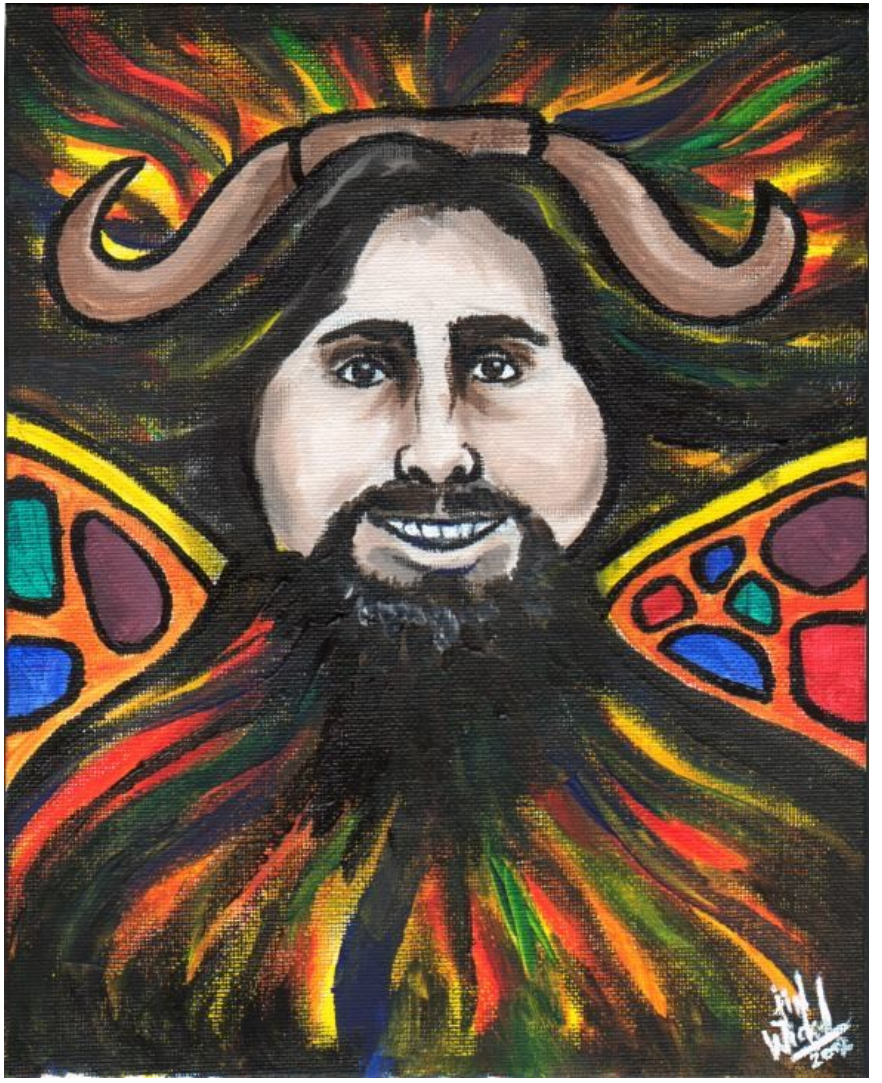
1989 version 1.0 of General Public License (GPL)

- evolved from GNU emacs license.
- creationism is false!
- successful software evolves. It does not spring fully formed from the Head of Zeus, like Palas Athena.
- successful software licenses are no different



Only rms had the unique vision and stubborn inflexibility to create the GPL.

- “"There isn't a lawyer on earth who would have drafted the GPL the way it is,' says Eben Moglen, Columbia University law professor and Free Software Foundation general counsel. 'But it works. And it works because of Richard's philosophy of design.'"
- `http://www.oreilly.com/openbook/freedom/ch13.html`



- By Jin Wicked

1987 RMS releases gcc under GPL.

- gcc is the foundation of all **free** “unix-like” OS efforts and many other projects.
- gcc has been ported to most all platforms; It is the foundation of portability.

By about 1990 almost all the GNU utilities had been released.

- GNU emacs, gcc, gdb
- bash, bison (was YACC), gawk (was awk).
- Only one piece of the puzzle remained to complete the GNU program.
- The **Operating System!**

GNU begins work on HURD

- Used a Mach microkernel. (more about mircokernels later).
- microkernel had to be debugged as well as kernel!
- 16 years later still no release!

Cygnus support, first free software support firm.

- rms made the GNU manifesto as a political statement, but Michael Tiemann and John Gilmore saw it as a business plan.
- They create Cygnus support, selling support for gcc and gnu utilities.

- Cygnus was assimilated by Red Hat 1999.

Meanwhile, back at the ranch....

- Unix (tm) was developed.

UNIX

Ken Thompson and Dennis Ritchie

- Created Unix as a work for hire while employed at .....

Bell Labs

- code originating at Bell Labs was always the property of ATT or it's successors in interest. It was not free software by any stretch of the imagination.

but,

- Code was licensed to various universities including the University of California at Bezerkley!
- New code began to be created independently there!

Gradually...

- Much ATT source code was replaced with Free code.
- Then, the serfs became restless from paying high ATT license fees...

Then Keith Bostic and others reduced the ATT contribution to BSD to just 6 files!

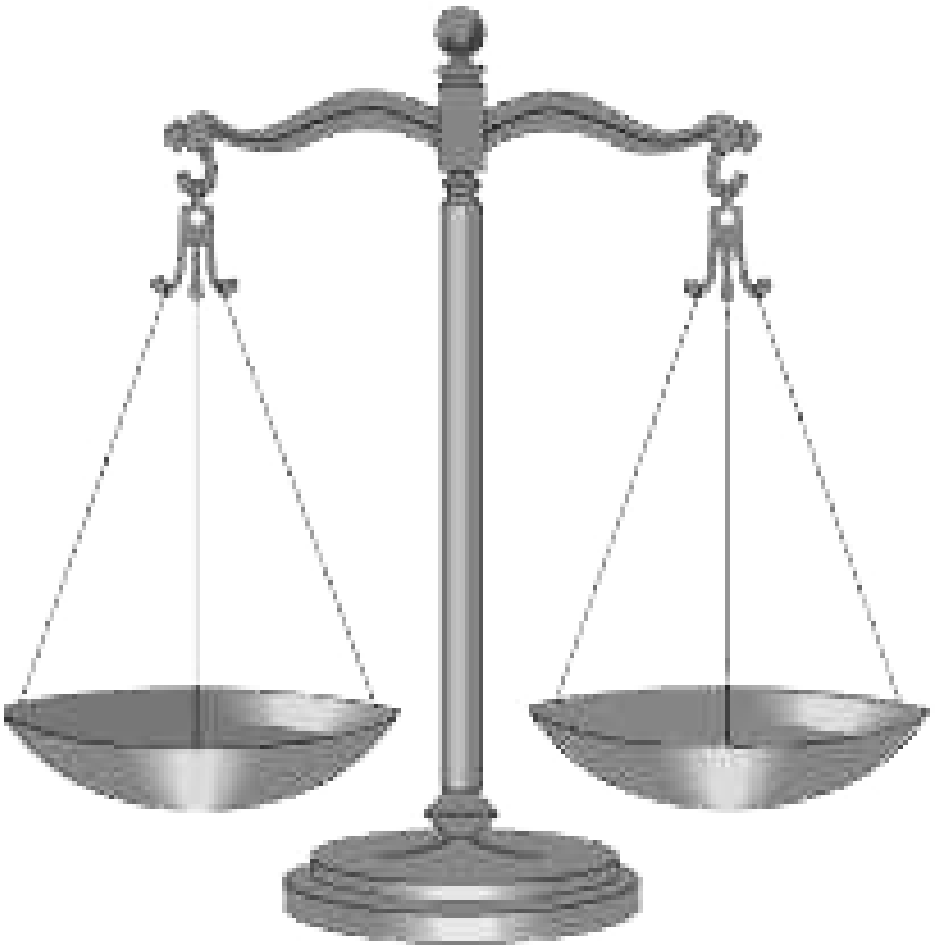
- This allowed the Computer Systems Research Group (CSRG) to release Networking Release 2.

This allowed Bill and Lynne Jolitz to fill the gap, writing the remaining 6 files!

- This allowed them to create a BSD distribution for the 386 completely free of ATT claims, which he called 386BSD.

But ATT's lawyer's did not agree!

- lawsuit between Berkeley Software Design, Incorporated (BSDI), Berkley and Unix System Laboratories (USL) (a mostly-owned subsidiary of AT&T spun off to develop and sell Unix) caused a 2 year delay in which the status of BSD was in doubt.



The final settlement resulted in freedom for 386/BSD.

- Some say the Settlement was a victory for BSDi with only a bone thrown to USL to protect their lawyers from embarrassment!
- In any case the end result was a BSD free of onerous entanglements with USL.

“Secret” settlement discovered and released by Grocklaw!

- Acquired document through California's public records act.
- Because settlement was never filed with any court, Grocklaw was not bound by the secrecy clause.
- `http://www.groklaw.net/article.php?story=20060713064700703&query=USL++BSDI+settlement`

Meanwhile BSD forked!

- In 1993, 386BSD forked into FreeBSD and NetBSD. In 1996, OpenBSD forked from NetBSD.

Why did not one of the BSDs assume the place of the dominant Free OS in the place of Linux?

- Many theories.
- Lawsuit delay.
- Loss of developers because of lawsuit.
- The forks.
- BSD projects tended to use “cathedral” style development, Linux used “bazaar” style.

NetBSD®

OpenBSD

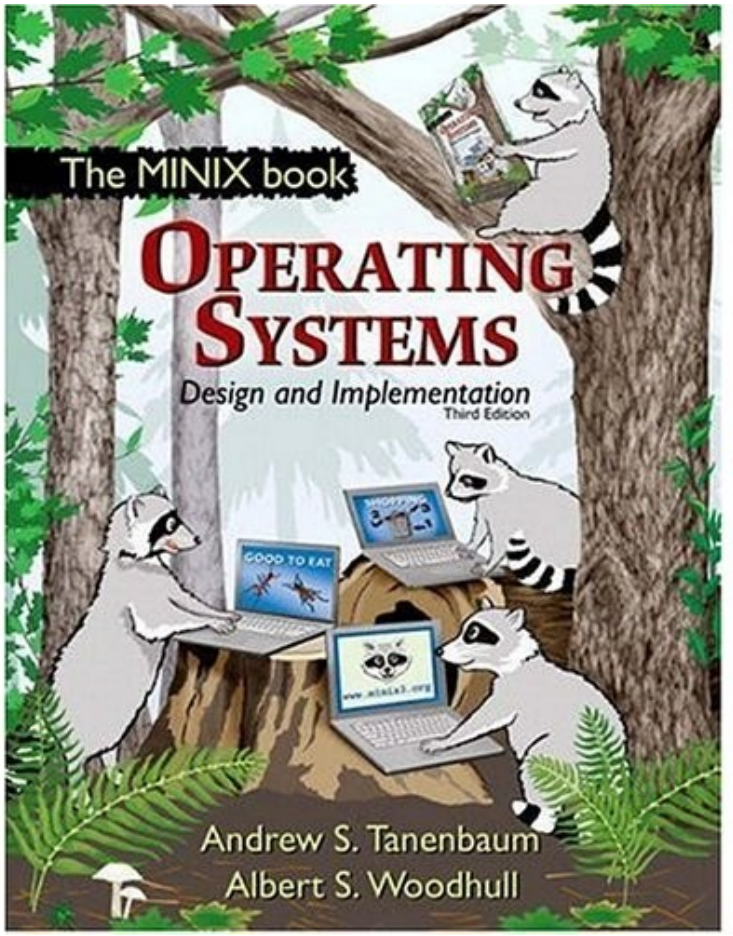
Many BSD advocates seem to be quite hostile toward Linux, even Today

Andrew Tanenbaum writes minix

- Written as a teaching tool, not free.
- Deliberately kept minimal for teaching purposes.
- Uses microkernel.
- Included gcc development environment.

Andrew Tanenbaum writes *Operating Systems: Design and Implementation*.

- Book is influential Operating System design circles.
- In the summer of 1990, the book “lived on Linus Torvalds' bed.”



Linus Torvalds begins some bare metal experiments on a 386. Apr 1991

Linus asks for the POSIX standards.

- In June 1991, Linus made a usenet request for the POSIX standards.
- There is only one reason that he would ask for these; He was thinking of an POSIX compliant Operating System!

Linus asks for suggestions on comp.os.minix (25 Apr 91)

- I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).
- I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months [...] Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT protable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-("

He planed to call his system “Freax”

- He was using Minix as a development system to cross compile his code, until Linux was advanced enough to compile and link itself.
- He included no code from Minix.

No LINUX?

- Recently, during FISL 6.0 (Fórum Internacional de Software Livre), in Brazil, Marshall Kirk McKusick was probably the main atraction for BSD community. Gentle and kind, he wore the I BSDCon Brazil shirt and talked about Unix history, open source model of development, BSD license, Beastie, Matt Dillon and others BSDs. Indeed, he still revealed unusual Linus Torvalds confesion, the kernel Linux creator.
- According to McKusick, one of the FreeBSD creators, Linus said: "I wouldn't have to create Linux if there wasn't any lawsuit against BSD in the early 90`s. It wouldn't be necesary".
- Linux version 0.11 when lawsuit started
- `http://www.myfreebsd.com.br/static/mckusick-20050608.html`

Sep 1991 Linux version . 01released!

- Intended for reading only!
- You were stuck with a Finnish keyboard!
- Remember, Linus Released it Just for Fun!

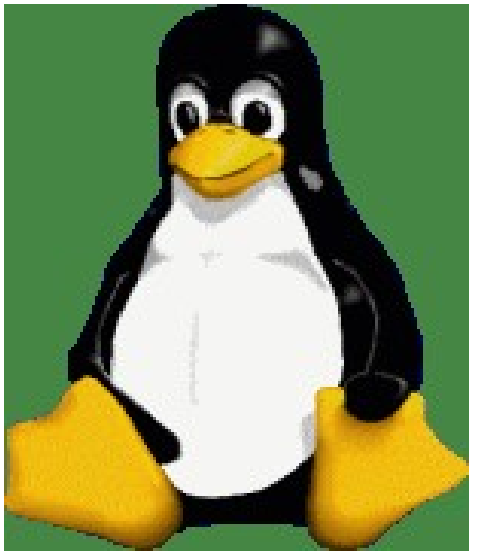
Tanenbaum-Torvalds microkernel vs monolithic kernel Debate came in two waves!

- First wave was in 1992. Tanenbaum claims “LINUX is obsolete”.
- Second wave was in 2006. Linus claims “The whole 'microkernels are simpler' argument is just bull,...”
- Linus lost 1st argument and won the 2nd.

Big picture

- Microkernels are a form of Bondage and Discipline Programming.
- 14 years ago in 1992, Tanenbaum said: “Linux is obsolete” and “it's all over but the shoutin”.
- 14 years later Minix 3 does not compete with Linux in its own huge problem space. Neither does any other microkernel design.
- It is time for microkernel advocates to come up with something that works on real world problems or stop shouting!

Linux and GNU join forces.



- In 1992, the GNU project adapted GNU utilities for use with Linux, while they waited for the Hurd. (long wait).
- Linus released Linux under the GPL. “Best decision I ever made” -- Linus

Could rms have created the OS portion of GNU/Linux?

- NO!
- HURD failed for many reasons, some of which are rms.
- OS projects, if developed quickly require the “bazaar” model (other develpers)
- Projects headed by rms tend to fork!

- Examples: Lucid emacs, xemacs, egcs, glibc
- Rms believes in the microkernel

Could Linus have created the GNU portion of GNU/Linux?

- NO!
- Linus is not a compiler developer, he could not have created gcc.
- Linus is a talented OS designer, but he does not have the vision necessary to create the GPL, a necessary component of the development model.
- Linus is not that crazy.

- Only Stallman was crazy enough to create the GNU project. He planned to create compiler, runtime library, utilities, the symbolic debugger, OS – the whole smear. This is clearly crazy!

libc forks from glibc.

- When GNU and Linux were learning to live together, when dinosaurs roamed the earth, some Linux developers thought that glibc would be a good C library for gcc running on the Linux OS. However, they thought it should be adapted for Linux as a separate project, so they forked it! They called the fork libc.
- glibc was then at version 1.x.

Both branches develop.

- libc develops some features and bug fixes in versions 2.x, 3.x, 4.x, 5.x.
- FSF'S glibc developed some advantages: Better downward compatibility, multiple language support, multi-thread support!
- The Linux programmers realized that the fork had been a strategic mistake. They decided to merge their changes back into glibc.

libc merges back into glibc.

- But glibc was still released as version 1.x and version 2.0 was a fixin to be released.
- “The version numbers were a minor problem: The GNU/Linux guys had already reached 5.4.47, while FSF was just hitting 2.0. They probably pondered for about a millisecond asking Stallman to make his next version 6.0 for their benefit. Then they laughed, said "This is Stallman we're talking about, right?", and decided out-stubborning Richard was not a wise idea. So, the convention is that Linux libc version 6.0 is the same as glibc 2.0.”
- from *Fear of Forking*, Rick Moen

XFree86 added to Linux.

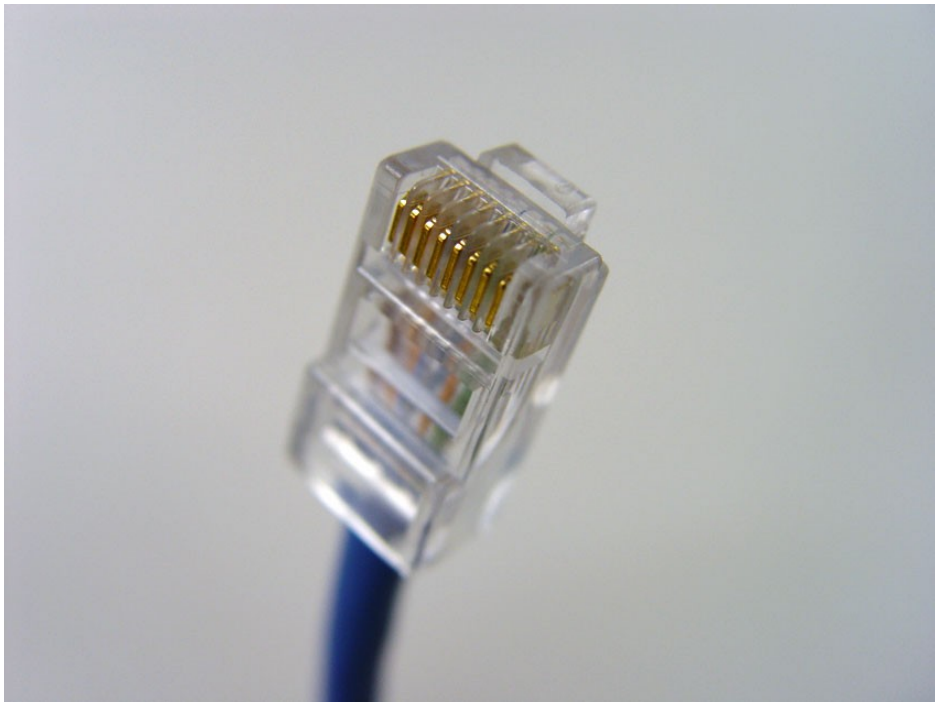
- Orest Zborowski writes a patch adding Unix sockets to Linux. This allows Linux to run Xfree86.
- XFree86 was forked from X consortium code because the license was too restrictive. 1992

X.org forks from XFree86

- Xfree86 development had become too cathedral-like.
- New licensing restrictions upset a number of people, including FSF. License may have been incompatible with GPL.
- So X.org forked from Xfree86. 2004.

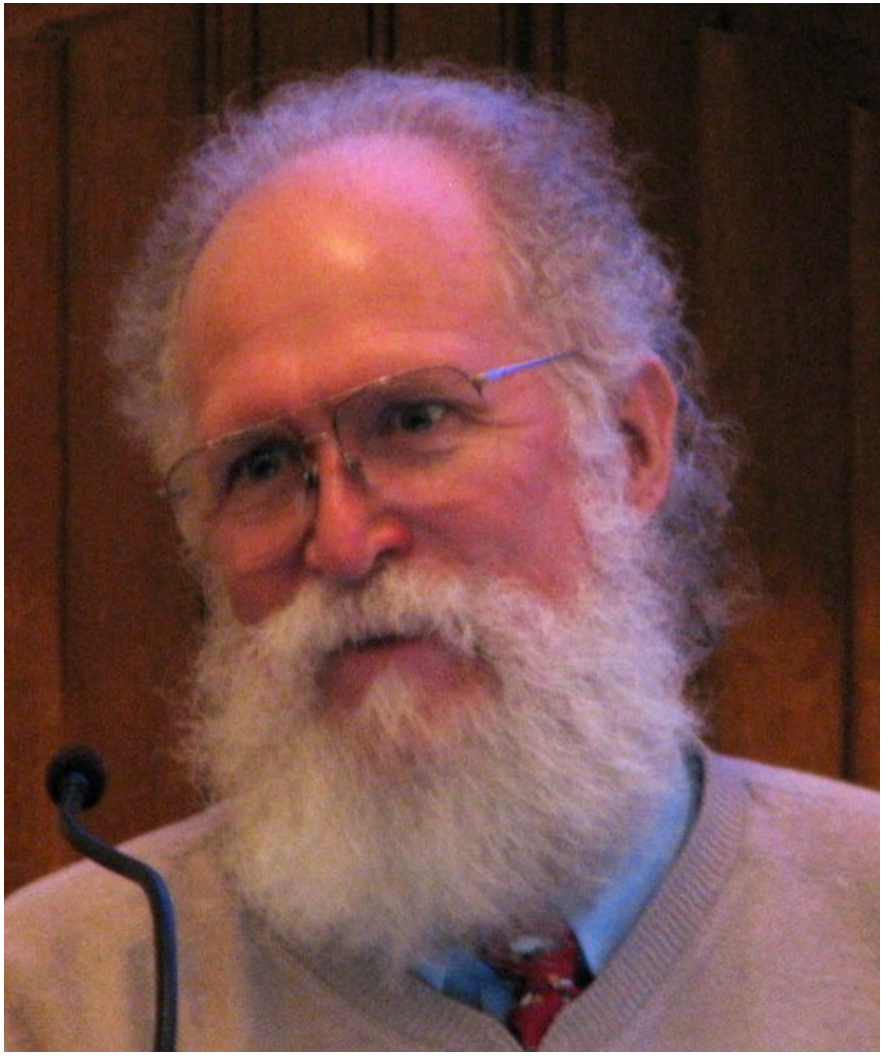
Linux connects to the internet!

- Fred van Kempen begins working on Linux Internet code. Net-2. 1993
- Alan Cox forks this code, debugging it.
- Linus accepted Alan's fork, which became standard.



Linux ported to DEC alpha

- Jon “Mad Dog” Hall gets DEC to loan Linus a DEC alpha.
- Linus ports Linux to DEC alpha.
- Today Linux is probably ported to more architectures than any other OS.
- alpha, mac, power PC, x86, x86-64, Crusoe, sun sparc, others.



David Miller ports Linux to sun sparc

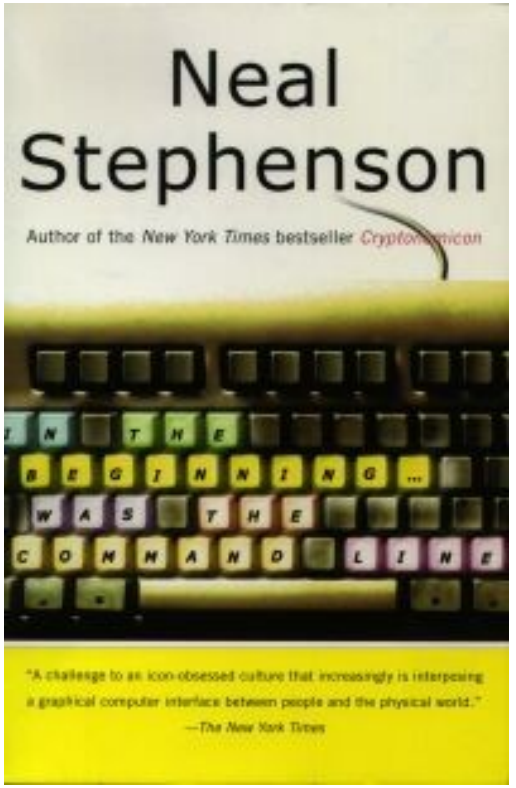
- 1999 David Miller ports Linux to sun sparc
- Sun corporate opposed this port, but they could not stop it.

Distros begin (1994)

- Various distros begin in 1994. Debian, slackware, SuSE, and Red Hat.
- Linus collects a lot of money from Red Hat stock options.

The reason people do not use LINUX.

- A 1999 quote from Neal Stephenson's book *“In the beginning was the command line”* explains why people do not use Linux.
- Hacker with bullhorn: "Save your money! Accept one of our free tanks! It is invulnerable, and can drive across rocks and swamps at ninety miles an hour while getting a hundred miles to the gallon!"
- Prospective station wagon buyer: "I know what you say is true...but...er...I don't know how to maintain a tank!"
- Bullhorn: "You don't know how to maintain a station wagon either!"
- Buyer: "But this dealership has mechanics on staff. If something goes wrong with my station wagon, I can take a day off work, bring it here, and pay them to work on it while I sit in the waiting room for hours, listening to elevator music."
- Bullhorn: "But if you accept one of our free tanks we will send volunteers to your house to fix it for free while you sleep!"

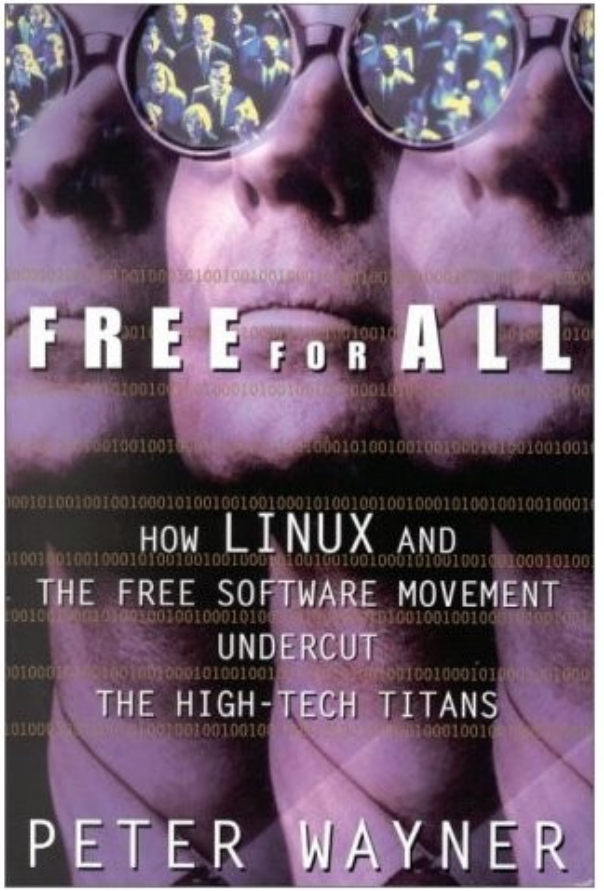


- Buyer: "Stay away from my house, you freak!"
- Bullhorn: "But..."
- Buyer: "Can't you see that everyone is buying station wagons?"

Free for all

- 2000, written by Peter Wayner

- "How many open source developers does it take to change a lightbulb?" The answer is: 17. Seventeen to argue about the license; 17 to argue about the brain-deadedness of the lightbulb architecture; 17 to argue about a new model that encompasses all models of illumination and makes it simple to replace candles, campfires, pilot lights, and skylights with the same easy-to-extend mechanism; 17 to speculate about the secretive industrial conspiracy that ensures that lightbulbs will burn out frequently; 1 to finally change the bulb; and 16 who decide that this solution is good enough for the time being.



2000

The Y2K disaster did not happen.

EGCS fork

- EGCS(Experimental/ Enhanced GNU Compiler System) was a fork from gcc.
- The fsf was interested in stability and resisted improvements.
- improvements included g77(fortran)
- Pentium optimizations, c++ improvements, and OS variants.

EGCS branch was successful

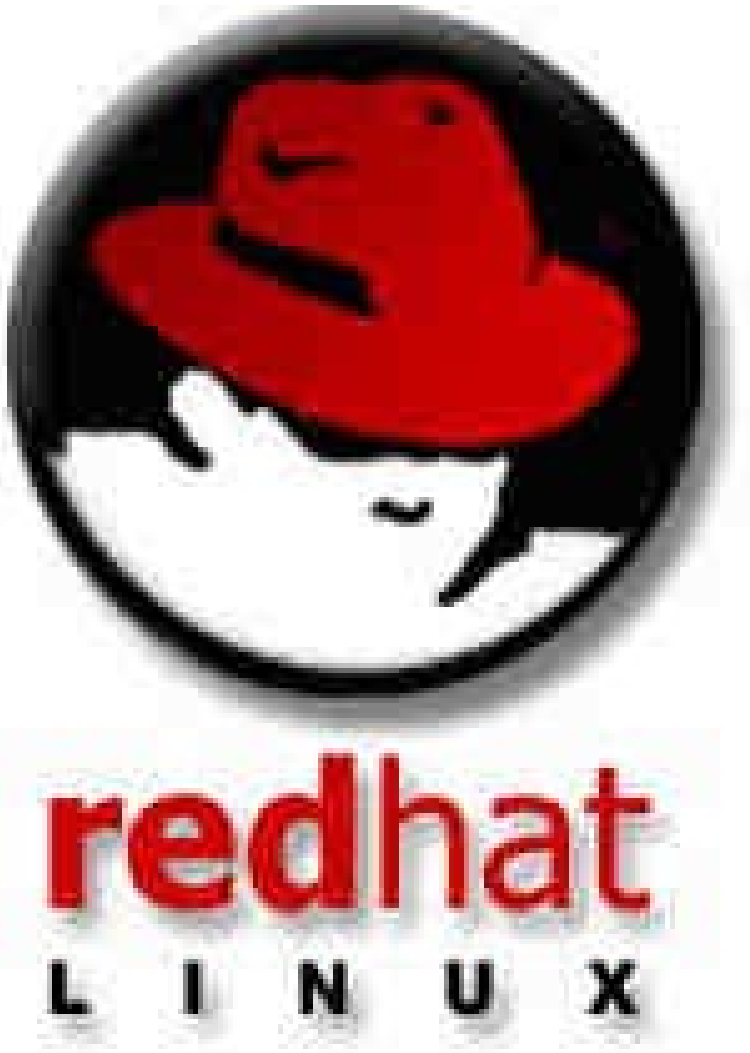
- egcs was so successful that distros began using egcs rather than gcc.
- egcs used the "bazaar" model.
- in 1999 the fsf stopped development on the gcc branch, and blessed egcs as the “official” gcc.
- The developers of egcs became leaders of the newly merged gcc.

The Red Hat gcc 2.96 fiasco

- Just as the egcs fork healed, Red Hat decided to use an unreleased, buggy, developmental branch of gcc called gcc 2.96 for RH 7.0.
- The FSF complained that this code was not for release and was
- incompatible with both the previous and following releases of gcc.
- Developers refused to support RPMs compiled with gcc 2.96.

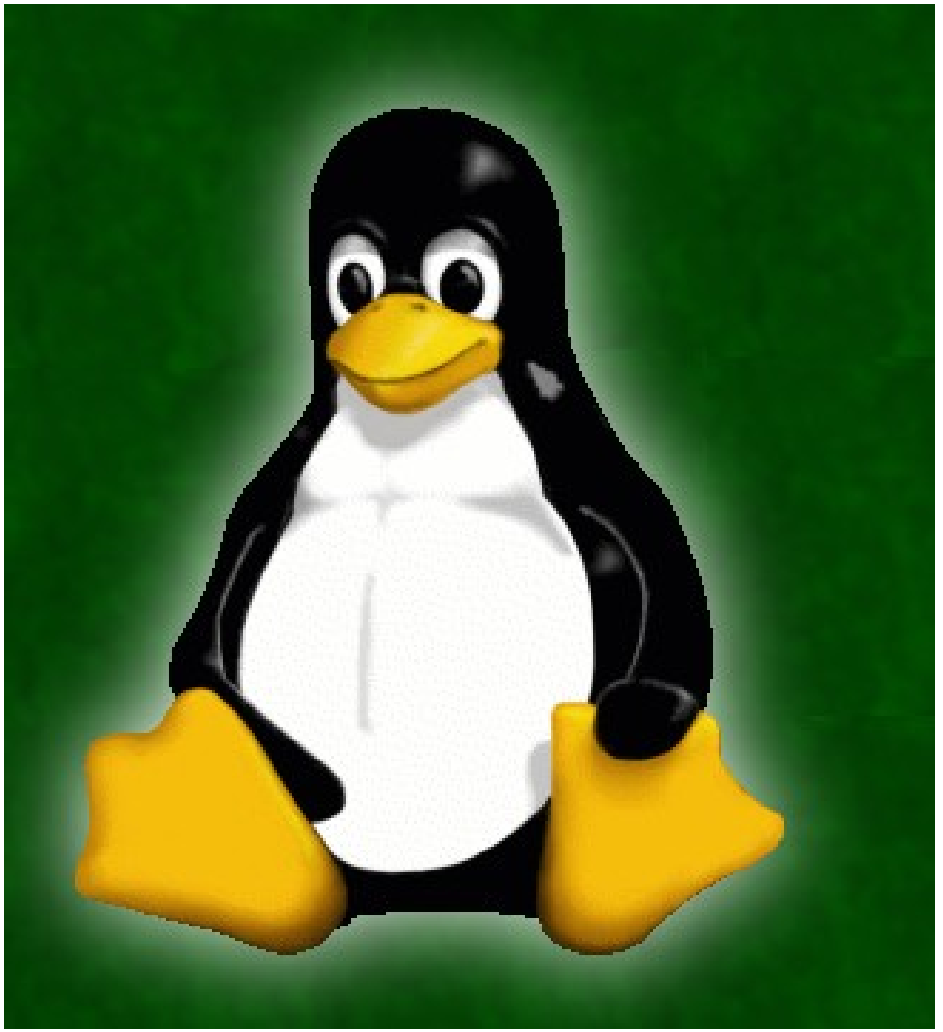
Suspicions

- binary RPMs from RH 7.0 were incompatible with all other distros that did not use gcc 2.96.
- Many suspected that this was intended in order to create vendor lockin.



Tux become Linux mascot.

- “I've always liked penguins, and when I was in Canberra a few years ago we went to the local zoo with Andrew Tridgell (of samba fame). There they had a ferocious penguin that bit me and infected me with a little known disease called penguinitis. Penguinitis makes you stay awake at nights just thinking about penguins and feeling great love towards them. So when Linux needed a mascot, the first thing that came into my mind was this picture of the majestic penguin, and the rest is history.”



Homer Simpson

- Albert Cahalan complained that TuX looked too much like Homer Simpson.

Battle of the titans

- Around 1998 it seemed that MicroSuck's Internet Exploder was about to destroy Netscrape.
- Then, as an emergency survival desperation measure, Netscrape open sourced creating Mozzila!

Free Mozzila grows

- Because netscrape did not own all the code, all code could not be released. Much of the code was not up to free software standards. Thus initial development of the free version was slow.

- Eventually it served as a base for the firefox we know today.

Mozzila Firefox

- began as a fork of the Mozzila project in 2004. Created by Dave Hyatt and Blake Ross.
- Early Firefox documentation described cookies as “Delicious Delicacies”.

Open Source

- In 1998 at the Open Source summit (retroactively named), the term “open source” was invented to avoid all of this embarrassing talk about “freedom” when talking to suits.
- rms was not invited to attend.

Open Source Summit

- Attending were Linus Torvalds, Larry Wall, Brian Behlendorf, Eric Allman, Guido van Rossum, Michael Tiemann, Paul Vixie, Jamie Zawinski of Netscape, and Eric Raymond.
- Primary issue was how to take advantage of the open sourcing of the Mozzila source code.

Open Source Initiative created

- to promote open source software.
- Maintains a definition of open source and a list of licenses thought to be open source.
- They try not to disturb the suits.

FSF and OSI

- the FSF also maintains a definition of Free Software and a list of acceptable Licenses.
- You would think that FSF's list would be stricter.
- But it is not so, for example, the C++ boost library license is listed by the FSF as a Free Software license, but is unlisted by OSI.

- This may be just laziness on OSI's part.

The Halloween Documents

- In October 1998, an anonymous source leaked the infamous Holloween Documents to Eric Raymond (esr).
- The documents describe Microsoft strategy to destroy open source.

- key tactic
- “de-commoditize protocols”

The Halloween Documents

- MS acknowledges “threat of open source”.
- Embrace, extend, extinguish (de-commoditize protocols) is MS's best strategy.

- Patents may help MS.

Linux fork?

- In 2004, Linux almost forked!
- Linus was dropping patches on the floor and key developers were very upset.
- Problem solved by a reorganization.
- “Linus does not scale.”

The BitKeeper Debacle

- A proprietary distributed version control system.
- Linus decided to use BitKeeper for Linux development. (It was offered gratis Larry McVoy Linus' personal friend).
- Rms expressed concern about the dangers.
- Several key developers including Alan Cox refused to use it.

Andrew Tridgell hacks BitKeeper

- Andrew Tridgell (tridge) creates SourcePuller, a BitKeeper interoperability tool that would allow developers not using BK to have access to BK metadata.
- Tridge did not have a BK license and broke no laws or agreements. He used the same methods he used to reverse engineer Samba.

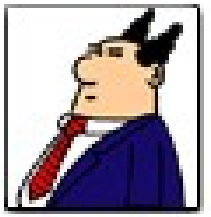
Linus accuses Tridge of “Screwing people over”

- Larry McVoy yanks BitKeeper so that Linux developers must find an alternative.
- Linus accuses Tridge of “screwing people over” even though he did precisely the same thing in creating Samba.
- The fact that key developers like Cox refused to use BK should have been the clue that something was wrong.
- Linus was violating the principle that freedom is more important than convenience.

Arker on Slashdot writes

- In fact, what Tridge has done here is the epitome of ethical behavior. Linus is stung now, understandably disoriented and angry because he's been proven wrong and, being human, his first response is to lash out at Tridge instead of thanking him. Give it a few years though... once his wounded pride settles down I'm sure he will, in fact, thank Tridge for this.
- Locking your data into a proprietary single-vendor format for the sake of temporary convenience was never a good idea. Everyone told Linus this, but he was too smart to listen. Now exactly what he was warned about has happened. And it was inevitable all along - if Tridge hadn't done it someone or something else would have - McVoy was a ticking time bomb. The fact that the guy isn't very stable didn't help, but honestly - McVoy could have been a saint and the thing would have still been a ticking time bomb. If Tridge's actions resulted in it going off a little sooner than otherwise, then he saved Linus and many others trouble in the long run. Replacing BK wasn't going to get any easier...

Save the cost of a hacksaw?


=?


- Was Linus Like the PHB that decides to continue to use Microsoft because of the conversion cost?
- When someone spends money to install a ring in your nose, they expect to get it back.
- In this case, is it smart to save the cost of a hacksaw?
- Freedom is not free, but slavery costs more.

The BitKeeper Debacle(continued)

- Linus' actions showed a lack of vision.
- The debacle had no long term consequences because Linus replaced BK with git. (The cost of a hacksaw.)
- Except perhaps, to the Linus, Tridge friendship.

Indian state of Kerala decides to use Linux rather than Microsoft in Schools

- rms traveled to Kerala to persuade leaders. (2006)
- 3 year plan to replace MS with Linux.
- Kerala is India's most technically advanced state.

SCO vs the World

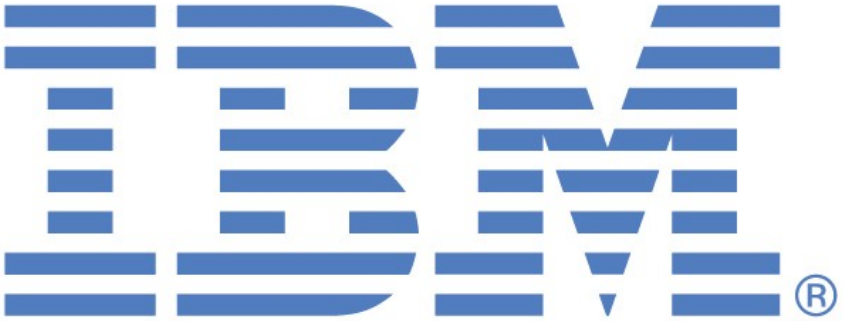
- In 2003 SCO claimed to own UNIX system V and that Linux was a derivative of UNIX.
- SCO's absurd position was debunked by many people including Eric Raymond.
- There is no chance that SCO's executives actually believed its claims unless they are idiots.

SCO vs the World

- Nevertheless, SCO tried to extort money from Linux corporate users in violation of the GPL.
- It also began lawsuits against a number of companies.

SCO vs IBM

- SCO begins a lawsuit against IBM. SCO turns out to be a master of delay, but it could not find any evidence of copyright violation under current law.
- SCO's lawsuit was deep frozen when SCO declared bankruptcy.



Groklaw monitors SCO

- SCO attempts to use FUD (fear uncertainty and doubt)
- Its every move is refuted by Groklaw.

- Groklaw is run by Pamela Jones, a journalist with para-legal training. She runs the blog as a service to free software community.
- She throws cold water on all of SCO's grandiose claims.
- Groklaw was not a factor that SCO's executives took into account when making their extortion plans.

SCO vs Novell

- SCO begins a lawsuit against Novell. SCO claims to own the UNIX system V copyright in a contract dispute. This also turns out to be false.
- This case leads to bankruptcy for SCO.

SCO vs Novell

- SCO is able to create many delays in bankruptcy but in the end SCO will be annihilated and its false claims will die with it.

There is more.

- I have barely scratched the surface of free software history.
- I encourage you to begin your own investigations.
- perhaps you have a different view of some of the events mentioned here.
- Perhaps I have prodded you to back up your views with your own research.

copyright

- This document contains a number of trademarked logos. They are included by virtue of fair use. I obviously, do not hold the copyright to those logos. If you copy this document, you are responsible to insure that ***your use*** of those logos is fair use.
- The rest of this document is
  - Copyright (c) 2008 Paul Elliott.
  - Permission is granted to copy, distribute and/or modify this document
  - under the terms of the GNU Free Documentation License, Version 1.2
  - or any later version published by the Free Software Foundation;
  - with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
  - A copy of the license is included in the section entitled "GNU
  - Free Documentation License".