



There can be only one! A summary of the Unix standardization movement

by [Vernard C. Martin](#)

Perhaps the world's most popular operating system today is also the hardest one to define. Unix is undoubtedly one of the most widely used operating systems. Several versions of Unix compete in the commercial marketplace with products such as DOS, MS windows 3.1, Windows NT, and VMS.

Even with its widespread use and 25 year history, in some cases telling what *isn't* Unix is difficult. The push to standardize Unix has been growing for quite some time now with several groups mixing it up. Whether the issue is about to come to a head is unclear. The only outcome has been verification of an old gunnery sergeant's saying ``Ain't nothing easy when you're doing it for real!''

Why is Unix so hard to standardize?

An important difference between Unix and other OSes is the plethora of tools and applications that are traditionally bundled with it. Though these programs are not part of the actual OS kernel and the system can run perfectly well without them, the exclusion of these tools would be extremely detrimental towards the overall usefulness of the operating system. Such tools include lex, yacc, awk, grep, and sed. To properly standardize Unix means standardizing these tools as well.

By far, the most important reason it is so difficult to standardize Unix is the strange and unique development history surrounding it. To really track its standardization process, you have to track Unix's history as well which is no easy feat. Almost every

book that covers Unix in any substantial amount will have an obligatory paragraph or two about its colorful past. By and large, most of them are either too vague or inconsistent. A really good source of information on Unix history can be found in the book ``A Quarter Century of Unix'' by Peter H. Salus, which proved vital in the author's writing of this article. I will not attempt to rattle off the entire past of Unix. Instead I will give you those bits of history that emphasize the diversity of the Unix development and thus contribute to the problem of standardizing Unix now.

Some basic Unix history

The Unix family of operating systems have been in existence since around 1969. The exact date varies depending on what you actually consider Unix to be. However, most folks will probably agree that the system that Ken Thompson and Dennis Ritchie of Bell Labs wrote for the PDP-7 was the start of Unix. Those guys didn't have a fully documented idea of what Unix was going to be or even a name. They got the name later as a pun on MULTICS. ``Emasculated Multics is Unics.'' The change from CS to X happened later and nobody seems to remember quite why. (It's interesting to note that one of the main reasons for writing the first Unix kernel was to get better interactive performance from the computer to play a game called Space Travel.) In any event, having a standard for Unix was obviously the last thing on their minds.

Work continued on Unix and more folks at Bell Labs joined in including such greats as Brian Kernighan. In 1971, it started being used as part of a text processing system within Bell Labs. This resulted in three things: the development of the programming language C (by Kernighan and Ritchie), a complete rewrite of the OS and supporting programs in C, and the formation of the AT&T Unix Systems Group (later known as Unix System Labs). All of this was completed by 1973 which is impressively fast even by today's software schedules.

Unix was mainly being used internally and no official standard was needed. At the time, AT&T was under a consent decree and could not market computer products. Unix was released to universities for educational purposes. The majority of all development work on Unix that was done by AT&T took place at Bell Labs. The first version widely available outside Bell Labs was Version 6 in 1976. (The version number for early Unix systems corresponds to the edition number of the Unix Programmer's Manual that was current when the distribution was made; the code and the manuals were revised independently.) Since Unix was only available from one source, there was a de facto standard: whatever you got from AT&T was Unix.

Much modification and revision of Unix was done by sundry people at a fairly large number of locations over the next few years. This parallel development by groups with different goals led to many Unix variants. With no standard to go by and the OS evolving quickly, anyone could have easily predicted that creating a standard later would be a nightmare. Most of the development efforts took place at universities. One of the more significant of the development groups was at Berkeley and resulted in the Berkeley Software Distribution of Unix, a main branch of the Unix OS family.

In 1978, AT&T Version 7 was distributed. During 1979 to 1982, Bell Labs combined several AT&T variants into a single system, known commercially as Unix System III. More features were later added and an internal release developed into System V. Meanwhile, the good folks at Berkeley busily worked on their own variant which was quickly gaining popularity. Many features of BSD Unix migrated over to be found in System V Unix. Several versions of System V were released, the current version of System V is release 4 (commonly referred to as "Sys V R4").

[[figure](#)]

True standardization attempts arise

The first attempt at standardization of the Unix OS began with a proposal to the Board of /usr/group, an organization dedicated to the promotion of the Unix Operating System. The proposal suggested the formation of a Standards Committee and by 1984 the committee's efforts had been adopted by the membership. Though the group had no official status with any standards granting organization, its results were highly influential when the ANSI committee for standardization of the C programming language started their deliberations. However because of the divergent developments of System V, BSD, and others, the standard was inadequate. In early 1985, the /usr/group committee was merged with the newly formed IEEE POSIX Working Group (POSIX stands for Portable Operating Systems for Computing Environments) and the /usr/group standard was adopted as a first draft.

Also in 1985, AT&T published its System V Interface Definition (SVID). A new edition was published five years later. In the interim, X/Open (which has nothing to do with X Windowing System) published the X/Open Portability Guide (XPG). The XPG has had several revisions with the latest being XPG4.

By 1989, a multitude of IEEE subcommittees focusing on the POSIX standard had been formed (over 20) but only two standards documents were produced, POSIX.1 and POSIX.2 (recently renumbered to POSIX 1003.1 and POSIX 1003.2). They focus primarily on the kernel library calls and the shell definition. Even so, POSIX was not really a Unix standard. It is an interface description, not an implementation. Many quote the POSIX standard as ``a generalization of Unix."

The here and now

Recently there has been an even more urgent rush to unify the different Unix camps into The One True Unix. The Unix trademark was owned by AT&T through its subsidiary group Unix Systems labs. In early 1993, Novell bought USL from AT&T and along with it, the Unix trademark. Under the new ownership, USL modified System V R4 so that it could easily interoperate with Novell's existing product, NetWare. This new version of Unix was released as UnixWare. This was an obvious attempt to gain an even bigger share of the networking and OS market before the release of Windows 95 or Windows NT.

UnixWare never gained in popularity and did not catapult Novell to the top of the stack as the company had hoped. However, it did provide a really big and much needed wake up call to the commercial Unix community. Unless a serious unification process occurred, no commercial Unix would survive the oncoming onslaught of Microsoft's 1-2 punch of Windows 95 and Windows NT. The different Unix camps lessened their infighting and concentrated their efforts to bringing about a unified standard.

In October 1993, Novell announced its plans to give X/Open the trademark rights to Unix. Novell officially gave rights to the trademark to X/Open in early 1994. It took all the intervening time for the legal matters to get ironed out for this to occur. This step put the rights to determine what could be called Unix into one ruling body. By doing this, they almost single handedly started the most successful unification attempt to this day.

X/Open had been working on what was called Spec 1170. Spec 1170 is a draft set of more than 10000 APIs (Application Programmer Interfaces) which describe a Unix-like operating system. It was their latest attempt to try to create a Unix standard. When completed it was to be the basis for granting any product the right to use the Unix trademark. The Spec 1170 standard has been ratified within the X/Open body and is now called the Unix Standard Specification. However, the name ``Spec 1170" is still in

wide use.

X/Open has not announced any formal testing methods that can guarantee compliance to the X/Open standards yet. Novell has released formal methods for determining whether ports of UnixWare to non-X86 architectures can bear the UnixWare name. Testing specifications for UnixWare are actually a superset of the Spec 1170 standard before it was ratified. The UnixWare testing methods will very likely greatly influence any methods X/Open does formally adopt. Currently talks are in progress to discuss X/Open combining efforts with the OSF (Open Software Foundation) to use OSF's DCE (Distributed Computing Environment) specifications as a foundation for interoperability testing. X/Open plans to use the Unix trademark to brand several classes of Unix.

In general, Spec 1170 has not been unanimously accepted in the Unix community. Several folks have said that it can best be described as a ``historical compatibility interface." However, now that it has been ratified by the X/Open members, several commercial vendors such as Microsoft and DEC have announced that they will add Spec 1170 compliance to their operating systems when the testing process is formalized.

Many Unix advocates hoped that the System V R4 specification would be the unifying move. So far indications are that it hasn't placated enough of the Unix world for it to be accepted as the end all Unix definition.

Conclusion

Based on the history of Unix, the OS itself has been continually changing. Perhaps the OS can never be truly nailed down. With its deep roots in academia and the endless quest to add Yet Another Feature, Unix has grown beyond the normal understanding that any commercial product has. All attempts to standardize it from a commercial standpoint have faltered badly. Who knows if the existing groups can find a common ground on the endlessly shifting battlefield. Even fear of extinction by another OS has not really hastened the process much. Only time can truly tell if a standard will ever be reached. In the mean time, Unix continues to evolve.

References

1

Salus, Peter [***A Quarter Century of UNIX.***](#) Addison-Wesley Publishing Company, Menlo, CA 1994.