

Lessons Learned From Sendmail

(The Good, The Bad, The Ugly)

Eric Allman, UC Berkeley (again)
EuroBSDcon, Lillehammer Norway, September 2019

Disclaimers

Please be kind — I wrote this talk at 6:00am today.

Nothing specifically BSD, although that was the first platform.

- Altos, Amdahl UTS, Apollo DomainOS, AIX[2345], AUX[34], *BSD, BSD/OS, ConvexOS, Cray CSOS, Darwin, Dell, DG/UX, DYNIX, Encore UMAX V, HP-UX[89], Intergraph CLIX, Interix, IRIX[456], ISC, Linux (many distros), Mach, MPE-iX, NCR MP-RAS, NEC EWS-UX/V, NEXTSTEP, SCO OpenServer, SCO UNIX, Omron LUNA/UNIOS, OSF/1, Pyramid DC/OSx, QNX, RISC/os, Solaris, Sony NEWS-OS, SunOS, System V, Tandem NonStop-UX, TitanOS, Tru64, Ultrix, UNICOS, UnixWare, etc., etc.

I haven't worked on sendmail in years now — these are from memory.

Background

Originated on Unix 6th Edition on PDP-11 (INGRES RDBMS, ARPAnet connection).

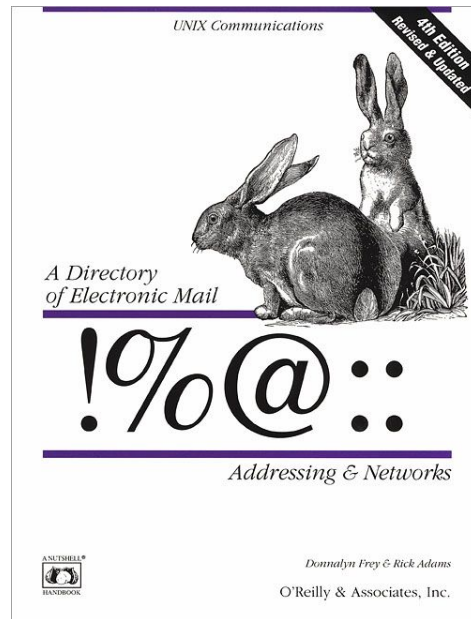
Pre-Internet: ARPAnet, Berknet, UUCP, CSnet, PurdueNET, DECnet, ... very ad hoc, highly heterogeneous.

First instance: delivermail — routing only, no header rewriting, no queue, no state, no DSNs, compiled config.

→ `ucbvax:research!decvax!wrl::foo@berkeley`

Side note: this was the origin of `<sysexits.h>` and `syslog`.

Also: `mpx` files (mentioned by Warner yesterday) used to implement `syslog` (no named pipes, sockets, etc.).



4.2BSD, CSRG, and the Internet

WNJ talks me into doing the MTA (on the condition that I change the name) ... how hard can it be? (answer: way harder than expected)

Required persistent state, queue, long-lived daemon, header rewriting, delivery status notifications, ... the list goes on and on.

➤ Mistake #1: try to maintain PDP-11 compatibility

Top priority: reliable delivery. Hell hath no fury like a professor whose grant proposal was “lost in the mail”.

Oh yeah: sendmail was not my job — nights & weekends project.

A Brief History

Wrote sendmail when I was at Berkeley (INGRES Project student and staff) → v5.

Went to a startup (Britton-Lee) and walked away.

Sendmail splintered, every vendor different, plus IDA, KJS, and several others.

I returned to Berkeley (Mammoth Project), attempt to centralize infrastructure; in the process of modifying sendmail to support eric@cs.berkeley.edu (instead of eric@mammoth.cs.berkeley.edu) I started merging versions → v8.

Left Berkeley, had disastrous job, licked wounds, then co-founded Sendmail, Inc.

Left after 13 years, eventually landed back at Berkeley, where I am now.

About the Standards...

Rudimentary at best, mostly still in development, often ambiguous, changed (in many cases literally) overnight.

- Success: rewriting rules as a concept. Allowed substantial changes trivially. Sometimes it's easier to solve a general problem than a specific problem.
- Mistake: syntax. Fine when config files were a few dozen lines, terrible when they went to hundreds of lines. More about this later.

DRUMS (Detailed Revision and Update of Messaging Standards) Working Group stirred the pot again several years later (v8 era).

IETF process (at least around messaging standards) is deeply flawed IMHO.

Rewriting Rules

Mistakes were made.

- Tabs. Stole the idea from make. SIF later told me it was the biggest mistake he made in make, even worse than `$<`. Live and learn.
 - Lesson: don't blindly follow the pack.
- Obscure syntax (which only got more obscure with time).
 - `R$+@$=w` `$1@$D`
 - Single letter macro and option names (ultimately got fixed, but painful).
- Looked at the config file, horrified, but decided to keep it because it was already running on (“at least”) a dozen machines.

About Back Compatibility: Support would be so easy without the users

- Mistake: letting my installed base cripple me.
 - Compare with C: Incompatible changes v4 → v5, again v5 → v6, yet again v6 → v7. Made for a much cleaner language.

A dozen sites? A hundred? Think tens of millions.

I should have been willing to change more things, notably defaults.

But beware of blindly following the trendy new thing: legacy matters, shiny things corrode with time, ...

... Oooohhh squirrel!!!

Starting a Company

I started Sendmail, Inc. in an attempt to get back to coding (other than maintenance).

- Glad I did — I learned a lot.
 - ◆ Sendmail was one of the very first “mixed” open/proprietary source companies.
 - ◆ Quite the roller-coaster ride, through the heady days of the Internet Bubble and then the not-so-heady days of having that bubble burst (but we survived).
 - ◆ At peak ~200 employees, offices all over US, London, Paris, Tokyo, briefly in Beijing.
- But complete failure on my goal — pretty much didn’t code for a decade.
- Tied my hands regarding technical decisions — companies talk up innovation but fear risk.

I’ve got a whole talk on “so you want to start a company...”

Random Other Observations (Code)

- Using m4 (1st attempt: procedural): Mistake.
- Using m4 (2nd attempt: declarative): Mixed success.
- Trying to make m4 output “pretty”: Clear mistake — it’s assembly code, not intended for human consumption.
- All functions in one binary: Mistake.
- “Wrapping” interfaces to buffer against portability issues: Clear success.
- Forking to do memory management: Mixed. Threading didn’t exist at the time, fork-exec expensive, IPC weak (no sockets, named pipes, etc.)

Random Other Observations (Design)

- Leverage everything you can: Success.
- Steal from the best: sendmail 8 took ideas from IDA, KJS, SunOS.
 - But mostly generalized them — e.g., “maps” (databases) taken from IDA, but instead of dbm only, allowed for arbitrary types.
- Encourage outside innovation.
 - Milter (mail filter) interface allowed lots of good work (and by the way, allowed for threaded implementations).
- Don't let yourself get too caught in the past.

Questions?