History, Evolution and Future of Public Access Unix Systems

By cmccabe@sdf.org

This article explores the history of public access Unix [1] systems, combining first-hand experience of contributors, historical accounts from related documentaries, and materials gathered from online archives. In exploring this history, one goal of this article is to provide overdue recognition to public access Unix systems by clearly defining them as a category distinct from bulletin board systems (BBSes), web forums, IRC channels, and other computer-based social venues. And through this history and definition, this article will make a broader argument, that Unix itself is a communications and social medium, distinct from printed text, television and the World Wide Web (WWW) -- and that it can provide a unique and powerful role in society because of the special way in which it shapes social interactions. Finally, this article aims to encourage participation in and support for public access Unix systems, partly by brainstorming on what the next generation of systems might look like.

A central thesis of what follows is that public access Unix systems have played and continue to play an important role in fostering non-commercial online communities. And because of the way that Unix shapes social interactions among users, it is particularly relevant today. As concerns rise about the exploitative, addictive and shallow world of the commercial WWW, public access Unix systems offer a meaningful alternative in which people can be valued for their interactions with others rather than for the resale value of the information a site collects on them.

With this thesis in mind, this article is written for the following audiences:

- People who are already users of public access Unix systems, both experienced and those just getting interested in them,
- Potential users in the form of people who are concerned about issues of privacy, user exploitation, and low quality, sensationalistic content of mainstream media and social media outlets.
- People involved in media studies or those concerned with the problems of corporate media in present-day society.

Table of Contents

Definition of Public Access Unix Systems

Public Access Unix Systems are defined as computers running a Unix (or Unix-like) operating system that provide shell access for free or low cost to members of the general public by remote connection for recreational, educational or other use of Unix. This is distinguished from public access Internet providers which, although occasionally Unix, primarily serve as a pass-through for Internet access. This is also distinguished from BBSes which are often not based on Unix, and which provide access to a BBS software interface rather than a Unix shell.

Similar to public access Unix systems are those that provide shell access restricted to members of a particular institution, such as a university or research organization. In fact, Unix as a social computing platform arose from this context in Bell Labs and several academic institutions in the 1970's. The distinguishing factor is that public access Unix systems are open to the general public, in much the same way as public libraries are distinguished from university libraries. Not surprisingly, public access Unix systems also tend to have a larger and more diverse set of leisure uses than institutional systems.

Another important dimension of this definition is the use of these systems for conferencing. Conferencing facilities (essentially threaded, persistent conversation forums) are not limited to public access Unix systems and in fact cross-cut many types of multi-user systems. On some systems, conferencing has become the dominant attraction, overshadowing the other uses of the system. The popularity of conferencing on some public access Unix systems is an example of their value of providing a venue for online social communities.

Origins

> "What we wanted to preserve was not just a good environment in which to do programming, but a system around which fellowship could form. We knew from experience that the essence of communal computing, as supplied by remote-access, time-shared machines, is not just to type programs into a terminal instead of a keypunch, but to encourage close communication."
>
> -- Dennis Ritchie [2]

Although time-sharing operating systems were initially designed in the early 1960s to solve the problem of efficiently sharing computer resources by multiple users, they were very quickly adapted to become tools for communication among system users. CTSS [3], the very first time-sharing system, had by 1965 introduced the commands ‘WHO’ to list other users currently connected to the system and ‘MAIL’ to allow users to exchange electronic messages. Public user profiles originated with the ‘FINGER’ command, developed originally on a DEC-10 system in 1971 [4] before coming to Unix in 1980. And the first bulletin board program was developed for the PLATO system in 1973 [5].

Unix's well-documented history began in 1969, as a descendent of the early time-sharing systems. One goal in designing Unix, as stated in the Dennis Ritchie quote above, was to facilitate collaborative work between the system users. To support this goal, even the early Unix systems had many social utilities-- including 'who', 'mail', and 'write' --allowing users to interact with each other, and many others were added by the early 1980's (Appendix A). And even the earliest Unix systems had a file system permissions scheme that allowed files to be kept private to the user, shared with a group, or made available to all users on the system.

While these utilities proved very popular, and user communities developed within these systems, access to Unix systems was largely restricted to the research institutions where they were operated because computing in general was expensive and very few people had computers and modems at home. But in the late 1970's and especially 1980's, the commercial availability of home computers and modems changed all of that, and changed the culture of computing forever.

Computer based social networking outside of research institutions first became available to home users during the early 1980's, largely in the form of bulletin board systems (BBSes). Early BBSes were simple computers (often not Unix) to which users could connect via modem. These systems rarely had Internet access and users primarily drew from local calling areas because long-distance charges were exceedingly expensive at that time. The dial-in connection gave users terminal access to the machine and a limited ability to interact through the functions of the BBS software interface. Processor and modem constraints meant that early interaction among users was mostly limited to simple exchanges of text notes -- literally analogous to a physical bulletin board.

As an example of modem constraints, the early 1980's Hayes Smartmodem supported 300 bits/sec of data transfer. The article you are reading now, in plaintext format, is about 60K in file size. It would have taken more than 27 minutes to download this article.

But despite these limiting factors, being able to interact with others through your home computer was an incredible novelty and computer-based social networking took off.

The conditions that precipitated BBSes in the early 1980's also yielded the first public access Unix systems. Like BBSes, these systems were open to home users through a modem connection. And like BBSes, the early users were primarily electronics and computer hobbyists who wanted to socialize with like-minded people around their hobby interests. It is true that some early public access Unix systems offered BBS services and therefore acted as hybrid systems -- the dividing line between these two types of system was sometimes blurry.

Early Public Access Unix Systems

There is some question about which was the first public access Unix system [6], but most sources point to one set up in 1982 by Mike Myers in Ann Arbor, Michigan. M-Net, as it was called, ran System III Unix on an Altos 68000 and primarily supported conferencing through two programs called PicoSpan and Party [7].

Also in 1982, Randy Seuss started up a separate public access Unix system in Chicago [8]. Originally called wlcrjs, it changed its name to Chinet around 1984. Chinet ran Unix on a Compaq "portable" with a 4 MHz 8088 processor, 640k of memory, and a 10 MB hard drive. It supported conferencing on variants of PicoSpan, and also offered Internet access including e-mail and newsgroup postings over UUCP [9].

Throughout the 1980's, many other public access Unix systems were set up. Some rose to notoriety, such as Sausalito, CA's The WELL (1985) and New York City's Big Electric Cat (1987). So many systems had sprung up by around 1985 that an enthusiast named Wayne Warble began sharing a compiled list of "Public/Open Access Unix (*NIX) Sites" that he called "nixpub" sites. In 1989 maintenance of the list passed to Phil Eschallier (now of 10 Types Inc. and RCH Solutions) who regularly distributed it on USENET [10]. The list consisted of 35 systems in 1987, and rose to 156 systems by 1995. Appendix B will serve as a continually updated list of known public access Unix systems, largely sponging off what was recorded in these nixpub lists for the historical systems.

Like M-Net and Chinet, many of these early systems offered shell access but appear to have been primarily used for their conferencing forums or, later, USENET, email, and file transfer (over UUCP [9]). Users were typically people who had day jobs in academia, the government, or corporate research, and conferencing topics tended to be of a higher technical or intellectual level for that reason.

Computer hobbyists were the most common user of these early public access Unix systems, and their interests included not just computer-based social networking, but also the architecture and operation of the computers themselves. These more technically minded people tended to be frustrated with what they saw as the toyish power of the single-tasking home computers of the day (e.g. the TRS-80), and saw modem-accessible public access Unix systems as an opportunity to use a "real" operating system. The Unix OS was prohibitively expensive for home users during that time [11], so public access Unix systems provided these enthusiasts a welcome opportunity to use it.

There was wide diversity among these early public access Unix systems, from small-scale hobby projects that attracted a scattering of local users, to heavily used conferencing forums. M-Net, for example, was quite popular, with around 25,000 users at one point. By the late 1980's some cities had access to a Telenet service called PC Pursuit [12] that allowed users to

dial into a local number and get access to systems in more than 20 other end-point cities across the United States. This allowed some systems to cultivate user bases from a much broader area.

Public access Unix systems were similar to BBSes in their use for text-based conferencing and file sharing, but Unix offered one big difference; at its core, Unix was not built for socializing alone, but rather for programming. As John Biggs put it in a 2011 TechCrunch article [13], "Dennis Ritchie and [Ken] Thompson worked together to build Unix as a hacker's paradise, a place to test small programs and share the results". While BBSes have sometimes been referred to as "the original computer-based social network" [14], one way of describing public access Unix systems is that they were the original online "hackerspaces". Hobbyists could work on coding projects in an environment that revolved around teaching and learning from each other and enjoying conversation with interesting people. As will be discussed below, this communal coding focus is seeing a resurgence in popularity more recently.

Change Over Time

In the late 1980's and early 1990's a number of new public access Unix systems came online, sometimes evolved out of old BBSes and sometimes as de novo services. These systems combined the social spirit of the BBS era, the thrill of accessing the Internet's exploding amount of information, and the innovative spark of a new and better equipped DIY programmer generation.

Some of the main influences on these systems were formative changes to the Internet, increases in modem speeds, and growth and diversification of user numbers and interests. One big change was that many systems began providing Internet connections, allowing their dial-up users to reach a broad variety of other networked resources. And for another, many new social utilities and other programs were added to the Unix environment --IRC, USENET, Gopher, and others. And of course, in the early 1990's the Linux operating system was developed, and the corresponding explosion of the hobbyist programmer and sysadmin movement drew even more people into running and using Unix-like systems [15].

One of the key technological changes impacting public access Unix systems was an increase in modem speeds. From the period of the early 1980's through the mid 1990's, dial-up speeds continued to rise, going from a now hard to imagine 300 bits/sec speed of early Hayes Smartmodems to 28,800 bits/sec by 1994. And with these speed increases came changes in usage patterns. Early access was so rare, and modem speeds so slow, that connected time was precious. Every word typed, every post made was thoroughly thought out before sending. Content was terse and purposeful.

But in the mid-1990's, dial-up Internet access from services like AOL or CompuServe had become a strong force in the landscape. Along with the rapid growth of information and

resources on the Internet, dial-up Internet service brought online a wave of new users from the general public. In addition, the WWW and the Mosiac web browser gave people a new medium for using it. Local systems such as BBSes and public access Unix systems were no longer the only attraction in town. The era of the BBSes precipitously declined [16], and public access Unix systems felt the effect as well. This change was reflected in both declining user numbers and shifting patterns of system usage. As 1980's veteran Phil Eschallier put it, "we went from brilliant physicists, compsci, EE, etc. to pictures of cats" [17].

This was also a period in which malicious user behavior became a problem. Spam mailers, script kiddies, and people hosting and sharing illicit files became a problem. But despite this the value of community still shone through. As Ross Turk (now director of Red Hat portfolio evangelization) put it in describing his first experiences on Nyx.net, "You could tell it had already been the subject of much compromise - there were a bunch of disabled services and cumbersome processes (like a sign-up form requiring notarization) already in place. But it wasn't the system that made Nyx special, it was the community that it allowed to be created" [18]. For example, NyxMud's support for user scripted game elements was replaced with the "Wand of Creation", a limited options menu of sorts; but still, "it was a busy place with a vibrant community of players and wizards."

The 1990's was a time of reckoning for public access Unix systems. Some systems lost their user bases and closed as the meowing of cat memes replaced interesting discussions. Some others saw that providing Internet access was the bigger draw for new users, and they became early Internet service providers (ISPs), many of which were either outcompeted or consolidated into some of the big ISPs today. And yet others carried on similar to before, but with a more refined sense of purpose. The ones that persisted can be thought of as a second generation.

The first generation public access Unix systems had been largely focused on the novelty and fun of newly accessible computer-based socializing. These second generation systems continued that, but also began to develop a sense of larger purpose. Grex.org (1991), SDF.org (1987), and Nyx.net (1988), all of which still exist today, were among this second generation. A quick scan through their mission, principles, and philosophy statements highlights a common theme: community education, public education, spiritual and intellectual enrichment, promoting a better-informed citizenry, collaboration, non-commercialism, choice, privacy, cultural enrichment, scientific research.

These systems were consciously focused on having a positive impact on society through the social medium of the Unix system. They saw change on the horizon from the now commercial Internet and knew they had something important to protect from it. Still, the overall trend of usage was one of decline and some began to see public access Unix systems as just a quaint, nostalgic reminder of a time gone by [19].

Current State

In 2014, a funny thing happened. As Facebook soared past 1.5 billion users, and snatched up Whatsapp for $19 billion; as the world reeled in Edward Snowden's revelations about the U.S. surveillance apparatus; and as remaining public access Unix systems continued to see flat or declining usage; a person by the name of Paul Ford [20] (then of ftrain.com blog notoriety) registered the http://tilde.club URL and pointed it at a Fedora Linux server with open registration for anyone to use. He sent out a casual tweet to his Twitter followers and, to his surprise, within a few days tilde.club had hundreds of new account registrations.

Users logged on to tilde.club and started putting the system to work. They created and shared ascii art or simple retro-cool pages; they wrote clever command line programs to impress other users; they blogged about their interests; they emailed, chatted and wall'ed each other. After a while, the number of new users grew so high that Ford had to restrict sign-ups. By no means were these usage numbers notable compared to even moderately sized commercial websites, but the creative energy pulsing through the tilde.club community was something different and special.

Although tilde.club had to close its doors to new sign-ups, the interest of many members of the current generation had been piqued and others excitedly opened similar sites [21]. One example of this was tilde.town, created by admin vilmibm who took inspiration and general system design from tilde.club. Similar to tilde.club, registration was open to anyone, and users were allowed (encouraged!) to express themselves freely with the utilities a standard Linux box makes available.

To this day, tilde.town is still active, very active, consisting of users with all levels of experience, from non-developers who just want to socialize, to both novice and highly experienced programmers. Someone is always working on and testing out new projects, often resulting in hilarious digital whirligigs or rube goldberg machines of code that other users can play with. For example, one user created a tilde-coin that is awarded for answering distorted captcha questions from an IRC tilde-bot. Another user created a multi-user virtual world of prose that continuously expands as users log in and describe more interconnected spaces. And another user created a retro BBS called BB&J. There is a curses-based "botany" app that lets you test your digital horticulture skills. Or you can homestead in the virtual world of cadastre. Eventually you will be able to grow fields of crops through a (currently in-development) project called asciifarm. (The developers of these programs are listed in Appendix C)

There are so many other little projects on tilde.town that a new user can take weeks just to get an idea of what is available. Not all projects are perfect, of course; some are buggy, others are confusing to use. But, they were made to share with others, and the developers learned a lot while building them (somewhat analogous to the paper you're now reading).

Another interesting and active system is called Devio.us. This system hosts a community dedicated to open source software and the Unix operating system, particularly OpenBSD. In addition to an expertly managed shell service, it has an active forum and IRC channel through which geeks, developers, IT professionals, and enthusiasts chat, interact, and network around shared interests.

Meanwhile, over at SDF.org there continue to be thousands of logins a day. Captained by sysadmin Stephen Jones (smj), SDF has developed into a fully articulated social community, with free and paid membership tiers that provide varying levels of system access. It supports numerous ways for users to interact, including IRC, an old school bulletin board program called 'bboard', and a multi-room chat program called 'commode'; among great but lesser known utilities like 'oneliner' and an anonymous blogging tool called 'happening'. It also hosts several special interest groups including a ham radio users group (SDFARC), a python study group, a group of gopher protocol enthusiasts, a crowd of Minecraft players, and a user-driven Internet radio service ("anonradio") on which any user can sign up to broadcast their own shows. Among the system's users are digital artists, computer historians, web developers, gamers, and of course many, many programmers. All experience levels are represented, from young learners to veterans and computer science professors.

One aspect that is fairly unique to SDF.org is its large number of users with a passion for retro computing and telecommunications. While retro computing is only one of the many user interests on SDF, it is one that has arisen partly because of the system's long and continuous history (more than three decades now). Users frequently converse about how to bring various pieces of ancient hardware back to life, or how to use old networking protocols. Users also occasionally post pictures, documentation or comments about old telephone technology. In addition, through a connection with Seattle's Living Computers Museum, sysadmin smj has facilitated online access to old systems like a CDC 6500 supercomputer and a simulated PDP-10 running the TOPS-20 operating system, both of which SDF users can log into. And of course, you'll also see a lot of appreciation for the timeless value of classic text-based games [22].

Through the various retro computing activities on SDF, you sense a deep respect for facets of computing and the Internet that are currently being paved over by corporate saturation. If you get into discussions with SDF users, you'll find that concern with the broader direction of the Internet runs deep.

--

There are not an overwhelming number of places like tilde.town or SDF online these days, but there are still quite a few (Appendix B), and the communities are welcoming. In them, you'll find an online social space for people who are not amused by the mainstream social media; for

people who are driven by the spark of curiosity and the interest to work on hobby computing projects in a collaborative, sharing environment with others.

Broader Context of Current State

Despite the enthusiasm of this article, public access Unix systems are now a small corner of the Internet, and users by no means restrict themselves to the bounds of these systems. For one, there is a lot of traffic between different systems. A user on on grex.org, for example, might simultaneously be logged into tilde.town, or someone from Nyx.net on Skylab.org. And for another, these users traverse the broader Internet. But in response to the commercially driven WWW, many share a particular view on the increasing dominance of corporate control over mainstream media and social networking sites.

These concerns are reflected in comments by the inventor of the WWW himself, Tim Berners Lee: “The system is failing. The way ad revenue works with clickbait is not fulfilling the goal of helping humanity promote truth and democracy. So I am concerned” [23].

Adding to the concerns of Berners Lee is a growing litany of complaints. You simply cannot use the web without being tracked, with or without your consent, [24] and there is no telling what the profiles built about you are being used for. Even when you do agree to the legalese of a given terms of service, organizations often have a flippant attitude about ensuring that your data is kept safe [25]. And the content coursing through the WWW is problematic too. Within social media sites, discourse is often reduced to "echo chambers" in which facts and truth are distorted to support tribalistic finger pointing [26]. Further, services like Twitter and Facebook have become vectors for corporate and government propaganda campaigns [27]. And mainstream news outlets have allowed the crowding out of real journalism by sensationalist, clickbait news [28].

Underlying all of this is a deepening awareness that users are being exploited. Brinknews.com reporter Giovanni Buttarelli describes this user exploitation as "digital sweatshops" in the sense that "the digital information ecosystem farms people for their attention, ideas and data in exchange for so called “free” services" [29].

As awareness of these problems rises, more and more people are getting fed up with the commercial WWW. This presents a special opportunity to raise awareness of public access Unix systems which can offer a non-commercial, individual-oriented and intellectually stimulating refuge from the mainstream Internet media's disappointing circus of exploitation and sensationalism.

Unix as a Distinct Medium

One of the core arguments of this article is that public access Unix systems are a unique form of social and communications media, different than television or the WWW. This is important for explaining why these systems can serve as a sanctuary for some forms of social life in the face of the commercial noise of the WWW. Rather than lending itself to disengaged consumption of soundbites and video clips, Unix encourages the user to put in effort. You can't get started using Unix if you're not willing to read manual pages, to think about how and why to use various utilities, to break and fix things through good old trial and error, to (gasp!) learn a programming language or two, or even to learn how to bend the tools to your own needs. Once you get over this learning curve, Unix is actually pretty easy to use, and extremely powerful; but the learning curve is a barrier to entry that weeds out users who aren't willing to put effort into their engagement with the system.

In addition to the rite of passage of the Unix learning curve, this article also argues that the Unix philosophy [30] itself provides a perspective that helps a thinking person cut through the "fake news" and propaganda of mainstream media. This philosophy is often stated something like "create small, simple tools that do one thing, and do it well." Or, put another way, use the KISS principle -- Keep It Simple, Stupid. This approach is focused on software or system design, but can also be used to analyze these things. By breaking a subject down into component parts, complexity can be made analytically tractable and logical inconsistencies can be more easily called out.

But why might this perspective extend beyond software? Just as any form of habit can be developed through repetitive action, regular practice of building and analyzing software in line with the Unix philosophy builds mental habits too, specifically always desiring to clearly understand how something works rather than passively accepting a hand-waving explanation about it. Analytical thought habituated on this philosophy might also yield a healthy skepticism when faced with the red, blinking headlines from foxnews.com or cnn.com, or the sourceless comments floating through Facebook and Twitter. Put crudely, programming the Unix way has a side effect of giving you a good BS detector.

So this effortful medium shapes its own use in a few ways. For one, it weeds out low-effort participation. For two, it pushes its users to practice an analytical, skeptical mindset. And for three, it provides learned users with a diverse palette of tools and utilities for collaboratively building and sharing creative output.

The Future of Public Access Unix Systems

Up to this point, this article has attempted to define and illustrate the concept of public access Unix systems through recounting some of the history. It has also attempted to frame these

systems as an important form of community-oriented communications and social media, distinct from mainstream commercial sites. But the future is where the action is. Rising concerns about the corporate-dominated Web are creating conditions in which public access Unix systems may appeal to an even broader audience; conditions which certainly underscore the value of these systems. The time is right for users to take a deeper appreciation of these systems, and for new users to join and help these systems grow by creatively participating.

So, what next? Where do we go from here?

This article hopes to stimulate discussions around a vision of the future for public access Unix systems; a discussion that needs active participation from many people. This may begin in part by identifying the enduring characteristics of current systems, as well as the challenges they face in sustainability. It is also helpful to look across existing systems to identify characteristics that may hail the next generation, those systems that will carry the torch of providing engaging social spaces for computer hobbyists and enthusiasts who aren't amused by the "tittytainment" [31] of modern corporate media.

Enduring characteristics of current public access Unix systems

Informally surveying users of public access Unix systems tends to identify a few common characteristics that are considered advantages over other online social venues. Many of these are found individually in other online platforms, but one topic-- that these systems can serve as an online hackerspace --is one for which Unix has no strong competitor, and is the central value around which the others revolve. Some of these characteristics are listed here.

Privacy and data ownership. Given that there is so much concern about the practices of large commercial social media sites, it is not surprising that users appreciate an alternative that respects and puts them in charge of their data. It goes a long way when a system administrator pledges commitment to respecting user privacy and data ownership. See SDF.org's "ethical social network" page [32] as a great example of this.

Hackerspace. Unix is not for everyone but rather appeals to that sliver of society that wants to creatively engage with building things on the computer and to leverage powerful software tools in a collaborative environment. Unix is an environment for both exploration and development of software, and for sharing of solutions. This inventor's lab or engineer's playroom is an increasing attraction of public access Unix systems, and one that allows them to provide an online parallel to physical makerspaces.

Retro computing. Unix provides a continuous and well-documented lineage throughout much of the history of computing. Because of this, and because it is still in very active development, public access Unix systems provide an excellent environment for retro computing enthusiasts. As described above, although much more than just retro computing, SDF.org is again a great example of this.

Unique subculture. Facebook has the network effect of "everyone you know is already using it". By contrast, public access Unix systems have the draw of a certain subculture that is both hands-on with computing technology and fully aware of the exploitative nature of mainstream media culture. For people who feel an affinity for this subculture, public access Unix systems have their own attractive network effect. The networks also tend to be small enough that interactions between people don't get drowned out in the racket and caterwauling of advertisement-riddled, million-plus user web forums.

Embracing diversity. The raw text of the Unix shell interface allows users to meet on an intellectual level, where the only common form of impatience is with people who demonstrate malicious behavior or an unwillingness to do work (e.g. those who don't RTFM). This provides a valuable outlet for people who otherwise might not have as many opportunities to express themselves socially. This is not to say that users of these systems are free of biases-- socially ignorant and offensive people can be found everywhere, of course --but Unix provides an environment where ideas and abilities can be valued above things like age, skin color, sexual orientation, physical ability, religion, or politics.

Unix as medium. As stated above, this article aims to add to this list of characteristics by raising awareness about Unix's uniqueness as a communications and social medium. The primarily text-based tools for social interaction on Unix systems are a double-edged sword. The initially inscrutable shell interface serves as a repellent for many potential new users on the one hand. But on the other hand, the medium of the interactive command line is a large part of what gives Unix users some resistance to the entertaining-ourselves-to-death effect of TV and the commercial WWW.

Challenges facing current public access Unix systems

Putting enthusiasm in check, the reality is that there are far fewer public access Unix systems now than there were in the 1990's; and some remaining systems are struggling to survive. There are strong headwinds to establishing a sustainable public access Unix system, including time for system upkeep, system administrative knowledge, cost of hardware and hosting, and the vagaries of maintaining an active and interesting user base. For future public access Unix systems to be successful, organizers must put serious consideration into establishing a sustainability model that addresses all of these concerns.

Sysadmin time and knowledge. A bare minimum requirement for running a public access Unix system is, of course, the need for a sysadmin (or a team) with the time and ability to manage the system. This can be challenging because support is almost invariably based on spare-time, volunteer effort, and people with this level of experience usually already have time-intensive day jobs. Many existing systems have started to show their age as long-standing sysadmin teams have moved on or been pulled more into other parts of their lives. Many former systems went offline because they were unable to replace trusted sysadmins.

Operations costs. Another obvious challenge facing public access Unix systems is the recurring costs of the network connection, system hardware, and other resources. Some systems cover these costs out of the organizers' pockets. Others raise money through membership fees. Registering as a non-profit organization (e.g. 501(c)(3) or 501(c)(7) in the U.S.) can facilitate receiving donations as well. SDF.org, Grex.org and Nyx.net are set up this way.

Unlike Facebook or Google, public access Unix systems do not earn revenue by selling user data to the highest bidder. This is a strong attraction for these systems, but it increases the need to find other sources of support.

User base. There are many free or cheap Unix shell providers online, but what sets a good public access Unix system apart is its user community. For a new system starting up, attracting a critical mass of users is not easy. And for established systems, inability to retain active users or to attract new users can mean that an existing user base may lose interest or age out.

Further, an interesting and welcoming user base is itself an attraction, but it only takes a very small number of toxic users to poison the culture. Providing the necessary mechanisms to support a healthy social environment and to reform or isolate toxic users requires focused attention.

Trust. An absolutely critical topic for public access Unix systems, related to both the sysadmin and user base challenges is trust. For a system to survive over time, it must eventually manage turnover of system administrators. But giving a new arrival that level of system access requires an exceedingly high level of trust in the new person's competence, reliability, motivations, and dedication to the system's founding principles. How can you establish this trust?

Likewise, unless a system is prohibitively locked down, users have access to utilities that can be abused to degrade system performance for others, or worse, drag the system into legal problems (e.g. spam, blackhat hacking or the storage or sharing of illegal content). With semi-anonymous users potentially logging on from anywhere in the world, how can good behavior be enforced?

Characteristics of next-gen public access Unix systems

The pure text-based forums of early public access Unix systems or BBSes were once a hot attraction because there were no alternatives online. Nowadays, such forums alone are unlikely to attract many new users from the general public because similar venues are available as web forums, IRC channels, or elsewhere. Instead, this article argues that a viable future direction for public access Unix systems is to embrace and build on the unique value of Unix in the model of an online hackerspace. And examples of this direction do already exist.

Of course, a pure hackerspace model is not the only option. SDF.org, for example, already has a well-established system built around a menagerie of user interests. So long as sysadmin smj doesn't engage in any risky undercarriage inspections of moving buses [33], SDF looks like it will be around for a long time to come. And yet other systems have attracted and maintained an active user base largely because of their interesting real-time chat forums (e.g. The WELL, ECHO NYC, or Blinkenshell.org).

Taking into account the challenges outlined above, and considering examples thus far reviewed, below are some characteristics that may be of value to next-gen public access Unix systems.

Hackerspace model. Taken to its extreme, an online hackerspace could open up all aspects of the system to continual development and improvement by its users -- from user-developed IRC bots, games, utilities, and artsy code novelties, to the infrastructure-as-code of the system itself. At one end of this spectrum are systems like tilde.town on which users develop and share fun IRC bots and other novelties. At the other end, an exciting example in progress is hashbang.sh which is developing the hashbang system as an open-sourced code repository [34] and serving as a training ground for new sysadmins.
To quote lrvick, hashbang.sh president and admin:

> "Engineering is an artistic craft that is most rewarding when shared with others. Hackerspaces-- be they physical or digital --give people the resources, mentorship, and accountability to learn and create whatever they want without the constraints of corporate shareholders, academic guidelines, or NDAs. Find a community to push you to create, and to share in the enjoyment of your creations. It is a positive feedback loop that can last a lifetime."

This model is especially attractive now, as computing skills are important for more of the general public than ever before. Public access Unix systems can provide a training ground for interested users at all skills levels, just as physical makerspaces bring together experts, learners, and tools in a project-oriented environment.

The hackerspace model also opens the doors to other new public access Unix systems based around different interest topics. Imagine for example, teaching and learning communities built around game development, or digital art, or data science, or security.

Emphasis on learning. Hand in hand with the hackerspace model, a Unix shell in an environment populated by other enthusiasts is a valuable resource. In general, the goal should be to establish cultural norms of learning-by-doing, collaborating on learning, and helping others to learn. With the right structure, this might even extend to some form of an **apprentice/mentorship model**. Such systems might involve users in projects through which they can learn skills, demonstrate abilities, and grow upward toward useful contributors to the system itself.

While public access Unix systems are often built on charitable values, this by no means implies that users take vows of poverty upon signing up. To the contrary, many users are rightfully motivated by the career advancement they may achieve based on skills and experience gained from within these systems. If a system develops a reputation for putting out hot IT job candidates, this would further the attraction for new users, both those coming to learn and those who have a passion for teaching.

Mechanisms for establishing trust. As described above, the need for trust among sysadmins and users is critical for a public access Unix system to operate successfully. But there is no simple formula for establishing trust. The social science literature has examples of how trust can be established within communities, and many suggestions revolve around frequent and long-term communication among community members. One mechanism that is often used on IRC boards is to elevate user privileges over time as a user demonstrates good intentions and abilities. This mechanism might be extended to use demonstrated skill set as an additional criteria for elevation.

Users can play a role in maintaining expectations and levels of trust by self-policing each other as well. For example, some core Unix utilities like 'top' can help users monitor other users and remind them when they're crossing certain lines. Setting high expectations for users to take responsibility for a healthy system culture is important.

Cost control. Unless a system has a sizable user base that pays membership fees, or it has a generous benefactor (often the system organizer), a strategy for cost control is very important. Some existing systems cling to running their own hardware, for nostalgia or other legitimate interests, but this model can incur considerable costs. Other systems are now experimenting with cloud hosting. In a cloud model, a goal should be to develop the ability to scale usage up or down relative to demand and/or available funds. This implies a system for monitoring and automatically scaling based on environment conditions. Developing infrastructure-as-code that includes a bare-bones configuration should be a hackable goal for users.

User base life cycle. If multi-generational lifespan of a system is a goal, it is important to balance the need to attract new users against the need to carefully limit new membership to people who will be a good fit in the system's culture. Using hashbang.sh as an example again, providing clever hurdles to entry can serve as one such selective mechanism. On the other hand, producing cool stuff and giving people the opportunity and tools to develop their skills will attract new users.

Privacy protection and non-exploitation of users. As described in the Enduring Characteristics section above, a commitment from system organizers and administrators to security, privacy, and respect for user ownership of their data is a fundamental pillar of a good public access Unix system. Along with a solid mission statement, discussed next, this should be explicitly described for users to read.

Mission. - All of these goals and the challenges listed further above are more easily met when both the user base and administrative team are dedicated and motivated by a shared sense of purpose. A clearly phrased mission statement that resonates with users is invaluable in galvanizing a public access Unix system's user base and staff.

This paper was written under the strong belief that public access Unix systems can provide a valuable resource that resonates with the very core of human nature -- the need for community, and the sense of having a valuable role in that community that is based on one's interactions with others rather than the market value of the data in one's user profile. A mission statement revolving around this belief, and providing people a refuge from everything that is wrong with the commercial WWW, can go a long way to motivating users and staff.

So a condensed suggestion for next-gen public access Unix systems is:
Embrace the learning, doing and teaching of the hackerspace model; emphasize a mission that resonates with users; and involve motivated users in establishing trust and driving operations costs down. And never forget that an active, welcoming community is the main attraction, so make sure everything else revolves around it.

Summary

This article has surveyed the history of public access Unix systems, a category of online social space that is overdue for well deserved recognition. Further, the article has made an argument for their potential value for modern society and provided some ideas for future systems.

From the first systems in the early 1980's until the present day, public access Unix systems have seen wild swings in popularity and witnessed radical changes to society that have been greatly impacted by the Internet.

As print and television have both in turn surrendered their roles as the dominant forms of media in the U.S., and as the WWW has taken over, mainstream public discourse has lost any semblance of intellectualism [35]. For anyone concerned with this trajectory of events, and particularly for those interested in computing, public access Unix systems provide not only a refuge from the storm, but a vibrant and engaging alternative communications and social medium.

If you are motivated by this vision, there are great opportunities in front of you to enjoy these public access Unix systems, or even to help them survive. Sign up and log into a system (see Appendix B for some choices) while keeping in mind the challenges these systems face (see the Challenges section above). Be an interesting, active user. Spend effort learning, and encourage others who are trying to learn. Stop and spend some time to appreciate what others are doing. Help to establish trust among the user community by setting a good example for

others. Consider donating. Be vocal (but tactful) about suggestions for system improvements. Volunteer your skills.

This article will end with a proverbial glass of water that is either half full or half empty, depending on how you want to see it. Concerns are rising about the growing influence of corporate media giants, sometimes referred to as "BAADD" (big, anti-competitive, addictive, and destructive to democracy) [36]. Their negative and exploitative impact on individuals, on public discourse, and on society in general is increasingly obvious. The solution to this problem does not begin with finger pointing at big corporations. It begins with the individual who rejects passive media consumerism, who instead consciously follows their own path.

Mixed amongst the crowd of users building fascinating software or leveraging the power of public access Unix systems in other ways, you'll find an increasing number of people with a keen awareness of how important this is.

Acknowledgements

Thank you to everyone who provided feedback on drafts of this paper and helped illuminate parts of this history. I am extremely appreciative of your input and feedback. In particular, I would like to thank Grex user *papa* who provided extremely valuable formative input on the paper. Several directions of the paper's development were based on his input. Writing this paper has been a learning process for me, and that will not end. I would like to continue revising and updating this article based on new input and aspects of the history that have been omitted. Please contact me at cmccabe@sdf.org with any feedback, suggestions, or questions. --cmccabe

References

[1] In this article, the term 'Unix' is used in a generic sense to refer to any Unix-like operating system, regardless of whether or not the system in question has been certified by The Open Group, including FreeBSD, Linux distributions, macOS and other proprietary implementations, NetBSD, OpenBSD, etc.

[2] The Evolution of the Unix Time-sharing System | Dennis Ritchie | Presented to the "Language Design and Programming Methodology" conference | Sydney, Australia | Sep, 1979 | http://cm.bell-labs.co/who/dmr/hist.html

[3] Compatible Time-Sharing System | Wikpedia https://en.wikipedia.org/wiki/Compatible_Time-Sharing_System

[4] History of the Finger Protocol | Rajiv Shah | Jun 2, 2000 | Draft Paper http://www.rajivshah.com/Case_Studies/Finger/Finger.htm

[5] PLATO | Wikipedia
https://en.wikipedia.org/wiki/PLATO_(computer_system)

[6] "Notes On The World's First PA Unix System" | Slashdot post/discussion | Mar 19, 2000 |
https://tech.slashdot.org/story/00/03/17/0450236/notes-on-the-worlds-first-pa-unix-system

[7] On the origins of MNET...

"The Once and Future M-Net " | James Howard | Mar 14, 2000 | Great Green Room |
http://rootprompt.org/article.php3?article=170

"A Partial History of Computer Conferencing in Ann Arbor" | Jan Wolter | ~1997 |
https://web.archive.org/web/20160805221621/greatgreenroom.org/cgi-bin/bt/backtalk/wasabi/begin?item=19

Arbornet and M-Net Records: 1985-2002 - History | University of Michigan, Bentley Historical Library
https://quod.lib.umich.edu/b/bhlead/umich-bhl-2012017?byte=70603759;focusrgn=bioghist;subview=standard;view=reslist

[8] "Chinet -- Public Access since 1982" | https://www.chinet.com

[9] UUCP (Unix-to-Unix-Copy-Protocol) was, to quote Wikipedia, "a suite of computer programs and protocols allowing remote execution of commands and transfer of files, email and netnews between computers"
https://en.wikipedia.org/wiki/UUCP

[10] Many of the original 'nixpub' lists can still he found here:
https://groups.google.com/forum/#!searchin/comp.misc/nixpub%7Csort:date

[11] Unix and Personal Computers: Reinterpreting the Origins of Linux | Christopher Tozzi | Channel Futures | Apr 27, 2015 |
http://www.channelfutures.com/open-source/unix-and-personal-computers-reinterpreting-origins-linux

[12] PC Pursuit | https://en.wikipedia.org/wiki/Telenet#PC_Pursuit

[13] "What Can We Learn from Dennis Ritchie" | John Biggs | Tech Crunch | Oct 15, 2011
https://techcrunch.com/2011/10/15/what-can-we-learn-from-dennis-ritchie/

[14] "The Lost Civilization of Dial-Up Bulletin Board Systems" | Benj Edwards | The Atlantic | Nov 4, 2016

https://www.theatlantic.com/technology/archive/2016/11/the-lost-civilization-of-dial-up-bulletin-board-systems/506465/

[15] (Linux and) the Enduring Magic of Unix | Ross Turk | Jan 1st, 2018 | Red Hat Enterprise Blog | https://rhelblog.redhat.com/2018/01/12/linux-and-the-enduring-magic-of-unix/

[16] BBSes declined but never went entirely extinct, and there are a number of people still passionately dedicated to maintaining the lineage and historical documentation. In fact, both the history and current state of BBSes are another fascinating topic. Like public access Unix systems, BBSes provide an online social medium that is an alternative to the commercial WWW. Nowadays, many people use BBSes for that alternative purpose, but many users are driven by the nostalgia and retro-computing aspect. So big is the retro computing interest that new hardware has been released to mimic old modem connections (see the WiFi232 link below). There are many great historical summaries of BBSes online, so this article steers clear of replicating any of that content. Instead, here are some pointers, if you're interested.

Bulletin Board Systems, History | Wikipedia |
https://en.wikipedia.org/wiki/Bulletin_board_system#History

"The BBS Documentary" | Jason Scott | May 2005 | http://www.bbsdocumentary.com
A must see!

The Wonderful WiFi232: BBSing Has (Literally) Never Been Easier | Blake Patterson | May 30, 2017 | Byte Cellar |
https://www.bytecellar.com/2017/05/30/the-wonderful-wifi232-bbsing-has-literally-never-been-easier/

BBS Corner
http://www.bbscorner.com/
"...Helping BBS Sysops & Users since 1996"

BBSing... All About Bulletin Board Systems
http://www.bbsing.com/

Telnet BBS Guide: https://www.telnetbbsguide.com
"the largest active listing of Dial-Up and Telnet accessible Bulletin Board Systems on the Internet"

[17] Phil Eschallier | Personal communication, Jun 5, 2018.

[18] Ross Turk | Personal communication, Jun 4 2018.

[19] In Noisy Digital Era, 'Elegant' Internet Still Thrives | Todd Bookman | Apr 17, 2012 | NPR - All Things Considered |
https://www.npr.org/2012/04/17/150817325/in-noisy-digital-era-elegant-internet-still-thrives

[20] "I had a couple drinks and woke up with 1,000 nerds" | The story of Tilde.Club
Paul Ford, Oct 9, 2014 |
https://medium.com/message/tilde-club-i-had-a-couple-drinks-and-woke-up-with-1-000-nerds-a8904f0a2ebf

[21] Other Tildes | http://tilde.club/%7Epfhawkins/othertildes.html

[22] On ASCII Games...
SDF's ASCII Game Page | http://asciigames.com
Text-based Games | Wikipedia | https://en.wikipedia.org/wiki/Text-based_game

[23] Tim Berners-Lee on the future of the web: 'The system is failing' | Olivia Solon | The Guardian | Nov 15,2017 |
https://www.theguardian.com/technology/2017/nov/15/tim-berners-lee-world-wide-web-net-neutrality

[24] Surveillance and user tracking
EPIC Electronic Privacy Information Center | Online Tracking and Behavioral Profiling
https://www.epic.org/privacy/consumer/online-tracking/
and
Online Behavioral Tracking | Electronic Frontier Foundation
https://www.eff.org/issues/online-behavioral-tracking

[25] Experts Call Facebook's Latest Controversy a Social Media 'Breach of Trust' | Linsdey O'Donnell | Mar 20, 2018 | Threat Post |
https://threatpost.com/experts-call-facebooks-latest-controversy-a-social-media-breach-of-trust/130572/

[26] "Escape the echo chamber" | C Thi Nguyen | 09 April, 2018 | Aeon |
https://aeon.co/essays/why-its-as-hard-to-escape-an-echo-chamber-as-it-is-to-flee-a-cult

[27] Facebook and Twitter are being used to manipulate public opinion | Alex Hern | Jun 19, 2017 | The Guardian |
https://www.theguardian.com/technology/2017/jun/19/social-media-proganda-manipulating-public-opinion-bots-accounts-facebook-twitter

[28] How technology disrupted the truth | Katharine Viner | Jul 12th, 2016 | The Guardian |
https://www.theguardian.com/media/2016/jul/12/how-technology-disrupted-the-truth

[29] We Are All Working in Digital Sweatshops | Giovanni Buttarelli | Brink News | May 21st, 2018 | http://www.brinknews.com/we-are-all-working-in-digital-sweatshops/

[30] "The Art of Unix Programming - Chapter 1. Philosophy" | Eric Raymond | https://homepage.cs.uri.edu/~thenry/resources/unix_art/ch01s06.html
Also available in printed form (Addison-Wesley, 2003)

[31] "Tittytainment" is a term coined in Martin and Schumann's (1997) book "The Global Trap: Globalization and the Assault on Democracy and Prosperity", referring to the disengaging effect of passively watching TV or surfing the web -- like an infant becomes docile and satisfied while nursing. It is similar to older phrases like the Latin "*panem et circenses*" (bread and circuses), or Karl Marx's "opiate of the masses".

[32] SDF - The Ethical Social Network | http://sdf.org/?tutorials/social_network

[33] Bus Factor | Wikipedia | https://en.wikipedia.org/wiki/Bus_factor

[34] hashbang.sh - github repo | https://github.com/hashbang

[35]
"Amusing Ourselves to Death" | Neil Postman | 1985 | Viking Penguin

"Amusing Ourselves to Death - Part II" | Nick Ingram | ClearThinking.co | July 18, 2016 | http://clearthinking.co/amusing-death-part-two/

"My dad predicted Trump in 1985 – it's not Orwell, he warned, it's Brave New World"
Andrew Postman | The Guardian | Feb 2nd, 2017 |
https://www.theguardian.com/media/2017/feb/02/amusing-ourselves-to-death-neil-postman-trump-orwell-huxley

[36] "Competition in the digital age - How to tame the tech titans" | The Economist | Jan 18th 2018 |
https://www.economist.com/news/leaders/21735021-dominance-google-facebook-and-amazon-bad-consumers-and-competition-how-tame

[37] "The Simple Unix Command That Gave Birth to Social Networking" | Andre Torrez | 12/19/2011 | Gizmodo
https://gizmodo.com/5869385/the-simple-unix-command-that-gave-birth-to-social-networking

---

Some relevant sources that are very interesting, but not (yet) referenced in the text:

London Hackerspace | New User Induction Program (implying the type of responsibility needed from users)
https://wiki.london.hackspace.org.uk/view/Induction

Teens are abandoning Facebook in dramatic numbers, study finds | Olivia Solon | Jun 1, 2018 | The Guardian |
https://www.theguardian.com/technology/2018/jun/01/facebook-teens-leaving-instagram-snapchat-study-user-numbers (This article summarizes some of the Pew Research Center study.)

Rob Sweeney - some thoughts about the history of Big Electric Cat as one of the first public access Unix systems (Wayback Machine)
https://web.archive.org/web/20010409063112/http://www.rsie.com/personal/compcomm.html

"The History of Social Networking (3-part video documentary)" | Rory Cellan-Jones | BBC
https://www.bbc.co.uk/programmes/b00y5jgf

Decentralize It! | Paul Ford | Aug 23, 2017 | Track Changes |
https://trackchanges.postlight.com/decentralize-it-2c9f0bc2ed8e

Bowling Alone - Robert Putnam - "described the reduction in all the forms of in-person social intercourse upon which Americans used to found, educate, and enrich the fabric of their social lives. He argues that this undermines the active civil engagement, which a strong democracy requires from its citizens."
https://en.wikipedia.org/wiki/Bowling_Alone

Appendix A - Early Social Utilities Included in Unix Systems

Note: one of the most common forms of feedback on this paper has been about the origins of various social commands cited. These things are facts, and everyone is understandably concerned making sure the facts are kept straight. Some different accounts of the history can be attributable to the fact that Unix was not the only time-sharing system in the 1970's and social computing was not unique to Unix. Many of the commands listed here were either borrowed from, or had analogies in, other time-sharing systems. I have attempted, to the best of my ability, to identify these commands' first appearance in Unix systems. If you know of better information than listed below, please reach out to me. This is not intended to be a definitive source of this information, but rather to provide general context to the early history of the Unix OS.

Command	**Description**	**Orig. Unix Release**
who	see a list of who else is logged in to the system and what they're doing -- a similar program, 'w' has been referred to as the command that gave birth to social networking [37]	Version 1 AT&T UNIX
mail	send messages to other users, originally on the same system, but later on other networked systems	Version 1 AT&T UNIX
write	like 'wall' but sends a message to a single other user rather than to all users.	Version 1 AT&T UNIX
mesg	turn on or off the ability of others to message you through 'write' or 'talk'	Version 1 AT&T UNIX
wall	a concatenation of 'w' (write) and 'all', prints a message on the terminal of all other logged in users. One of the more annoying of Unix commands.	Version 7 AT&T UNIX
talk	chat with another user, originally only on the same system, but (as of 1983) also users on other networked systems.	4.2 BSD
finger	look up information on other users, including their location, contact information, and any notes the user has left (in their .plan or .project files) about what they are up to (like an early blog)	first written in 1971; then appeared in BSD3.0

last	look up the last time your friends were online with 'last username'	3.0 BSD

Appendix B: List of Known Public Access Unix Systems

While researching the history of public access Unix systems, I came across hundreds of old systems, and many new ones. I am compiling a list of these systems and will continue to expand it and add detail based on input from people like you.

Two things to note about this list: (1) not all systems listed here label themselves as "public access Unix systems", but they are included if they seem to follow the spirit of the definition used in this article. And (2), some system names are included here based on vague references I've come across on the Internet, and these ones are often sorely needing additional input from well-informed readers like you.

If you know of public access Unix systems that are missing from this list. Please contact cmccabe@sdf.org to have them added.

For the time being, the list is being maintained in a Google Sheet (oh, the irony!). However, note that this Sheet does not contain all the references I have -- completing this list is phase II, after the main text of the article is stabilized.

You can view the list of systems here: https://docs.google.com/spreadsheets/d/1TlXEK_f0RIs14x6c5RVjEY9aYuqJOjvNywJ_p0TQ1Og/edit#gid=1099430226

Appendix C: Developers of Programs Cited in the tilde.town Section

Usernames for tilde.town are written with a leading ~ (tilde), as in: ~username. Note that this is just a small selection of the user-created programs on tilde.town.

Developer	**Program**	**Description**
~login	tilde-coin	A (non-crypto) currency that can be exchanged among users.
~krowbar	tilde-bot	An IRC bot that, among other things, provides distorted captcha questions that can earn users tilde-coins.

~ne1	holodeck	A multi-user virtual world of prose that continuously expands as users log in and describe more interconnected spaces.
~desvox	BB&J	A retro BBS that lets users asynchronously interact with each other..
~curiouser	botany	A curses-based app that lets you grow a plant from a seed (or let it die from neglect).
~troido	cadastre	A virtual homesteading environment in which users submit ascii-art pictures of their plot.
~troido	asciifarm	An in-development game letting you grow fields of crops.