

History and FUD

by Peter H. Salus

USENIX Historian

peter@netpedant.com

Alexis de Tocqueville observed that it is easier for the world to accept a simple lie than a complex truth.

If you have been following the SCO Group shenanigans and/or the bizarre articles by Rob Enderle or the market analyses by Laura DiDio, you will recognize the validity of de Tocqueville's remark.

But a May press release from the head of the Alexis de Tocqueville Institution, Ken Brown, has agitated me sufficiently to devote time and space to trying to counter the FUD.

Brown released a "study" in which it is "revealed" that Linus Torvalds did not "invent" Linux, which, says Brown, has "questionable" roots.

Of course, Ken Brown doesn't go into detail—this whole thing is a teaser for a "book he is writing on open source software and operating systems." The "study" was promised for May 20, but I received email from a "customer service" person, informing me that "publication has been delayed," but that material was available online.

Eric Raymond, whom I respect, read the "excerpts" that were available. His com-

ments were devastating. Perhaps the kindest was, "This book is a disaster." Andy Tanenbaum, author of several good books as well as MINIX, stated in a published retort that Brown "is not the sharpest knife in the drawer."

(It may be worth noting that the de Tocqueville Institution is, at least in part, funded by Microsoft.)

It's actually quite easy to question Brown's assertions. But most important, one has to realize at the very outset that I don't think Linus has ever claimed to "invent" anything. (Nor am I sure that either Dennis Ritchie or Ken Thompson ever claimed to have "invented" UNIX—their 1983 Turing Award was for "the development and implementation of the UNIX operating system.")

Anyway, the roots of Linux are far from "questionable."

All knowledge builds on previous knowledge.

Sir Isaac Newton (1642–1727) said, "If I have seen further, it is by standing on the shoulders of giants." But that was derived from Robert Burton (1577–1640), who wrote, "Pigmaei gigantum humeris impositi plusquam ipsi gigantes vident" (pygmies placed on the shoulders of giants see more than the giants), deriving this from the Roman general Didacus Stella. Operating systems build on one another. My personal feeling is that it is relatively pointless to try to go back much more than four decades. But even then, at the point where IBM had transitioned from the 701 to the 704 and was moving from the 709 to the 7090, the first transistorized computer, it is clear that the big development was time sharing.

So, the first truly important implementation was Corbato's CTSS at MIT, which led to both the Multics system and to the Dartmouth Time Sharing System.

Dennis and Ken built UNICS (its original name) on their experiences with Multics following Bell Labs' withdrawal from the Multics project in spring 1969. Many important features (like | “pipe”) were suggested or instantiated by others. Pipe was suggested by Doug McIlroy and coded by Brian Kernighan.

At the 1979 USENIX Conference in Toronto, AT&T announced its new licensing fees, including \$7,500 per CPU for academic institutions. This led Andrew Tanenbaum of the Free University in Amsterdam to create MINIX:

I decided to write a new operating system from scratch that would be compatible with UNIX from the user's point of view, but completely different inside. By not using even one line of AT&T code, this system avoids the licensing restrictions, so it can be used for class or individual study. (A.S. Tanenbaum, *Operating Systems, Design and Implementation*, 1987)

Several years later, a student in Helsinki, Finland, wrote an operating system, “just for fun,” which he based on MINIX. Linus Torvalds was going to call it “Freax,” but his sysadmin persuaded him to use “Linux.”

Linux was just a kernel. Thanks to the near-universality of the Internet, it has been augmented and improved by tens of thousands of users.

So here we are—Linux is part of an implementation of a UNIX-like operating system, inspired by MINIX, and using a large number of GNU tools and applications.

Be ashamed, Mr. Brown!