

A SHORT HISTORY OF THE UNIX LANGUAGE

Much of the progress of computer hardware, software, and networks during the last quarter of the twentieth century was the result of UNIX, a concept that embodies visionary ideas, deliberate generality, and openness. The approach taken and the notations embodied in UNIX influence operating systems today, including the new Apple Macintosh X operating system.

Computer operating systems in the mid nineteen-sixties normally performed isolated, individual, and limited tasks, running single applications, one at a time, and under the control of an individual operator. Unlike today, computers did not communicate with other computers. In fact, computers from the same manufacture usually could not communicate with each other without the use of specialized interpretive software. Companies upgrading computers frequently had to purchase new operating systems along with the new hardware, requiring companies to manually re-enter old data into the new system, an expensive task.

In an attempt to counter these negative aspects of computers the Multiplexed Information and Computer Service (Multics) project began in 1964 as a joint effort between the Massachusetts Institute of Technology (MIT), Bell Labs¹ and General Electric. From its conception, Multics was a robust, general purpose time-sharing system that was modular in design. Since multiple users had access to the system at the same time, security was an important feature of Multics.

Bell Labs withdrew from the project in 1969 because of increasing costs and the growing complexity in the attempt to fulfill the all the needs of the many involved stakeholders. (Bell-Labs, 2002) Multics eventually became a product under the Honeywell Company, but it never

achieved the success of UNIX, which ascended from the collaboration of two researches, Dennis M. Ritchie, and Kenneth Thompson, who had been involved in the Multics project.

In the spring of 1969 at the Bell Labs Computing Science Research Center in New Jersey, Ritchie, and Thompson along with Dough McIlroy, Rudd Canaday, and J. F. Ossanna began working on their own alternative to Multics. Their goal was to develop a convenient, interactive, usable computer system, which would allow multiple users to access the system at the same time, in what was know as time-sharing. Ken Thompson, from the beginning, wanted to write an operating system of his own, and to create an environment in which to do future research work. (Bell-Labs, 2002) At the conclusion of an informal meeting between Ken Thompson and the other Bell Lab researches in early 1969, Rudd Canaday read the meeting notes over a telephone to the Bell Lab's dictation service. That was the official beginning of what would eventually become UNIX.

Once Thompson located an old PDP-7 computer later that summer, he began working on creating a paper file system, which Ritchie referred to as the "chalk file system", as it emerged from discussions on chalkboards in the Computer Science Research Center. (Bell-Labs, 2002) The first part Thompson worked on was the operating system, what would become know as the kernel. The kernel is the lowest level or core of the system. It provides the interface to the hardware's chips and switches, and is unique to each make and model of computer. Thompson then created some simple utilities to copy, edit, delete, and print files, and finally developed the command interpreter, or shell which is the user interface to the kernel. The first shell was a simple single "command line" for entering text commands. Thus, UNIX came into existence. The first customer to use UNIX was the Bell Lab's Patent Department, but other departments

soon followed. Dennis Rickie acquired a newer PDP-11 computer to continue their research, which provided Thompson the opportunity to develop a new programming language.

Ken Thompson wrote the initial version of UNIX for the PDP-11 using assembler language, but ultimately he knew he wanted a higher-level language. He tried FORTRAN, but quickly abandoned it. He then switched to the B language, which was a derivation of BCPL (Basic Combined Programming Language) developed in 1966 at the University of Cambridge by Martin Richards. (Richards, 2009) Out of the B language emerged C, which was written in the summer of 1973 by Thompson. (Bell-Labs, 2002) The C programming language quickly became the de facto standard of programming languages, mostly for the same reasons UNIX was popular. Continuing refinements eventually lead to C+, C++, and finally Microsoft's C# programming language.

One area of research that stands out within UNIX is that of pipes. The concept of pipes gave the programmer the ability to string together different processes. Dough McIlroy explained pipes as methods of switching data streams, like a “switchyard for data systems” or connecting garden hoses together to change the flow of data. McIlroy may have come up with the idea of pipes, but Thompson was the one that make it work. (Bell-Labs, 2002) Security was another facet of UNIX that was added later to facilitate separation of user data. Originally, the use of UNIX was only within trusting environments, so security was not an issue. However, with multiple un-trusted users on a system, the need for better security became clear and became part of the UNIX system.

Developments within UNIX and C lead to a new philosophy in computing. Dough McIlroy articulates this philosophy as: “Write programs that do one thing and does it well.

Write programs that work together. Write programs that handle text streams, because that is a universal interface.” (Bell-Labs, 2002) It was these three axioms that lead to replacing the bulky monolithic applications of the nineteen-fifties and sixties with simple, interconnected software tools under the control of multiple individuals. In addition, open architecture, good documentation, and published standards encouraged the creation and modification of software utilities and the sharing of software utilities between programmers. This shrank development time, which also facilitated the popularity of UNIX’s growth. It is for these reasons that UNIX has become so successful over the last 40 years. The early days of the Internet were also dependent upon UNIX as most the VAX² computers used to support Ethernet and the Internet, had UNIX as their operating system.

While UNIX originated at Bell Labs, it did not stay there. In 1976, Thompson took a sabbatical to teach at the University of California-Berkeley (UCB), and of course, he taught UNIX. Word quickly spread within the academic community about the new operating system. After Thompson returned to Bell Labs, professors continued to work on and expand UNIX under a special academic licensing agreement with Bell Labs. Many of these enhancements collectively became known as the Berkeley Software Distribution (BSD) Version 4.2, which many other universities purchased. Thus, two versions of UNIX were available, the official UNIX of Bell Labs/AT&T, and the unofficial Berkeley unix version.

Many new features and refinements came with time, not only at Bell Labs, but also from university across the nation. UNIX was a product of Bell Labs and AT&T; therefore, they owned the UNIX name. UNIX is not an acronym, but rather a pun on the Multics acronym, and intended as an inside joke between the Bell Lab researchers. UNIX implied a simpler form of a

system, a single instead of a multi. (Ceruzzi, 2003, p. 157) The spelling of UNIX with all upper case or small upper case letters (UNIX) is a registered trademark, presently held in trust by The Open Group. Spelling it with lower-case letters (unix), or less appropriately with a capital U (Unix) indicates it is the generic version of the UNIX, which is not a registered trademark. (UNIX, 2009)

Over the past four decades UNIX and unix mimics have been offered by scores of companies, most with unix sounding names such as HP-UX from HP, AIX from IBM, Solaris from Sun Microsystems and, True64 –UNIX from Compaq. In addition, there is FreeBSD, NetBSD, and of course, Linux, which is a unix type operating system developed in 1991 by Linus Torvalds. Linux is a truly open source and community supported operating system, which is available free to download by anyone.

While the displacement of UNIX is unlikely anytime soon, it has not become the dominate operating system for all computers as predicted by some observers in the nineteen-nineties. While some of this was due to the strong competition offered by Microsoft, it has also been due to the rise and popularity of Linux. (Babcock, 2006) Over the past decade, the popularity of Linux has expanded, mainly do to independent companies, such as Red Hat Linux, providing support to an otherwise community only supported product. With this support assurance, many corporate customers have taken advantage of the lower cost of Linux, especially with the rising cost of the commercial software offered by Microsoft and Solaris.

Therefore, in an indirect way, the sprite and philosophy proposed by Thompson and Richie in the early seventies still thrives in Linux, maybe more so than that in the commercial

sources. In an interview in 2002, Dennis Richie still marvels that “Thirty years after its creation, UNIX still remains a phenomenon.” (Bell-Labs, 2002)

¹ Bell Labs was a research division of the AT&T Company, which divested itself of its local Bell Telephone companies in 1984, and which later became known as Alcatel-Lucent.

² VAX (Virtual Address eXtension) computers were a relatively inexpensive 32-bit computer offered by the Digital Equipment Corporation in 1978.

References

- Aden, David. (2009, March 5). *Open source -- it it free?* Retrieved March 7, 2009, from Government Technology Web Site: <http://www.govtech.com/gt/625338?topic=117691>
- Babcock, Charles. (2006, January 23). *What's left of Unix*. Retrieved February 9, 2009, from Information Week Web Site: <http://www.informationweek.com/news/software/operatingsystems/showArticle.jhtml?articleID=177102427>
- Bell-Labs. (2002). *UNIX History at Bell Labs*. Retrieved February 9, 2009, from Alcatel-Lucent Web Site: <http://www.bell-labs.com/history/unix/>
- Ceruzzi, Paul E. (2003). *A history of modern computing* (2nd ed.). MA: Massachusetts Institute of Technology.
- MIT. (n.d.). *Multics: History of Multics*. Retrieved March 3, 2009, from Massachusetts Institute of Technology Web Site: <http://web.mit.edu/multics-history/>
- Richards. (n.d.). *BCPL*. Retrieved March 5, 2009, from Wikipedia Web Site: <http://en.wikipedia.org/wiki/BCPL>
- UNIX. (n.d.) The Oxford Dictionary of Abbreviations. Oxford University Press, 1998. Oxford Reference Online. Oxford University Press. Franklin University. Retrieved on 23 February 2009 from <http://0-www.oxfordreference.com.olinkserver.franklin.edu:80/views/ENTRY.html?subview=Main&entry=t25.e19018>