What I Saw at the Evolution of Plan 9

Geoff Collyer
geoff@collyer.net

ABSTRACT

This is a sort of oral history of Plan 9 and related work at Bell Labs and my home. It is necessarily a subjective view; others may disagree. It is subject to my recall of events; dates may be off.

Dramatis Personae

These are the main people mentioned (perhaps implicitly) in this paper. Their roles were not as limited as they appear here. Those mentioned above the dividing line worked at one time at the Bell Labs Computing Science Research Center (at times Center 1127) or its successors.

login	name	roles (abbreviated)
`ken`	Ken Thompson	Unix, Plan 9 & C creator, compilers, filesystems
`dmr`	Dennis Ritchie	Unix & C creator
`rob`	Rob Pike	Unix developer, Blit, Plan 9 & Inferno creator
`rsc`	Russ Cox	rob's summer intern, Plan 9 developer
`philw`	Phil Winterbottom	Plan 9, Brazil, Inferno, Alef developer
`seanq`	Sean Quinlan	cached-worm inventor, venti creator
`sean`	Sean Dorward	Limbo compilers, venti creator
`presotto`	Dave Presotto	*upas* creator, Plan 9 networking & compilers
`brucee`	Bruce Ellis	Plan 9 & Inferno fan, compilers
`jmk`	Jim McKie	Plan 9 developer & maintainer
`geoff`	Geoff Collyer	Plan 9 user & maintainer; researcher, support
`forsyth`	Charles Forsyth	Plan 9 fan, compilers, Inferno maintainer
`miller`	Richard Miller	Plan 9 fan, compilers, Raspberry Pi port

Prehistory

Unix energised me rather intensely, starting in 1976 with Sixth Edition: [Tho1975] a powerful, coherent system with simple, regular interfaces, device access via special files, novel pipes, a hierarchical file system, relatively free of arbitrary limits and restrictions, written in a high-level system implementation language, with source and documentation on-line, with sophisticated tools like *yacc*, [Joh1975] and able to run on modest and popular hardware. I was an undergraduate at the University of Western Ontario in London, Ontario, Canada, and my point of contact was Eric Gisin at the University of Waterloo. I persuaded the UWO Computer Science department to obtain its first Unix licence, for a DEC PDP-11/34 [Cor1973]. Charles Forsyth, then one of the informal student administrators of Waterloo's PDP-11/45 Math/Unix, was a great help in upgrading our Sixth Edition system.

Contemporary interactive systems (e.g., DEC's TOPS-10, [Cor1977] BBN's TENEX, [Bob1972] Honeywell's GCOS) [Hon1980] provided less (and less interesting and elegant) service on considerably larger machines; Multics [Cor1965] required *much* larger machines, but was influential: ideas and some people moved from Multics to Unix.

I followed the progress of Research Unix through Tenth Edition [Com1990] and managed to run Sixth through Ninth Editions [Lab1986]. By contrast, the ongoing mauling of Unix by the Unix vendors in the 1980s was demoralising. Features were more important to them than general mechanisms [For1990]. POSIX then standardised even the mistakes [Eng1990].

I first heard of what became Plan 9 when interviewing with Rob Pike in 1986. He was recruiting others to assist with developing software for the *Gnot* terminal, [Loc1987] a successor to the Jerq, Blit, DMD 5620, and Teletype 630 and 730 bit-mapped displays but with a network interface instead of a serial port, and a 68020 CPU, and later an AT&T CRISP CPU [Dit1987]. Rob gave a talk on the Gnot software in 1987 at the University of Toronto, where I was on staff, and which I found encouraging in the period that System V Release 4, the Grand Unified Unix, was being created by smashing System V into 4BSD at high speed.

First Contact

I found the 1990 UKUUG Plan 9 papers [Pik1990] as electrifying as the 1974 CACM Unix paper [Rit1974]. Plan 9 seemed to be a continued evolution of Research Unix but rethought for modern hardware, with the benefit of 20 years' experience, and different where it made sense (e.g., per-process file namespaces, no filesystem links). Notably, it was designed for distribution, with specialised systems connected over a variety of networks, including Ethernet and Datakit [Che1980]. The initial terminals were Gnots.

I got the single-floppy bootable PC distribution at a Usenix conference, probably in 1991, and later the 3-floppy distribution. These provided a tantalising taste of the system, overcoming the ugliness of PC hardware.

Arrival at Murray Hill

In July 1994, I arrived at Murray Hill, NJ to work for AT&T under Ted Kowalski creating the operating system for a TV set-top box called Homecenter, borrowing bits from Plan 9 [Col1996, AMD1994]. The Second Edition of Plan 9 was issued in 1995 for a nominal fee but not as freely-available source, and supported systems such as the IBM PC and the Nextstation, based on the Motorola 68020. A few components of the system, such as user-mode IP processing, the *8½* window system, and *help* (now *acme*) were written in the *Alef* language, [Win1995] created by Phil Winterbottom, which focussed on concurrent programs and implemented CSP channels [Hoa1978].

In 1995, AT&T split, Homecenter was cancelled, I moved to Unix support in Research in the newly-created Lucent Technologies, and most of center 1127 (Computing Science Research) moved to Lucent.

Multiprocessor PCs and Other Systems

IBM PCs were initially resolutely uniprocessor systems. Companies such as Compaq began making multiprocessor PC systems out of Intel x86 processors starting with the 386, but the Pentium Pro was the first x86 processor really designed to be built into multiprocessors, particularly with more than two processors. The memory ordering model from the 486 on has been the same, but starting with the Pentium Pro, CPU pipelines and store buffers became long enough to cause Plan 9 *sleep/wakeup* problems due to queued memory writes [Cox].

Intel guarantees cache coherence [Pat2009] (though not DMA coherence) on x86 CPUs; the problem is worse on CPUs that require manual cache flushing (e.g., ARM, PowerPC). Decent RISC-V systems provide cache and DMA coherence, which is very helpful.

Brazil

A development branch, Brazil, was already underway, driven by Phil Winterbottom. The emphasis was on greater efficiency (e.g., streamlined queues instead of Multiprocessor Streams™), but also included a few experiments, such as raw uncompressed raster graphics (`/dev/graphics`) instead of Bitblt, ultimately replaced by *draw*(3). `/dev/graphics` assumed a 300 Mb/s network that did not materialise. Nevertheless, the first version of *rio*, distinct from *8½*, was written for raw raster graphics. Sean Quinlan created a *rio* variant called *brio* that implemented something closer to Bitblt or *draw*(3) on top of raster graphics, for better performance (e.g., when rubber-banding a new window). IP processing was moved into the kernel, partly with an eye to new 100 Mb/s Ethernet. Brazil eventually become main-line Plan 9 before Third Edition.

Inferno

Inferno was a response to the needs of set-top box developers, a Plan 9 kernel with user-mode processes replaced by an interpreter for Limbo byte codes [Dor1997a]. Inferno itself could run on bare hardware or hosted on another operating system. Limbo in turn was influenced by CSP and Alef, intended for concurrent programming [Tec1997, Dor1997b]. The *draw*(3) interface first appeared here.

In 1996, I moved to a Research department building a multi-media message system, named Mesa, and incorporating fax, mail, and voice input and output. Today Apple's Siri is similar. The system was built on Inferno, hosted on Plan 9 quad-core Xeon PCs (for ECC memory). I designed and implemented an intended replacement for SMTP, RSMTP [Col2001]. We saw a much bigger improvement in Ethernet performance from changing to switched Ethernet than from moving from 10 to 100 Mb/s Ethernet.

Phil Winterbottom lead the *Pathstar* project, a combined telephony and IP router, positioned as a possible 5ESS phone switch replacement. It was built on a Plan 9 kernel, but running Inferno. In 1999, my department moved to Pathstar development, in the hope that a Siri-like replacement for dial tone would be appealing. In 2000, Pathstar was cancelled; Phil and then Ken Thompson left Lucent, which was in decline, and then the 2001 'dot-com' crash happened.

Lucent's Inferno Business Unit could not figure out what to do with Inferno, after making a hash of its documentation. So Lucent sold it to Vitanuova, who made a serious attempt to sell it, including converting Framemaker documentation back to manual pages.

New 9P and File Servers ...

Alef was replaced by *libthread* in C. Third Edition was issued in 2000 under an open-source licence with *draw*(3) graphics.

A new version of 9P, the file server protocol, named *9P2000* was created and Ken's file server kernel was adapted to speak it. A new authentication mechanism was included in *9P2000*. A few system calls were changed in conjunction with *9P2000*. Sean Quinlan and Sean Dorward created *venti*, a hash-addressed block store with deduplication. Sean Quinlan, Jim McKie, and Russ Cox created a new, user-mode file server, named *fossil*, that could use *venti* as its block store.

... and Return to California

In 2001, Lucent offered to buy out senior researchers; Rob Pike and others left for Google. I left for a start-up in California. Fourth Edition was issued in 2002.

I wrote early gigabit Ethernet drivers and bought an optical-disc jukebox. I was not ready to rely on *venti* nor *fossil* yet, but wanted to be able to handle archives larger than 2 GB, so modified Ken's file server kernel to implement 64-bit file and device sizes [Tho2005]. Bruce Ellis had recently improved code generation for the 386 to in-line most `vlong` operations, which made their use cheaper. 9P had always used 64-bit file sizes, but the original file server only implemented signed 32-bit sizes.

I implemented optional greylisting [Har2003] in *smtpd*, which greatly reduced the volume of mail spam. The arms race continues, however.

In 2004, Dave Presotto left for Google.

Return to Murray Hill

In 2006, Alcatel bought Lucent and became Alcatel-Lucent. Jim McKie recruited me to return to Research, to the remains of center 1127. Russ Cox had been holding the fort, but was looking to spend more time in academia. With Russ's help, I got the latest *venti*, using *libthread*, to work and to survive restarts. Previously, shutting down and restarting this version of *venti* had destroyed its block store.

I then copied the contents of the optical-jukebox file servers into a *venti* store on a RAID 5 volume on magnetic disks. The history goes back to the end of February 1990. Making it generally available would be difficult since it contains home directories and private files, which would be exposed. At Russ's suggestion, I converted the file server kernel into a user-mode file server, *cwfs*(4), and then retired the file server kernel, and the IL protocol along with it, as the file server kernel was the last remaining software that required IL. NAT devices tended to mangle IL headers anyway.

Ports

Jim McKie had long maintained the Plan 9 PC support (other than the compiler suite). He had already created the 64-bit *9k* kernel for AMD64 systems for a U. S. Department of Energy High Performance Computing contract under the project name 'HARE'* with participants Bell Labs, IBM Research, Sandia National Labs and Vitanuova [Col2010a]. Charles Forsyth produced the C compiler suite for it. Ultimately, the project instead used IBM Blue Gene PowerPC supercomputers at various U. S. government

* Portions of this project supported by the Department of Energy under Award Number DE-FG02-08ER25847.

laboratories. *9k* initially only ran on AMD64, not Intel64, systems, and was intended as a CPU server for use with *drawterm* or other terminal. It is highly compatible at user source level with the old Plan 9 kernel, implementing the same system calls and many of the same devices.

I ported Plan 9 to machines of several 32-bit architectures: PowerPC (Xilinx Virtex 4 and 5 FPGAs), Arm (Trimslice and others), MIPS (Routerboard) [Col2010b]. The Virtex ports were in support of testing an encrypted memory system. For the others, we were looking for a small, portable terminal that could attach to Ethernet, mouse, keyboard, and monitor as needed. (Ultimately, the Raspberry Pi 4 and later probably come closest.)

mk of a kernel on a 24-core 386 Xeon system used so much system time (probably on lock contention) that by default it was slower than with NPROC=4. Since then, use of semaphores for locking and monitor/mwait instructions to idle, instead of looping, should have fixed that. Our quad-core Xeons built a kernel from scratch in 2 seconds elapsed time.

The PC bootstrap programs *9load* and *9pxeload* had various problems, including becoming too large to fit in 640KB due to additional Ethernet drivers, and the need for modified device drivers. I replaced them with the new *9load* and *9boot*(8), using a variant PC kernel and the normal PC kernel's drivers instead of modified drivers [Col2013]. The new bootstraps are compressed with *lzip*, are self-decompressing, and can load 386 and amd64 kernels. *9load* can load *gzip*ped kernels.

Better IPv6

I worked toward the goal of being able to run a Plan 9 site using only IPv6 [Dee1998, Dee2017]. I had this working at home in California and it should be possible now, even with SLAAC address assignment, except for PXE booting from BIOS. I updated IPv6 support to match newer RFCs and modified *dial*(2) to try to connect to all IP addresses for a name at once, since it is common for a system to have multiple IPv6 addresses, plus any IPv4 addresses, but they may not all be reachable and up at a given time. IPv6 uses multicast for its equivalent of ARP, so some Ethernet drivers had to be fixed to implement (mainly accept) multicast.

In Memoriam

In 2011, Dennis Ritchie died. Aside from all his work creating C and Unix, Dennis was a major supporter and booster of Plan 9. His good taste and judgement kept C small and simple, at least through 1990 ISO C (thereafter, obsessive-compulsives began adding their favourite, if largely pointless, hobby-horses).

Return to California II

In 2014, I moved to Google in Mountain View, CA. Alcatel-Lucent was still in decline and would soon be bought by Nokia. I bought 2 Lenovo TS140 Xeon servers, which took a while to tame. I never got the integrated Intel i217 Ethernet controller to function, but was able to add PCI-E Ethernet cards. I wrote an NVME driver for Plan 9. I retired at the end of 2018.

I wanted to explore 10Gb/s Ethernet, but Intel's X540 cards run hot, and require cooling by the TS140s' fans, which are very loud. In 2019, I bought quiet Puget Systems Xeon and PC Engines APU2 AMD64 servers with ECC memory, and adapted *9k* for them.

9k for AMD K10

9k as I had copied from the Labs was not really ready for serious use on current hardware. It got odd traps, had old drivers, and could only use about 600MB of RAM, despite being nominally a 64-bit system. I fixed the traps, updated the drivers, made it run on Intel64 systems, and adapted the system to use many-gigabyte memories. I widened *malloc* and *memset* size arguments to `uintptr`, the unsigned integer type as wide as a pointer, and adjusted *fossil* and *venti* to exploit more than 4 GB of memory. `9k10cpu` has run on systems with 4, 16, 32 and 64 GB of RAM. Once the system was stable, some programs still got odd faults due to now-too-small thread stacks. I fixed the relevant libraries to use bigger thread stacks or allocate local data on the heap.

By using *lzip* to fit in the bottom 640KB and an uncompressing executable header inspired by Russ Cox, PXE can now load most PC kernels without *9boot*. PC hardware is sufficiently discoverable that `plan9.ini` can usually be dispensed with.

Future Times

The year 2038 is not far off and signed 32-bit times will go negative in that year. I converted the system to consistently use *ulong*s to hold times and adapted the `libc` functions and the few programs that compare times (e.g., *mk*, the *qer*(8) programs) to cast them to *vlong* first. Tests for failure now compare exactly with `−1UL`, not just any negative time. This extends times to the year 2106 without needing to use 64-bit times, and thus change disk formats, notably *venti* and *fossil*. I extended the `timezone` files for North America to 2106.

Escape from APE

I have tried to reduce reliance on APE, the ANSI/POSIX Emulation environment. My *mk* invokes *rc*, not *ape/sh*, so it is possible to build kernels without a working APE. Most other APE programs, such as *awk*, and *troff* and friends, were almost trivially converted to native programs. So far, only programs that use Unix's *select* require substantial changes. Only the `postscript` programs, *bzip2*, *gs*, *spin*, and *tex* still use APE.

In Memoriam II

In 2020, Jim McKie died. He had tackled the support in Plan 9 of the most obnoxious, recondite, and brain-damaged aspects of PC hardware, more-or-less cheerfully. Without his efforts and those of Russ Cox, I believe that Plan 9 support at the Labs would have collapsed after Dave Presotto left.

Licensing

In 2020, the Plan 9 Foundation, at `http://p9f.org`, was formed. I was one of the initial directors. In 2021, Nokia, the successor to Alcatel-Lucent, transferred ownership of Plan 9 to the foundation. The historical releases can be found there, relicensed under the MIT open-source license.

9k for RISC-V

In 2020, I ported *9k* to 64-bit RISC-V systems [Col2023]. Richard Miller supplied the C compiler suite [Mil2020]. RISC-V systems are becoming faster and larger, and some even resemble x86 PCs, with PCI-E slots. Some even have adequate programming documentation. So far, RISC-V is relatively simple and free of the historical mistakes made in other architectures, though there are vigorous efforts underway to add bad ideas and unnecessary complexity, usually just to win micro-benchmark contests. With luck, I may be able to replace my complicated Puget Xeon systems some day with simpler RISC-V servers.

Before RISC-V, ARM64 seemed like a potential x86 replacement, but a variety of inexpensive systems has not appeared (other than recent Raspberry Pis), and RISC-V is considerably simpler.

What I Saw Just a Lttle

There were various developments that I was aware of, but was not involved with much, notably AoE (ATA over Ethernet) and *Nix*, a modified *9k*. At Bell Labs, we used AoE to connect our main file server to a Coraid™ RAID device, which held our *venti* arenas and indices, and *fossil* file systems *main* and *other*.

What I Opposed

I think that Ethernet jumbo packets are a bad idea [Ste] and discouraged their use everywhere. Use of jumbo packets on some but not all devices on a network is awkward at the least, there is no agreement on maximum jumbo packet size, jumbos consume more memory than normal packets (obviously), and the IP checksums and Ethernet CRC are less effective as packet size increases. The Ethernet controllers will transmit continuously if there are packets to send, so increasing the maximum RPC size in `devmnt` (which I have done at home) is probably more effective at increasing throughput; I doubled MAXRPC (by negotiation) and saw 30% improvement on large copies.

Developments Elsewhere

Go seems like a worthy language [Tho]. I haven't spent much time with it to date, though.

Some people feel that Plan 9 needs a web browser capable of being used for banking and the like. The infeasibility of creating such a browser is discussed here, [DeV2020] but Plan 9 is a distributed system, so running a browser on a Unix machine using some combination of VNC and *u9fs* is feasible.

In related news, C++ continues to grow in size and complexity. It may be the largest programming language ever. IBM's 1965 PL/I specification was 164 pages, and PL/I was considered to be a huge and complex language. ISO's 2017 C++ specification is 1,605 pages. I'm certainly never going to understand it.

Acknowledgements

I have been lucky to observe or work with some brilliant people, and they were generally helpful and generous. Ken Thompson volunteered to help me set up a Plan 9 optical-jukebox file server.

David du Colombier has incorporated some of my personal changes into the *9legacy* distribution at `http://9legacy.org`.

Closing Thoughts

Plan 9 has not taken the world by storm, alas. UTF-8 has been a success, despite the silliness of recent Unicode additions (e.g., emojis) [Pik1993]. There have been by-products: `plan9ports`, ports or re-implementations of *sam*, *acme*, and *rc*. But some powerful parts have not been generally understood and adopted: e.g., the *cpu* command and underlying machinery, networks in the file namespace, remote access to devices via remote file system protocol, easy and routine cross-compilation, starting daemons as `none` and using cryptographic authentication to change user-id.

The evidence suggests that the popularity of an operating system (indeed, of most software) is inversely proportional to its technical excellence. Operating system choices are also somewhat religious: it is very difficult to change someone's choice. Apple managed it with Mac OS X, but there was moaning, wailing and rending of clothing from some old users for quite a while, despite the obvious superiority of OS X.

Perhaps in time, the lessons of Plan 9 will be broadly absorbed, though needless complexity and bad taste [Fou, Tor] will always oppose them, including in hardware design [Int]. For example, the ever-changing administrative interfaces and increasing access restrictions in MacOS. It took time for the lessons of Unix and predecessor systems (e.g., not writing in assembler, process protection via memory management, user-space command interpreters, filters, pipelines, hierarchical file systems) to be absorbed, and then in some cases misunderstood (e.g., `/proc` in Linux). And of course, we must defeat *git*.

References and Further Reading

AMD1994. AMD, *Am29200 and Am29205 RISC Microcontrollers User's Manual*, Advanced Micro Devices, 1994.

Bob1972. D. G. Bobrow, J. D. Burchfiel, D. L. Murphy, R. S. Tomlinson, ''TENEX, a Paged Time Sharing System for the PDP-10,'' *Comm. Assoc. Comp. Mach.* **15**(3), pp. 135-143 (March 1972).

Che1980. G. L. Chesson, A. G. Fraser, ''Datakit Network Architecture,'' *Compcon*, pp. 59-61 (Spring 1980).

Col1996. Geoff Collyer, ''Setting Interrupt Priorities in Software via Interrupt Queueing,'' *Computing Systems* **9**(2), pp. 119-130 (Spring 1996).

Col2001. Geoff Collyer, *A Really Simple Mail Transfer Protocol*, `http://www.collyer.net/~geoff/rsmtp/rsmtp.pdf`, April 2001.

Col2010a. Geoff Collyer, Charles Forsyth, ''A Cache to Bash for 9P,'' *Fifth International Workshop on Plan 9*, Seattle (October 2010).

Col2010b. Geoff Collyer, ''Recent Plan 9 Work at Bell Labs,'' *Fifth International Workshop on Plan 9*, Seattle, invited talk (October 2010).

Col2013. Geoff Collyer, ''Bootstrapping Plan 9 on PCs,'' in *Plan 9 Programmer's Manual* (December 2013).

Col2023. Geoff Collyer, ''Plan 9 on 64-bit RISC-V,'' *Ninth International Workshop on Plan 9*, Waterloo (April 2023).

Com1990. Computing Science Research Center, AT&T Bell Laboratories, Murray Hill, New Jersey, *UNIX Research System Programmer's Manual, Tenth Edition,* Saunders College Publishing (1990).

Cor1965. F. J. Corbató, V. A. Vyssotsky, ''Introduction and Overview of the Multics System,'' *Proceedings of the AFIPS Fall Joint Computer Conference*, Las Vegas, pp. 185–196, ACM, ISBN 978-1-4503-7885-7 (1965).

Cor1973. Digital Equipment Corporation, *pdp11/45 processor handbook*, 1973.

Cor1977. Digital Equipment Corporation, *decsystem10 monitor calls*, May 1977.

Cox. Russ Cox, *Spin & Plan 9*, `http://swtch.com/spin`.

Dee1998. Steve E. Deering, Bob Hinden, *RFC 2460: Internet Protocol, Version 6 (IPv6) Specification,* Internet Society (December 1998).

Dee2017. Steve E. Deering, Bob Hinden, *RFC 8200: Internet Protocol, Version 6 (IPv6) Specification,* IETF (July 2017).

DeV2020. Drew DeVault, *The reckless, infinite scope of web browsers*, `https://drewdevault.com/2020/03/18/Reckless-limitless-scope.html`, 18 March 2020.

Dit1987. D. R. Ditzel, H. R. McLellan, ''The CRISP Microprocessor: Collected Papers,'' Comp. Sci. Tech. Rep. No. 138, AT&T Bell Labs (February 10, 1987).

Dor1997a. S. M. Dorward, R. Pike, D. M. Ritchie, H. W. Trickey, P. Winterbottom, ''Inferno,'' *Proc. IEEE Computer Conference (COMPCON)*, San Jose, California (1997).

Dor1997b. S. M. Dorward, R. Pike, P. Winterbottom, ''Programming in Limbo,'' *Proc. IEEE Computer Conference (COMPCON)*, San Jose, California (1997).

Eng1990. Institute of Electrical and Electronics Engineers, *Portable Operating System Interface (POSIX), Part 1: System Application Program Interface (API) [C Language] (IEEE Std 1003.1-1990) = ISO/IEC 9945-1:1990,* IEEE, New York (1990).

For1990. C. H. Forsyth, ''More Taste: Less Greed? or, Sending UNIX to the Fat Farm,'' pp. 161-172 in *Proceedings of the Summer 1990 UKUUG Conference*, UKUUG, London (July 9–13, 1990). `http://www.terzarima.net/doc/taste.pdf`

Fou. Free Software Foundation, *GNU software,* anonymous ftp from `prep.ai.mit.edu:/pub/gnu`.

Har2003. Evan Harris, *The Next Step in the Spam Control War: Greylisting*, 2003.

Hoa1978. C. A. R. Hoare, ''Communicating sequential processes,'' *CACM* **21**(8), pp. 666–677 (1978).

Hon1980. Honeywell, *Level 66/Series 6000 General Comprehensive Operating Supervisor*, August 1980.

Int. Intel, *Intel 64 and IA-32 Architectures Software Developer Manuals*, `https://www.intel.com/content/www/us/en/developer/articles/technical/intel-sdm.html`.

Joh1975. S. C. Johnson, ''Yacc — Yet Another Compiler-Compiler,'' Comp. Sci. Tech. Rep. No. 32, Bell Laboratories, Murray Hill, New Jersey (July 1975).

Lab1986. AT&T Bell Laboratories, in *UNIX Programmer's Manual, Ninth Edition, Volume One*, ed. M. D. McIlroy, Murray Hill, New Jersey (September 1986).

Loc1987. Bart N. Locanthi, ''This is Gnot Hardware,'' 11276-870629-04TM, AT&T Bell Laboratories (1987).

Mil2020. Richard Miller, *A Plan 9 C Compiler for RISC-V RV32GC and RV64GC*, `https://ossg.bcs.org/wp-content/uploads/criscv64.pdf`, 19 Oct 2020.

Pat2009. David Patterson, John Hennessy, *Computer Organization and Design (4th ed.),* Morgan Kaufmann (2009).

Pik1990. Rob Pike, Dave Presotto, Ken Thompson, Howard Trickey, ‘‘Plan 9 from Bell Labs,’’ *Proc. of the Summer 1990 UKUUG Conf., London*, pp. 1-9 (July, 1990).

Pik1993. Rob Pike, Ken Thompson, ‘‘Hello World,’’ pp. 43-50 in *USENIX Technical Conference Proceedings*, USENIX, San Diego, CA (Winter 1993).

Rit1974. D. M. Ritchie, K. Thompson, ‘‘The UNIX Time-Sharing System,’’ *CACM* **17**(7), pp. 365-375 (July 1974).

Ste. Richard A. Steenbergen, *The Case Against Jumbo Frames*, https://-archive.nanog.org/sites/default/files/wednesday_general_steenbergen-_antijumbo.pdf.

Tec1997. Lucent Technologies, ‘‘The Limbo Language Definition,’’ in *Inferno User’s Guide* (1997).

Tho1975. K. Thompson, D. M. Ritchie, *UNIX Programmer’s Manual,* Bell Laboratories (May 1975). Sixth Edition

Tho2005. Ken Thompson, Geoff Collyer, ‘‘The 64-bit Standalone Plan 9 File Server,’’ in *Plan 9 Programmer’s Manual* (2005).

Tho. Ken Thompson, Rob Pike, Russ Cox et al., *The Go Programming Language*, http://go.dev.

Tor. Linus Torvalds, *Linux*, http://www.linux.org.

Win1995. Phil Winterbottom, ‘‘Alef Language Reference Manual,’’ in *Plan 9 Programmer’s Manual, 2nd ed.* (March, 1995).