UNIX: A View from the Field as We Played the Game

Clement T. Cole

Intel Corporation

Abstract: UNIX is a classic example of a "Christensen Disruptive Technology." It was a cost-effective solution, produced at the right time, built by researchers at AT&T for themselves, and was not originally considered seriously by its competition. The UNIX Operating System had simple goals. It ran on modest hardware, and was freely shared as a result of AT&T legal requirements. As a result, a new computing customer developed, a different one than was being targeted by the large firms of the day. UNIX was targeted at the academically-inclined; it was economically accessible, and since its Intellectual Property (IP) was published in the open literature and implementation was available to the academic community fundamentally without restriction, the IP was thus "free" and able to be examined/discussed/manipulated/abused by the target users. While its creators wrote UNIX for themselves, because they freely shared it with the wider community, that sharing fed on the economics in a virtuous circle as this community developed into a truly global one. I will trace a little of the history of a small newsletter to today's USENIX Association and some of its wider social impact.

Keywords: UNIX ; history of UNIX ; operating systems ; open system ; disruptive innovation

A Brief Personal History

In the mid-70s I was a student at Carnegie Mellon University (CMU), studying Electrical Engineering and Mathematics. I worked for the University as a programmer. Since that time, I have secured a number of degrees there and at other institutions and have since spent the next 40 years developing computing systems. I have been lucky enough to be part of some exciting projects: from core UNIX development, one of the first IP/TCP implementations, to many other technologies. I have published papers and secured patents in computing since that time and currently lead a team of engineers building supercomputers.

A Christensen Disruptive Technology

I contend that UNIX is a classic example of a "Christensen Disruptive Technology" (Christensen, 1997). It was a cost-effective solution, produced at the right time, built by research computer scientists at AT&T for an unserved or under-served market (themselves – who were computer programmers), which at its birth was not seriously considered by its competition. In those days, computers were designed to execute either high end scientific applications typically at the large national/government research laboratories or to perform so called "back-office" work traditionally supporting finance and accounting at large commercial business firms.

The point is that when UNIX was originally written, the purchasers of computing equipment were not primarily made up of the actual programmers of the computing system. Thus, the major computer firms tended to ignore the needs or desires of those the programmers, as they were not seen by the larger firms as the "true" customer since they did not directly pay the bills. However, the target consumers of the UNIX system were also programmers, and of course they did see the value (UNIX was "good enough" for themselves), and a community grew up around it because UNIX was a cost-effective solution for them to do their own work.

UNIX had simple goals, ran on "modest hardware of the day" (Ritchie, 1974) and was freely shared as a result of AT&T legal requirements (Pinheiro, 1987). The fact is UNIX might not have been successful if it had required much larger equipment to execute. A smallish DEC PDP-11/40 class system that a traditional UNIX system ran such as an PDP 11/34 with max memory (256K bytes) would just barely suffice, but that cost on the order of $50K-$150K after disk, tapes, etc. (1977 dollars). If you wanted a PDP 11/70 class system that could address as much as 4M bytes and offer many more services, it was closer to $250K[1] (1977 dollars). To help the reader scale to modern times, the cost of a graduate researcher might have been about $5K-$10K.

Similarly, while these prices may seem large to today's developer using a "personal computer" that costs $500-$1000[2] (in 2017 dollars), another key point is that in those days you did not own the hardware yourself, a user (programmer) like me, used a system owned/operated by someone else; typically owned by your employer or the university you attended. In comparison, the large systems of the mid-1970s that were installed to support scientific or back-office style work usually began at $500K and often reached multiple millions of dollars – so the UNIX systems at $50K-$100K really were modest (much less counting for inflation of the dollar or euro to today's prices).

---

[1] For comparison $50K/$150K/$250K 1977 dollars is $208K/$622K/$1M 2017 dollars.

[2] Similarly, $500/$1000 in 2017 dollars becomes $120/$241 in 1977 dollars (see http://www.dollartimes.com for reference).

The 1956 AT&T Consent Decree

The question is how did this new community occur in the first place? I previously mentioned that UNIX was "*freely shared as a result of AT&T legal requirements*." I am referring to the 1956 AT&T consent decree. This order had extremely important side effects for those of us in the computer business. It is well explained in detailed legal wording by John Pinherio in a paper published in the *Berkeley Technical Law Journal* (Pinherio, 1987), although it is a bit difficult to follow. Instead, consider a quick quote from Wikipedia on the history of AT&T here as this prose is directly to my point and easier to understand:

> In 1949, the Justice Department filed an antitrust suit aimed at forcing the divestiture of Western Electric, which was settled seven years later by AT&T's agreement to confine its products and services to common carrier telecommunications and license its patents to "all interested parties." A key effect of this was to ban AT&T from selling computers despite its key role in electronics research and development. Nonetheless, technological innovation continued.

My non-legal description of the decree is that in return for granting AT&T a legal monopoly for the phone business in the USA, AT&T had to agree to a number of behaviors. One of them was that they were not allowed to be in the computer business[3], but the other was that all AT&T had to agree to continue to work with the academic research community and industry at large as it had done in the past, and must make all of its inventions available to the academic community at no charge or by licensing them for "*fair and reasonable terms*" to their industrial partners – all of those licenses were monitored by the US government.

From a historical standpoint, and as a result of the decree, the electronics industry got a major boost with another invention from AT&T: the transistor. While it was invented in 1947 in Murray Hill at Bell Labs, clearly it was firms such as Fairchild Semiconductor, Texas Instruments (TI), Intel, etc. that would make the money on its invention. We as consumers and as a society clearly have benefited greatly. AT&T was required to license the device (the transistor) to anyone, and they did. In fact, because of the decree, AT&T had created an office in Murray Hill (New Jersey) called the "Patent and Licensing Group" whose sole job was to write those licenses for firms that inquired and asked for them. Ironically this is how UNIX got its start, as the word processing system for those same people. Indeed, a computerized

[3] They were also protected, as other firms such as IBM was not allowed to compete with AT&T in the phone business either.

help made a great sense for them due to the need to rewrite and reformat those sometimes complicated licenses.[4] (Ritchie, 1984).

The key point here is that by the late 1960s, early 1970s when the computer science community was made aware of the UNIX technology, AT&T was required by law to license its technologies to everyone that asks for them and actually had created and instituted the processes and procedures to do just that. By the original legal definition, the 1956 Consent Decree had made UNIX "open," but licensed. This would really be indeed "Open Source" as we think about it today. This is an important point I will come back to later in the paper.

At the time, doing computing research made perfect sense for AT&T, given their core business (telephony), since the "heart" of a telephone system was in fact a stored program, digital computer. In fact, we can look at the Bell System, the phone switching network, as the world's largest and most complex computer system. But by statute, AT&T is not allowed to be in the (formal) computer business. Also, as a side product of building the phone network, just like the transistor which was another core technology created by the researchers at AT&T, software and algorithms were being developed. Of course, the research in software and algorithms would lead to UNIX.

The Murray Hill team had then (and still has) many researchers with degrees in core science and technology such as mathematics, physics, and other academic fields who continue to publish papers about their ideas in the open literature. Those ideas were quite different from other computer systems being discussed at the time in the same places and journals. The same researchers developed the code and ran it internally for their own use; just like they built transistors and used them, their research was also "applied" or in patent terminology: "reduced to practice" from theory.

One of the groups in Murray Hill was a computer science research group, with a number of its members working on topics such as operating systems, compilers and languages, and similar technologies. In the mid-1960s, a number of members of this team had been working jointly with MIT and General Electric on a large-scale computing utility system, the Multics system which was part of project MAC[5]. Two of the members from AT&T were Ken Thompson and Dennis Ritchie. AT&T abruptly stopped working on the project. Ken and Dennis were operating system researchers so it is no surprise when AT&T left project MAC, they wanted to continue working on OS topics at Murray Hill.

---

[4] See also: "Dennis Ritchie obituary", Martin Campbell-Kelly, The Guardian Newspaper, https://www.theguardian.com/technology/2011/oct/13/dennis-ritchie (url accessed 201709041527EDT).

[5] *"CSAIL Mission and History"*, CSAIL, MIT, http://www.csail.mit.edu/about/mission-history (url accessed 2017115301402EDT).

Another research group in Murray Hill had been doing speech work using a PDP-7. While Ken was not in that specific group, he managed to borrow their PDP-7 when they were finished with their original experiment since Ken's team's proposal to purchase their own computer for research had been denied. Later, when the first versions of UNIX started to show promise when running on the PDP-7, Ken's group had to scrounge the $60,000 needed to purchase their first PDP-11 that they would use for the next phase of their work (Ritchie,1984). This work was after all research, prototyping, and exploration of ideas.

By 1974, when Dennis and Ken published the original UNIX paper in CACM (Ritchie, 1974), AT&T researchers had developed a technology their employeer was not allowed to directly sell, and in fact were required to make available to "*all interested parties*". But because they had published about their ideas (and thus the technology itself), the technology and the ideas behind it, drew interest outside of AT&T. Quickly the academic community started to ask about it. By the rules of the 1956 consent decree, AT&T was required by law to make UNIX available to people asking about it. The Murray Hill Technology license office did the same with the fee being a $150 tape copying charge. The fee covered what it cost AT&T to purchase, write and mail the tape back to you. The key point is that if you worked at an academic institution, it was extremely easy to get a license for UNIX for your institution and copies of the UNIX implementation with full sources from Ken and Dennis and many, eventually most, did.

As a result, a new and unexpected computing customer started to develop, which was different from what was being targeted by the large commercial computer firms and valuing different features from these traditional computing systems. This new type of computer customer, of which I was one, did not care as much about what those large systems could do, as their internal core IP of the big systems were not "freely" accessible to us and it did not actually serve us as well UNIX did in many cases. In addition, with UNIX we could do a lot with what we had and the core IP was freely available to us (open), so we could (and would) enhance it as we desired.

The UNIX Community Emerges to Share

As I mentioned, anyone that asked could license and obtain code from AT&T and use it. In fact, the term we used was that AT&T actually "abandoned" the sources at your front door. "*There was no warranty of any kind. You asked for it, there it was. You figure it out*." The first to ask was Prof. Lou Katz of Columbia University. Ken made a copy of the contents of

what we would later call the Fourth Edition available to him[6]. This exercise would be repeated many times over the next few years.

I cannot express the importance of the abandonment comment enough. AT&T senior management really did not have a desire that the actions of the company be seen by the US courts as being in the computer business so they did not want to have anything to do with you as a customer after they delivered the technology. As UNIX users, we were all fellow travelers; researchers doing what we wanted with this technology. Thus, AT&T actions encouraged us (as customers and licensees) to work with each other. Furthermore, the core AT&T research published about UNIX, using the sources and the intellectual property that their employees had developed continued to encourage us to do the same: research and publish about the same technology. This is exactly the same behavior we would later put a name too – we call it "open source."

Similarly, at the time of UNIX started to be available outside of AT&T, Digital Equipment (DEC) was releasing its own operating systems for the PDP-11: originally, DOS-11 and RT-11; which were followed by RSX-11 and RSTS[7]. As the manufacturer of the PDP-11 hardware, DEC was hardly encouraging its user base to use UNIX. They wanted their customers to use their technology and purchase licenses for their software products. They charged monetary fees for the use of the software such as the compilers and other layered products, which was a manner in which these firms made money. By our use of UNIX, we were customers of their hardware, but not of their software products. UNIX was free of real direct monetary cost to us as academics; hence UNIX became competition to DEC software products even if it was not a formal product[8].

Another force was clearly at work. AT&T was legally prohibited from supporting us as customers, and DEC did not want to provide support for the UNIX technology either because the software we had chosen to use (UNIX) was competing with its own; thus, we as users, quickly wanted and needed to talk to each other to share solutions to common problems. I observed that communal activity was forced upon us because it would not have been

---

[6] Private Communication, although this has been discussed and can be been verified elsewhere with a modern search engine.

[7] "Index of /pdf/dec/pdp11," http://bitsavers.trailing-edge.com/pdf/dec/pdp11/ (url access 201709041612EDT).

[8] I should point out there were eventually some fees for the commercial use of UNIX. In comparison to the cost of the hardware at the time, those fees were originally a few thousand dollars, and actually not much different than the fees DEC charged for its OS. In the case of UNIX a licensee also got the sources from AT&T. I'm going to ignore this part of the case for this argument because I personally never found those fees to actually be more than a small hurdle in practice. I was working in the industry from the late 1970s onward, and had installed a number of commercial UNIX sites, and had obtained the first commercial license for a University at CMU in 1978. The fact is that the cost of a PDP-11 or VAX hardware dominated the cost of a UNIX installation in the 1970s and 1980s.

possible to exist economically otherwise. One thing that was becoming increasingly obvious by this time was that the UNIX OS used the hardware differently (better) than the DEC software did, so it would often reveal hardware errors that the DEC standard software would not. A shared approach when working with the hardware vendor was important. If DEC field service was getting the same complaints from the Massachusetts Institute of Technology (MIT), Carnegie Mellon University (CMU), Harvard University, much less AT&T, there was a better chance we were listened to as a joint voice. That is to say, by acting as a community with common goals, not as individual users, we had more power over our key vendors, of which DEC was the most important.

Within the USA when UNIX was starting to spread to many sites, some tier-1 research and academics institutions had access to ARPANET[9] as part of their work with the US Government. Some of the primary Computer Science departments at the major research institutions that had received a copy of UNIX from AT&T were, of course, connected via email on it, but not all. The University of California, Berkeley (UCB) who would become so important in the UNIX story later, was not yet part of the ARPA community. But we all did have physical mail, we all read the same journals and published in the same conferences.

By the time the Sixth Edition was released, Ken added the following note to the UNIX installation (start up) document:

> – send your name(s) and address to:
>
> Prof. Melvin Ferentz
>
> Physics Dept.
>
> Brooklyn College of CUNY
>
> Brooklyn, N.Y. 11210

This mailing list would be called the *UNIX News*. It was published when Mel had enough information. In fact, some copies of some of these newsletters can be found at the USENIX Association web site today[10]. In time, the members of the mailing list started to meet regularly, originally in New York City. People came together and someone might talk about a new trick, or optimization. Sometimes Ken or Dennis even visited and brought a patch or two and they all often brought their own tweaks and additions. It was all very informal and collegial.

By the time of the Sixth Edition was released by AT&T, it was clear that UNIX was a hit in the academic community, although the commercial computing business hardly had noticed it. The number of licenses that were

---

[9] Ancestor to the Internet.

[10] USENIX Notes April 2010 https://www.usenix.org/legacy/publications/login/2010-04/openpdfs/usenixnotes1004.pdf (url accessed 201709041929 EDT).

assigned at this time was probably around 50-100, but that number would grow exponentially shortly. It had become the largest licensed item that the Patent and License Group had ever seen. The rules that the license declared had started to become more straightforward and a bit more formal: this was after all, AT&T's Intellectual Property (IP) and needed to be treated as such, by its licensees. Similarly, the groups of like-minded users started to come together all over the world.

Licensing and Code Access

Almost all software has some sort of license for its use and care associated with it. The UNIX license said the IP was owned by AT&T and we could use it and share it with other licensees – which meant that information could and would be exchanged freely just as it is today in the so called "open source" movement.

The standard practice of the time was that someone at each site would send a photocopy of the "signature page" of their AT&T license to some other site to demonstrate that your site had indeed signed the AT&T license agreement. At Carnegie Mellon University (CMU), we kept a file of the collected signature pages that we received from places like MIT, UCB, Harvard, Purdue University. UCB would have done the same, as would have MIT. The key point being that with such a piece of paper in place, the actual code then flowed between the sites quite easily and without restrictions.

One thing that was interesting and was different at each institution was the handling of source protections (*i.e.* which human users had access to the sources) — the core UNIX IP itself. At CMU, those of us with need for access to the UNIX sources (*i.e.* when we took the undergraduate OS course) had been required to sign a one-page, sub-license with the University stating we understood it was AT&T IP's and we would guard it appropriately. I have not heard of any other school doing something similar. On the other hand, once you had signed that document, you were added to a group that had access to the code and in an extremely free and unregulated manner. The impressions I have had at most other institutions such as MIT or Purdue was that if we had a reason to need access, it was granted in some manner.

When I have discussed this idea with some of my peers that came in the UNIX community, by the mid-1980s, early 1990s when UNIX moved from the DEC based PDP-11 and VAX systems to ones made by firms such as Masscomp, Sun and the like, it seems that for many students it was somewhat hard to obtain direct access to the UNIX sources: you had to be part of the closed "UNIX club" to obtain access to the AT&T IP (*i.e.* AT&T provided source code itself). Certainly, from a source code standpoint, this seems to have been particularly true at the largest educational institutions (at least of ones that I am aware in the USA, or so it has been expressed to me by people who I trust that lived in those times).

However, I counter the "closed club" argument with two points. First, the IP (the ideas) had been completely published by this time, although the source code had not been. The AT&T owned source was never the definition of UNIX, it has always been the ideas of how to build a system were and are the definition of UNIX; which I will discuss more in a minute. Second, for me, having grown up studying UNIX 15-20 years earlier as part of my formal education, using it for work, etc., I never found UNIX to be anything but freely available. However, I can see that if I had been at an institution that had been running UNIX (particularly a binary only implementation) and I had not been given access to the sources (as had been my experience working in the industry), it could have been considered "forbidden fruits" for those people even though the truth was the core UNIX IP (the ideas) really was not.

Teaching UNIX and the Lions' Text

The important thing that had happened by the mid-1970s that is extremely interesting was that the academic community began to study and teach lessons from UNIX. Previously, many schools had developed courses that used a "toy operating system" to teach. With the arrival of UNIX, a real OS could be examined, discussed and understood. In fact, one professor, John Lions of the University of New South Wales, wrote a booklet called *The Lions Commentary on the UNIX 6th Edition* for his students taking courses 6.602B and 6.657G[11]. He included the sources to the UNIX kernel in his booklet (Lions1977). This document itself was widely circulated as a bootleg photocopy and would become cherished by the UNIX community. The author still has his own nth generation xerographic copy from that time.

Clearly, the *Lions*' students, much like me, were immersed in the code and had access to the AT&T IP. The fruits were hardly forbidden. But his book does highlight a new issue that would become troubling for the UNIX community years later — the idea of the AT&T IP being a trade secret. AT&T controlled the publishing of the Lions' text because the book had the sources to UNIX printed in it. However, later books like *Design of the UNIX Operating System* (Bach, 1986), *Design of and Implementation of the 4.3BSD UNIX Operating System* (Leffler, 1989), and later *The Magic Garden Explained : The Internals of UNIX System V Release 4 An Open Systems Design* (Goodheart, 1994) which described UNIX as well, were not controlled. To get a copy of the Lions text a licensee had to obtain it from AT&T's patent license office and only licensees could do so – hence the tendency to photocopy them.

---

[11] The author still considers it the best treatise on how a modern OS works – although the MIT folks have updated it to use the Intel x86 architecture and a modern C compiler since sadly many current programmers no longer find the PDP-11 understandable. See *"Xv6, a simple Unix-like teaching operating system",* Russ Cox, Frans Kaashoek, Robert Morris, https://pdos.csail.mit.edu/6.828/2014/xv6.html (url accessed 201709031127EDT).

A key point is that by the time many of us left university in the late 1970s, we had been "mentally contaminated" by the AT&T IP – the core UNIX ideas from Ken, Dennis and the team in Murray Hill. Their ideas were now how we thought; they were the foundation of how we built systems, and led to how we would teach future programmers.

Absence of support and the USENIX Association

As I have said, when the UNIX community began to take shape, because of the consent decree and the fear of another anti-trust case against them, the AT&T management had officially and specifically abandoned us as UNIX licensees from a legal standpoint. The truth is that Ken and Dennis and the rest of the team in Murray Hill, NJ were wonderful people. They did want to help us as users of the technology and fellow members of the greater UNIX community, so when they could, they did. They were our friends and colleagues and we helped them when we could, too. But they were limited in what that could do. As it happened, Ken had a set of patches to the Sixth Edition he wanted to get out to the community to fix a number of issues that had arisen since the original distribution. There was no formal way to do it. Officially, such an update distribution was not allowed ("*no warranty implied*" said the license we had signed) and there really was not yet a formal mechanism even it had been allowed by the AT&T license.

At the time, Ken was driving his family to California for a sabbatical at UC Berkeley. He made a stop at the University of Illinois to see a soon-to graduate student, the late Greg Chesson who was about to start to work full time at Bell Labs. Somehow Ken's patches managed to get copied to a number of sites on the ARPANET. (I never knew how and Greg is not here to tell us). I know we got them eventually at CMU, and I believe MIT got them around that same time. There was a comment about the patches existing in the *UNIX News* and by then they were being passed by tape when different users got together. Simply put, the UNIX community was sharing what it had with each other. We were all in it together.

The meetings, newsletters, sharing of tapes were really the beginning of the USENIX Association [12] (USENIX, 2017). *UNIX News* started what became a series of conferences, in those days twice a year, winter and summer, when the UNIX users could come together[13]. The benefits of the conversations beside basic fellowship of course was we could share knowledge in person and specifically share implementations, additions and corrections. We would organize talks on different topics and to many of us

---

[12] "About Usenix", USENIX Association, https://www.usenix.org/about (url accessed 201709041858 EDT).

[13] The newsletter itself was renamed *;Login* which was a credit to the original UNIX herald and continues to be published under than name.

the highlight of the meetings was the creation and redistribution of a 9-track magnetic tape which held the collected offering from different sites. Archives of these distributions still can be found today in places such as the UNIX Source archives maintained by the UNIX Heritage Society[14].

Soon the conferences, called the semi-annual USENIX conferences, were not only in NYC. They were also in Cambridge (Massachusetts, USA) at Harvard University, then in California, Toronto and Boulder. But the idea was not just US centric, the conferences started to spread to Europe, too. The UK was first, with Cambridge (UK), but quite quickly, the continent got busy. A European UNIX Group formed. The USENIX Association started to publish proceedings of its conferences, and instead of having just two conferences a year, they started to sponsor special conferences on topics within the community. Given the academic seeds, the USENIX Association itself became a well-respected publishing arm and its conferences became preeminent, *i.e.* publishing papers at USENIX conferences was important for getting tenure if you were a Computer Science person at one on the top Universities. To this day, USENIX sponsored conferences, such as the FAST and the Security conferences, are the top conferences in those specific areas to publish for storage or security topics.

In fact, one of the most important gifts to the computer community that I can think of came from the USENIX Association, what we call the "open access" movement. Starting in 2008, when I was on the board and serving as the president of the USENIX Association, the open access movement was born to make sure sharing of information was easy[15]. The concept of open access simply put is that ideas need to be open to fuel creativity and experimentation and, thus, more and better ideas. As a result of this change, all of USENIX's publications have been available to anyone to read without any fees, which stems directly from the original UNIX philosophy, ideas and actions.

Open, Free, Libre and Gratis

Parts of the original UNIX source code and its descriptions of how it worked were published in journals, papers and books. At conferences, such as USENIX's, the sources themselves used to build the entire system were being "freely shared." Note that the description of the code was open. While you never did need to have an AT&T license to come to a USENIX conference, those that came, were covered by their employer or academic institution, at least initially.

---

[14] "The Unix Heritage Society", https://www.tuhs.org (url accessed 201709041904 EDT).

[15] USENIX Notes April 2010 https://www.usenix.org/legacy/publications/login/2010-04/openpdfs/usenixnotes1004.pdf (url accessed 201709041929 EDT). USENIX Update April 2012 https://www.usenix.org/blog/usenix-supports-open-access (url accessed 201709041930 EDT).

Thus, we had USENIX conferences or equivalents in Europe and we used them to trade code back and forth. The manner in which we traded code was often physically carrying or mailing a magnetic tape, but code changed hands from one site to another without hindrance. Indeed, we were careful to send the tapes home to a site under the care of a person. But, it was very much what we now call an "open source culture" as different groups modified the code. I never saw us fail to send a tape home to any licensed site.

As I said, this sharing was happening at a prodigious rate. Each site added features that were important to them or to fix issues that became acute in their environment. For instance, the original ARPANET code was done at University of Illinois (UofI); CMU would create disk recovery and Emacs, MIT a different Emacs and gave us compilers for many microprocessors; University of Delaware (UDEL) wrote a mailer, lots of people wrote editors, *etc*.

The most famous collection of modifications and additions to the UNIX "trade secrets" became the Berkeley Software Distribution (BSD) from the Electrical Engineering and Computer Science (EECS) department at UCB, distributed to their licensees (which of course all had an AT&T license). Since the 1960s and before the Computer Science part of the department has been created, the EE portion had operated its Industrial Liaison Office (ILO). One job of the ILO was to distribute "gratis" (free of any specific direct monetary charges) the sources to the code developed at UCB basically to anyone that asked, which in practice was the industrial partners around the world. Programs such a SPICE, SPLICE, MOTIS had seeded the electronics industry and mechanism to collect licenses for the codes and release tapes of them was already in place when the first BSD[16] for UNIX was made available as an update to the Sixth Edition. This was followed by 2BSD for the Seventh Edition/PDP-11 and then 3BSD and 4BSD for the VAX. Within a few years, the UC Berkeley team would get a DARPA contract and create the Computer System Research Group (CSRG) and start to push out BSD releases with prefixes of 4.1A, 4.1B, 4.1C, 4.2, 4.3, 4.4, 4.4Reno, 4.4Tahoe, and NET.

Here we see an interesting trend. By the time of the creation of UCB's CSRG team, we actually have an entire branch of the computing industry for UNIX, which has been named the "open systems" products to computer community at large (note the title on the SVR4 book I mentioned previously). We have small companies such as Masscomp and Sun or large ones such as Group Bull, Siemens, ICL, DEC, HP and IBM all producing products based on "open systems" technology. We even start to have a war within the community itself as to what is the definition of "UNIX".

---

[16] The first Berkeley UNIX Software Distribution was actually called BSD at the time of its release, but to distinguish it from later releases, modern people often refer to this as 1BSD.

Christensen Disruption Revisited

At the same time, many companies, including the original UNIX hardware manufacturer DEC, continued to produce competing computing systems that were not based on UNIX. Yet, UNIX which originally was not a product, is now an actual branch of the industry (the "Open Systems" branch) and it is thriving and growing at a tremendous rate. How did this happen?

As Christensen points out, disruptive technologies start off as a worse technology, but a different group of people values the technology. This is exactly how UNIX was created. It was targeted for a small inexpensive computer (PDP-7 originally, later PDP-11), written to provide a platform by programmers, for programmers, and as a place to experiment with ideas. It was not created originally as a large enterprise-wide computing service, not integrated together, nor delivered by a manufacturer. Even its humorous and irreverent name was a contrast to developers' experience with Multics (Organick, 1972).

As Christensen observes, in many ways that software would have been compared, looking at the Fourth, Fifth, Sixth or even the Seventh Editions of UNIX against its commercial peers of the mid-1970s (RT-11, RSX-11, RSTS, VAX/VMS), UNIX was considered to be "weaker or "not as good", if not a "toy" by many measures. Moreover, I mentioned at the beginning of this paper that programmers did not buy computers. But something extremely important had occurred: students like me had begun to graduate and were now in the workplace. We had been mentally contaminated with those open UNIX ideas, and while we may not have had the budget, we were asked by our bosses what to buy and we all wanted UNIX. What UNIX did for us was what we valued, not what the old systems had valued. And as Christensen predicts, the technology had gotten better — all while disrupting and displacing the mainstream because the growth curve of the UNIX community was much more rapid than the older curve.

Christensen predicts that worse technology will improve and over take the older technology, but what he did not predict and is interesting is that the case of UNIX, that because the technology was "open" / "free", its targeted users were the ones that could and did make it better! That is, this dynamic new market, which was made up of a whole new community of people who had previously been ignored in the old market, were the power that drove making the technology valuable. Many people in the community around the world worked to add to the core UNIX technology, because the core intellectual property had been freed by a side effect of that 1956 decree. Moreover, because it was in the open literature, published, taught, improved, the technology got strong because more people were able to contribute.

What does "Free" Mean

In perspective with the notion of "free software", was UNIX really free? The code was published. We read about it. We had access to it. Indeed, the UNIX source was licensed, and licensing did not change the freeness – as that is a separate quality. The ideas were published, the ideas were discussed. The source was shared within a community that had them for very little money. This is an important point: free does not mean it was without complete monetary cost (gratuity), but rather intellectually free (*libre*), which in this realm proved far more powerful. Some amount of money was spent for licensing. Some amount was spent on hardware. It is true that one can claim some people did not have access to the sources at all times (*e.g*. the 1980s and 1990s during the "UNIX Wars"), but the ideas were never locked away, they were never "closed."

So, the question is, at what point does an idea transition from a concept to thing? To me, while being able to use UNIX (the instance of it) made it real, and is what allowed us to come together, it was the ideas that really mattered and that was the part that was truly free.

In fact, the US courts came to the same conclusion, too. As I mentioned before, AT&T had considered UNIX a "trade secret." In the early 1990s, AT&T sued the UCB and a small firm made up of the former CSRG folks, called BSDi, on these grounds that they had used the trade secrets inappropriately and AT&T actually lost the case[17]. The finding of the court was simple: with the 1956 Consent Decree, AT&T was required to license its technology. When they licensed UNIX to UCB and similar academic institutions, then UCB and those institutions used to it teach people like me, we were by definition, "mentally contaminated" with AT&T's IP and they could not, by definition claim the IP was a trade secret any more. The ideas (the core IP) behind UNIX could not be locked up. The US courts declared the IP open and free because it had been published and described! (Pinheiro, 1987).

Economics *vs.* Technical Purity

The penultimate idea I want to explore about the driver for UNIX success is less obvious, but I think equally important. Engineering schools usually teach the best or optimal way to design different things. We take this thinking and often critique technical designs for some level of purity. Yet engineering is often made up of trade-offs, so one person's optimization might not be considered such by someone else.

When they started, Ken and Dennis were not trying to have an academically perfect system as they developed UNIX. They were not trying to build something glorious, or even a product for that manner. Technical

---

[17] "USL vs. BSDi documents", Dennis M. Ritchie, https://www.bell-labs.com/usr/dmr/www/bsdi/bsdisuit.html (url accessed 201709012004EDT).

purity was a non-goal; getting something they needed to get done was. Similarly, when users like me picked UNIX as a system, we picked it not because it was polished, came packaged, with a warranty or even included many programs traditionally associated with a computer OS (such as a real Fortran compiler); we picked it because it solved our problems inexpensively and effectively. UNIX was cost effective for us and we did not care about the deficiencies because its strengths were in features we cared about, which more than made up for its weakness. Because it was open, we could enhance it as we needed. And, enhance it we did. The key is that economics on UNIX was in our favor to enhance it. If you will, the cost to enter (purchasing the hardware was low), the software was basically free and student labor was inexpensive, so why not enhance it?

An International Sensation

One of the most wonderful parts of the UNIX story is that it crossed cultures and international boundaries. UNIX had been developed in the USA for programmers at AT&T. But the interesting thing is that the international language of programmers was problem solving and really is independent of most or many spoken languages. Even though UNIX lacked support to spoken languages other than English itself, most programmers spoke English already so that fact that it solved a set of problems that was globally common made a fine solution. It was not a single culture that made UNIX succeed, other than the one the technology created itself. As I hope I have demonstrated, UNIX was open and free so anyone, anywhere could be part of the community. As I pointed out, the best description of how the UNIX system worked was written in Australia, but it did not stop there. Professor Andrew Tanenbaum literally wrote volumes from his offices in the Netherlands[18]. It really was a worldwide effort with many hands being part.

In fact, in time, because UNIX was such an international sensation it made sense that it would become the leading system to support an internationalization effort, that was followed by many other systems later. For instance, the techniques used for the C programming language for systems like Microsoft's Windows were taken from the internationalization work that UNIX offered.

It is said imitation is the greatest form of flattery and UNIX started to be imitated. These "UNIX clones" sprang up all over the world. The first clone was done by an ex-AT&T person, Bill Plauger, who wrote a Sixth Edith clone called Idris[19]. I remember being extremely excited by a paper

---

[18] See: https://www.amazon.com/Andrew-S.-Tanenbaum/e/B000AQ1UBW (url accessed 201708010912EDT).

[19] Idris was marketed by his then firm, Whitesmiths, Ltd. Traces of the code seem to be gone on the Internet, but you will reference to it.

given by Michel Gien on the Sol Operating System at USENIX Summer 1983, as researchers in France were working on a clone of UNIX written in Pascal (Gien, 1983). The Pascal implementation would lead to the world-famous Chorus System done in C++. Tanenbaum and his students would write Minux a complete Seventh edition clone[20] and like Lions, published the full sources in his text book (Tanenbaum, 1987) as he wanted something with which he could teach, but at the time we were worried that he might be restricted. AT&T even investigated another Seventh Edition clone, called Coherent[21].

To me the height of this is today's Linux System, which clearly holds the UNIX banner high. The original kernel was written by a young Finn, we all know. Now the project has thousands of contributors around the world, with millions of lines of codes between the kernel and layered applications. What an amazing statement to the UNIX legacy, Ken, Dennis and the people that started it all.

The Virtuous Circle

Finally, you should notice a circular argument going on here, but that actually is the basis of my final point. UNIX was targeted at the academically-inclined, was economically accessible, and since the ideas which formed the basis of UNIX had been published in open literature and the implementation of those ideas had been made available to the academic community fundamentally without restriction, the "UNIX IP" was thus "free" and able to be openly examined, discussed, manipulated, even abused by the target users. Thus, my secondary thesis is that while its creators wrote UNIX for themselves, they freely shared it with the community and the sharing fed on the economics in a virtuous circle[22] that caused the community to grow. The more it grew, the more the demand for it became. As there is more demand for UNIX, there is more investment in its technology, and more people want to participate in developing the technology, which causes more demand — exactly what Krugman describes as a virtuous circle![23]

---

[20] *"MINIX 3"*, Minux Group, Vrije Universiteit Amsterdam, http://www.minix3.org (url accessed 201706191751EDT).

[21] *"Mark Williams Company Sources"*, Stephen A. Ness, Ness and MWC, http://www.nesssoftware.com/home/mwc/source.php (url accessed 201706171421EDT). The Coherent story in interesting in itself. Dennis was one of the people that was asked to investigate it. AT&T eventually concluded it was unique enough as to not prosecute and the system stayed on the market, although was not financially successful.

[22] See *The Fall and Rise of Development Economics*, Paul Krugman, MIT, http://web.mit.edu/krugman/www/dishpan.html (url accessed 201706191751EDT).

[23] Thank you: This paper would not have been possible without a number of people encouraging me to write this history down for the future generations to read. In particular, my daughter Leah Cole, my sister Caroline Cole, my wife Margaret Marshall and the editorial team of *Cahiers*

Bibliography

Bach M.J. (1986). *Design of the UNIX Operating System*, London, Prentice-Hall, ISBN 0-13-201799-7

Christensen, C.M. (1997). *The Innovators Dilemma,* Boston, Harvard Business School Press, ISBN: 0-87584-585-1.

Gien M. (1983). "The SOL Operating System", USENIX Association, 1983, *Proceedings of the Summer '83 USENIX Conference, Toronto, Canada, July, 1983*, Pages 75-78.

Goodheart B. & Cox J. (1994). *The Magic Garden Explained : The Internals of UNIX System V Release 4 An Open Systems Design*, London, Prentice-Hall, ISBN 0-13-098138-9

Leffler S.J., McKusick M.K., Karels M.J. & Quarterman J.S. (1989). *Design of and Implementation of the 4.3BSD UNIX Operating System*, Boston MA, Addison-Wesley Publishing Co, ISBN 0-201-06196-1

Organick E. (1972). *The Multics System: An Examination of its Structure*, Cambridge (USA), MIT Pres, ISBN 0-262-15012-3

Pinheiro J. (1987). "AT&T Divestiture & the Telecommunications Market", *Berkeley Technical Law Journal*, 303, September 1987, Volume 2, Issue 2, Article 5.

Ritchie D.M. & Thompson K. (1974). "The UNIX Time-Sharing System", *Communications of the ACM*, July 1974, Volume 17, Number 7, pp. 365-375.

Ritchie D.M. (1984). "*The Evolution of the UNIX Time-Sharing System"*, *AT&T Bell Laboratories Technical Journal*, October 1984, Volume 63, Number 8, Part 2, pp. 1577-1594.

Tanenbaum A.S. (1987). *"Operating Systems : Design and Implementation",* London, Prentice-Hall, ISBN 0-13-637406-9.

---

d'histoire du Cnam and their reviewers (especially Camille Paloque-Berges and a new friend from this endeavor, Loïc Petitgirard, who provided feedback and help in editing).