

# Concepts Linger, Code Doesn't

Dr. Warren Toomey  
Faculty of IT, Bond University

`wtoomey@staff.bond.edu.au`

Linux has inherited many ideas and concepts from the Unix system. This presentation will briefly review the history of Unix, outlining why Unix succeeded, and why it failed. I will also discuss the current SCO vs. IBM lawsuit.

# The More Things Change...

---

Linux is a general-purpose, multi-user, interactive operating system which offers a number of features suitable for desktop, server and embedded environments, including:

- A hierarchical file system incorporating demountable volumes,
- Compatible file, device, and inter-process I/O,
- The ability to run threads and processes,
- User environment (GUI, shell) selectable on a per-user basis,
- Thousands of software packages including a dozen languages,
- An extremely high degree of portability.

# ...The More They Stay The Same

---

UNIX is a general-purpose, multi-user, interactive operating system for the larger DEC PDP-11 and the Interdata 8/32 computers. It offers a number of features seldom found even in larger operating systems, including:

- A hierarchical file system incorporating demountable volumes,
- Compatible file, device, and inter-process I/O,
- The ability to initiate asynchronous processes,
- System command language selectable on a per-user basis,
- Over 100 subsystems including a dozen languages,
- High degree of portability. *1979 UNIX paper.*

# Bell Labs: Where It All Began



- Dennis Ritchie and Ken Thompson: the two main Unix designers.
- However, even from the beginning, Unix was a grass-roots team effort.

# Why Unix?

---

- Designed by programmers, for programmers. Interactive. Scriptable. Controllable.
- A malleable environment: apart from the kernel, users can alter everything.
- The toolbox approach: applications can be joined together to make more powerful tools, e.g. Joe Osanna's pipe construct.
- Source code was available: you could fix it.
- The developers took code back from the community: BSD, UNSW etc.
- Simple, elegant, uncluttered design which was discernible.

# 1970s: Research Unix

---

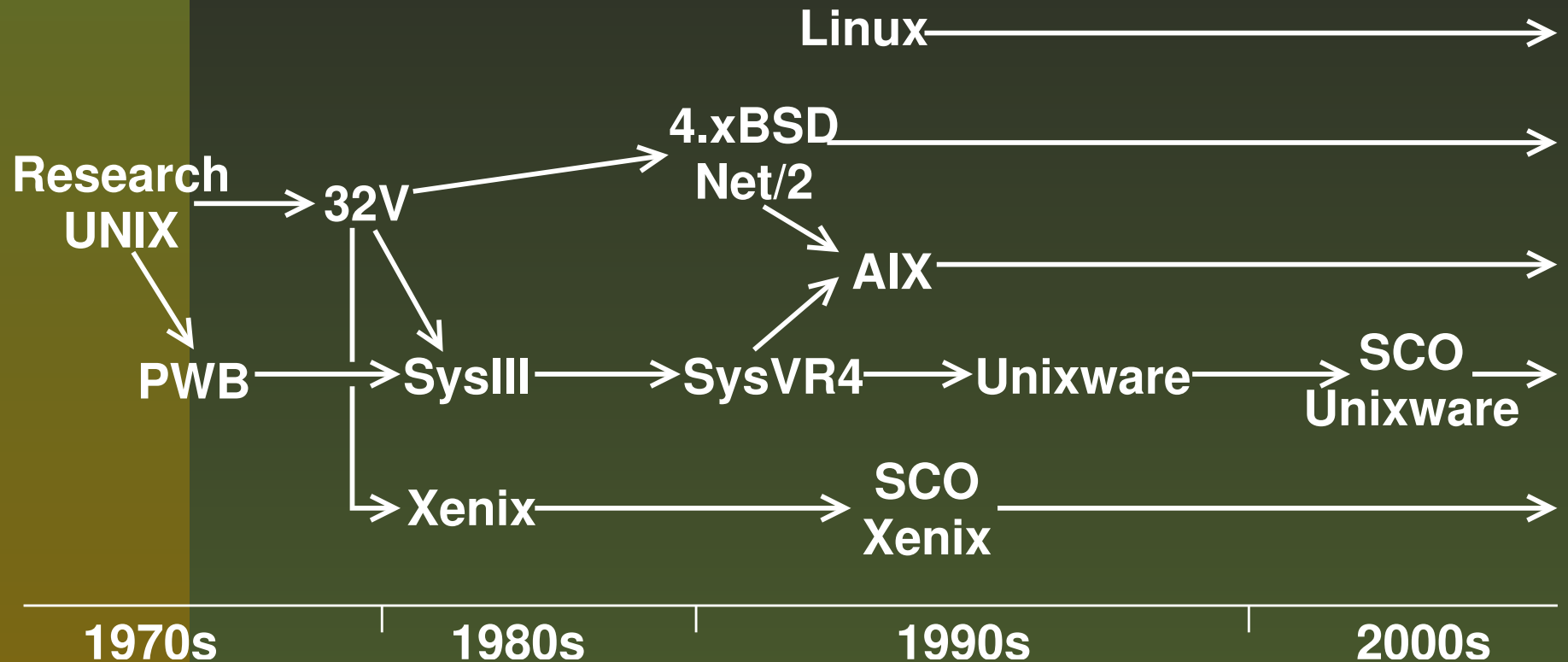
- Bell Labs produced 1st Edition to 7th Edition Unix from 1971 to 1979. Platform: 16-bit PDP-11.
- 1st to 3rd Edition: in-house. Out of the 32 1st Edition system calls, 22 are still in use.
- 4th Edition (1973) onwards was available to universities etc. for a small fee.
- 6th Edition (1976) was the first widely-used system: modified by many institutions, e.g UNSW's AUSAM.
- Wollongong Uni: first independent port of Unix, to Interdata 7/32 in 1977.
- 7th Edition (1979) seen as the “One True UNIX”. Ported to 32-bit VAX as “32V”.

# 1980s: Commercial Unix and BSD

---

- AT&T finally allowed to commercialise Unix in 1984. USL releases System V, derived from 32V.
- University of California, Berkeley, takes 32V, adds paging, networking, improves performance, rewrites most of the kernel & many utilities & libraries: produces 4BSD.
- Companies buy Unix source, sell binary derivatives.
- Sun: SunOS, IBM: AIX, Microsoft: Xenix, HP: HP/UX, Apple: A/UX, Pyramid: OSx etc.
- All slightly incompatible. Each is SysV or BSD based, a few are hybrids.
- Tension: SysV or BSD? Fragmentation => Unix Wars.

# Simplified UNIX Family Tree



- For way too much more information, see [www.levenez.com/unix/history.html](http://www.levenez.com/unix/history.html)



# 1990s: Open Source and Desktops

---

- Unix now seen as a small server & workstation system: not for PCs.
- Someone forgot Moore's Law: hardware power doubles every 18 months.
- Linux appears in 1991: tinker toy system, but gains geek following and thus developers.
- 4BSD loses "all" original Unix code. UCB releases Net/2 with no Unix license required.
- Later, USL sues BSDi and UCB for 32V license violation. Settled out of court in 1994 when Novell buys USL.
- PC hardware catches up to Unix. But Windows arrives with the desktop metaphor!

# End of 1990s: Unix is decrepit

---

- System V considered decrepit: old design, old code.
- The Unix concept is still going strong.
- AIX, Solaris are major rewrites: doing ok.
- The free BSDs: 4-year delay puts them behind Linux mindshare, but still ahead technically.
- Linux's growth & development is amazing: a testament to Open Source development model.
- Unix trademark sold to X/Open: the Unix Standard.
- Novell renames SysV to Unixware, sells SysV to old SCO. Perhaps.
- Caldera merges with SCO to get sales channels. SCO/Caldera renames itself as the SCO Group.

# 2003: SCO vs. IBM

---

- IBM has perpetual SysVR4 source license, used to develop AIX. IBM also contributes code to the Linux kernel.
- March 2003: SCO sues IBM for \$3B in damages, claims IBM violated source license, alleges IBM introduced UNIX code and methods into the Linux kernel.
- SCO continues to distribute Linux until May 2003.
- SCO claims “millions of lines of UNIX code” in Linux, shows ~30 lines of similar code in public.
- SCO demands license fees from Linux users to protect them from future legal action.

# Did Unix Code get Into Linux?

---

- Yes, ~20 lines of Unix code was placed into Linux. Not by IBM, but by SGI for the ia64 platform.
- The code was removed due to its “ugliness”.
- Code first appeared in UNIX in 1973, and is based on a 1968 algorithm by Knuth.
- Code was published in book form in 1997.
- Caldera *released* early UNIXes under a BSD license in 2002, *before* code was added to Linux!
- The Unix Heritage Society(which I run) was critical in tracing the code’s genealogy.
- I had to send SCO copies of the UNIX code from before System V!

# The Other Code Presented by SCO

---

- SCO presented another snippet of code in Linux which it claimed was from System V.
- The code turns out to be Berkeley Packet Filter code, written in 1991 and placed under the BSD license:

Copyright (c) 1990, 1991 The Regents of the University of California. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer...

- It seems that when the BPF code was integrated into System V, the BSD copyright notice was removed.
- Thus, if SCO believes that this is their code, someone violated the BSD license between 1991 and now.

# SCO vs. IBM: Current Situation

---

- SCO dropped all trade secret litigation with IBM.
- Lawsuit is still in the discovery phase: now taking declarations from involved parties.
- SCO definitely dragging their heels over the case, while gushing much FUD and press releases. Accusations of stock “pump & dump”ing.
- Suit hinges on SCO’s derivative theory: even if code did not come from SysV, if at some stage it was used with SysV code, the code is now derived from SysV.
- Original IBM SysV source license appears to deny this theory; so do all the original AT&T lawyers and executives.
- Will SCO win? No.

# So Where Now?

---

- Unix as an implementation is dead.
- Long live the rewrites (AIX, Solaris, the BSDs), and the rewrite from scratch (Linux).
- The core concepts of Unix, however, are still alive and thriving.
- 3 waves of Unix:
  - 1st Wave: Research Unix
  - 2nd Wave: Commercial Unix
  - 3rd Wave: Open Source Unix
- Linux is already (and will continue to be) a challenger to the Windows hegemony.

# Future Problems

---

- All monocultures are dangerous, even a Linux monoculture.
  - Avoid the NIH syndrome.
- We must ensure that portable code stays portable across the Unix systems:
  - `#include <linux/xxx.h>` is problematic.
- Too much diversity is also a problem:
  - Too many distros: repeat of the 80's Unix Wars when fragmentation prevented Unix's success.
- Patents are IMHO the next big threat to the IT industry, and to the Open Source Unixes.



# Metaphor Death

---

- Metaphors don't last forever:
  - 1970s Unix: everything's a file.
  - 1980s Sun: the network is the computer.
  - 1990s: the desktop and the browser.
- What will be the next metaphor?
  - 2000s: the phone is the information portal.
- Unix will survive as a core of functionality wrapped around the next metaphor.
  - e.g Mac OS X is a Unix core with a desktop metaphor above it.
  - Many phones and PDAs have a Linux core.

# Conclusion

---

- Concepts survive, Code doesn't.
- Moore's law and the change of metaphors obsoletes old code.
- The Unix concepts survive because they are powerful, elegant, useful and adaptable.
- Unix as a system will be subsumed by the next metaphor, like it has been with Mac OS X.
- Similarly, Linux will stop being a system and become a technology like Ethernet and the Web.
- The SCO thing is just an annoying "road accident" along the way.
  - Move along, nothing to see here, stop gawping.

# References

---

- [www.groklaw.net](http://www.groklaw.net): Tracks the SCO lawsuits, pro-Linux.
- [www.levenez.com/unix/history.html](http://www.levenez.com/unix/history.html): Unix family tree.
- [www.tuhs.org](http://www.tuhs.org): Collects old Unixes.
- [www.catb.org/~esr/writings/taoup/](http://www.catb.org/~esr/writings/taoup/): Excellent book on Unix philosophy.
- [www.linux.org](http://www.linux.org): Start point for all things Linux.