

Lessons of Free Software History

- Paul Elliott
- pelliott@BlackPatchPanel.com
- Summary about what can be learned from Free Software history.
- <http://www.Free.BlackPatchPanel.com/pme/linux/history-bookmarks.html>
- <http://www.Free.BlackPatchPanel.com/pme/linux/meaning.pdf>

Why does Linux and other Free software succeed?

- GNU/Linux is a fixin' to achieve world domination.
- Corporate theoreticians are in shock.
- How could this happen?
- What is happening?

• Why?



The Proprietary developers have not been idle.

- I don't know how many sucky versions of windoze and other proprietary software has come out.
- Billions of dollars have been spent paying proprietary developers.
- If the problem were a software coding method problem, why has it not been fixed? That is, why does proprietary software still suck?
- The answer is, it is not a software coding method problem, and it can not be fixed.

There is an answer from the history
of free software.

The answer is very simple.
The answer repeats endlessly in many separate
incidents.

Freedom works, control doesn't.



In almost every fork....

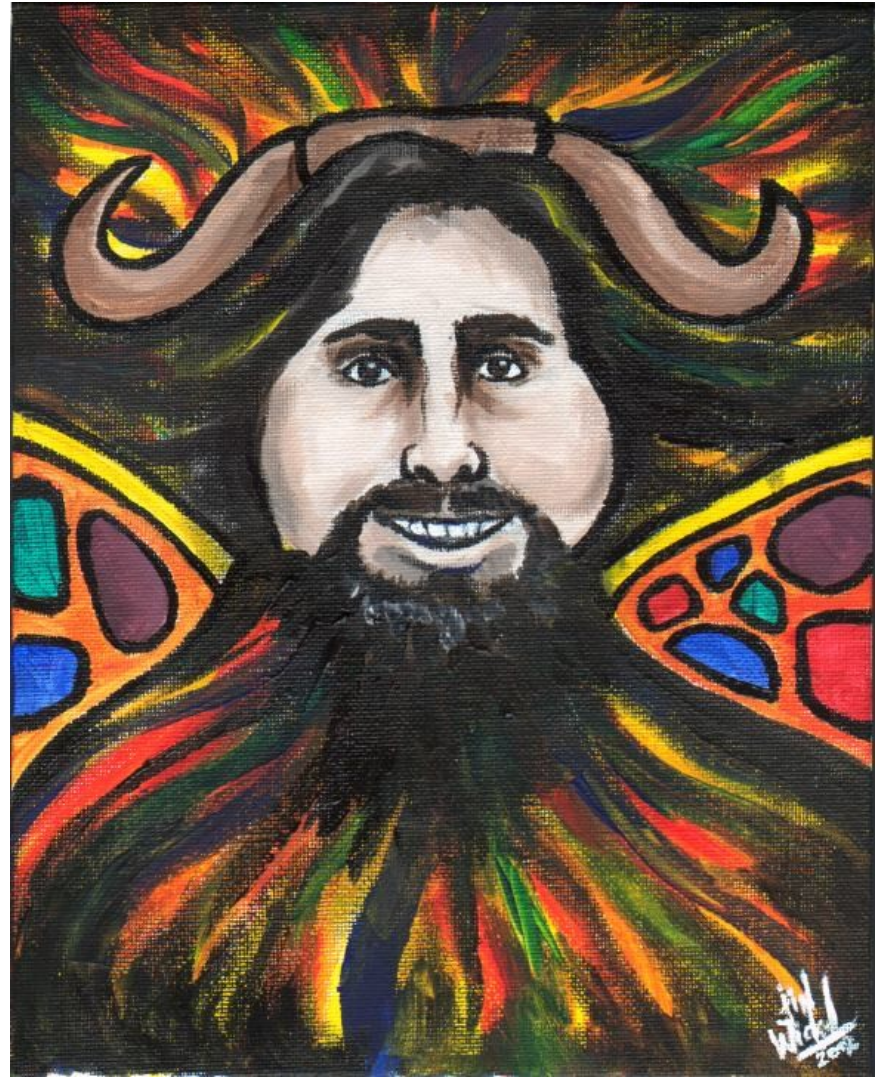
The “free” branch succeeded and the “control”
branch failed.

If someone pays money to install a ring in your nose, is it smart to save on the cost of a hacksaw?



RMS is the hero who discovered the power of Freedom in Software

- Experiences at MIT caused his discovery.
- encoded discovery into GNU manifesto, GPL.
- Applied discovery in creating gcc, GNU utilities.
- Failed to apply discovery in HURD.



• By Jin Wicked

HURD failed.

- Used “Bondage and Discipline” OS development technique from academia called microkernel.
- HURD never advanced to the point it could be “set free”. That is, let many developers work on it “Bazaar” style.



LINUX succeeds.

- Without understanding RMS' principles, Linus applied them “just for fun”.
- Rejected the microkernel and other forms of B&D programming.
- Created something that **ran** quickly.
- Set it free, so that other developers could help (bazaar style).



LINUX succeeds.

- Only a small percentage of today's LINUX code was actually written by Linus.
- Linus uses sound judgment in deciding what contributed code to include.
- If others disagree, they are free to fork.



GNU/LINUX and other Free Software can not be controlled.

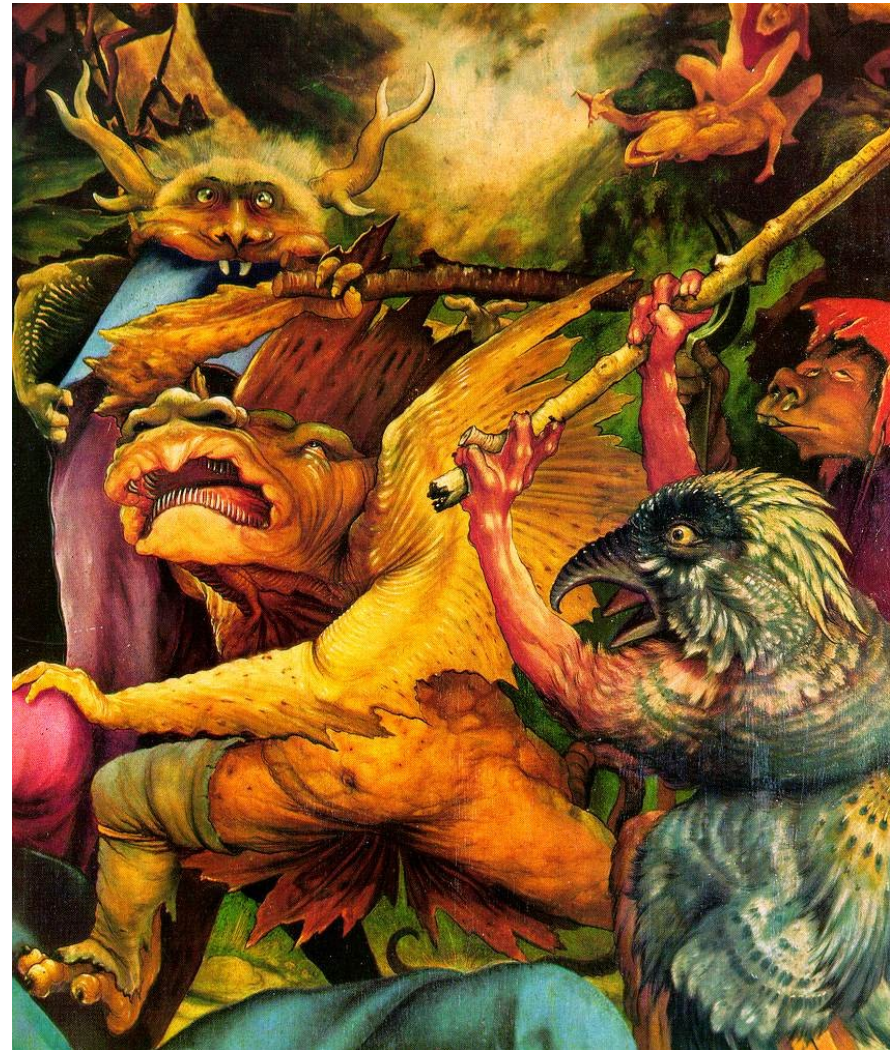
- If Linus (or any other free software author) were to go crazy or quit, other developers would inevitably take over, forking the project if necessary.
- Who are you going to sue, buy, or co-opt? No person or organization owns free software. Can MicroSuck sue to take away the free speech rights of thousands of free software developers scattered all over the globe? Source Code = speech.
- There is no central committee to mess things up or “compromise”.

The GPL protects GNU/LINUX

- Because of the so called “viral clause”, GNU/LINUX can not be attacked by embrace, extend, extinguish.
- GPL provides that any extension must be GPLed. (“viral clause”)
- Therefore if the extension contains any positive features, these features can be embraced by the original branch! So extinguish does not work!

Nightmare scenario

- If by chance, one of the BSDs had become the dominant free OS instead of LINUX, it would have been already attacked by MS using embrace, extend, and extinguish.



Nightmare scenario

- That is, MS would have created a windows/BSD hybrid capable of running both windows and BSD programs. As time went on, the BSD side of this monstrosity would have been “extended” thereby “extinguishing” the original BSD!



The GPL stops hybrid monster creation!



- If proprietary code is mixed with GPLed code, in a distributed product, then by the terms of the GPL, that proprietary code must be released under the GPL.
- This is totally unacceptable to proprietary developers. The proprietary nature of their code is the indispensable basis of all of their tricks and traps and it is the only reason they get paid! To them, it is *sine qua non*.
- This stops hybrid monster creation.
- It is also the true reason that the GPL is hated by proprietary developers and their lackeys!

Free software is a spontaneous order.

- Ancient Chinese philosopher Chuang-tzu may have been first to discover theory of “spontaneous order”. Friedrich Hayek is the first modern advocate of the theory.
- Other examples spontaneous orders.
 - Natural languages. (only the French seek to control the development of language, proof that it is a bad idea.)
 - Free markets – developed in antiquity
 - Organic evolution itself!

Free Software recipe for success.

- Let go of the urge to control.
- Don't use “bondage and discipline” techniques.
- Create something that **runs** quickly.
- Set it Free, accepting the input from many other developers.
- Guide the result. If others disagree, they are free to fork.
- **Let Freedom Ring!**

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".