

Daniel Lublin – Software developer & factotum

contact: d at surname dot se

Skills & knowledge

I have extensive, *hands-on experience* of software development and all sorts of information systems. As a *generalist*, I'm comfortable in diverse environments, and of a pragmatic *problem solving* orientation.

Social and communicative, and well-experienced in working together with new people. Code is often written in cooperation, and should in any case be written so it can be cared for collectively. Prefer simple, readable solutions which are easier to maintain and more reliable.

Moving freely between systems and technologies to get things done, I am not afraid to try something new, to experiment and tinker. Iterate until success. Always *learning* and *sharing* the knowledge.

I have extensive experience of most things Linux; the operating system, development platform, and server.

My knowledge spans all the way from development in various languages, onto server administration and deployment. Using all sorts of related tools and technologies, for efficiency, correctness and working productively as a team.

Service integrations, using APIs over suitable channels, or talking IPC. Metrics, data collection, processing, and database storage. Using version control, code-review and issue-tracking systems. Automated build-systems and integrations, testing.

Buzzwords for that:

- Go, Python, C, shell scripting, are most frequent these days
- gdb, strace, tcpdump, wireshark, git bisect, sed, awk, ...
- OpenEmbedded, Bitbake build system
- Postgresql, Mysql, Sqlite, and less sql as well
- Git, and Gerrit, Gitlab, Gitfoo... it's been a while, Subversion
- Jenkins, Gitlab CI
- Ansible, Vagrant
- Docker, QEMU, Virtualbox
- Debian, Ubuntu, Arch Linux, ...
- FreeBSD and others are familiar

I like to see development as a craft. As a practitioner of this craft, I know the importance of being familiar with my tools. I do get a lot of things done on the command-line, which leans itself well to automation.

Talkative section

I do development around, not limited to: text-, network- and web-oriented systems, standardized and own protocols and APIs, databases, automation, data processing, system integrations. Both higher and lower levels of the stack. Adaptation, integration and new solutions – often based on common, free software.

I have much experience of networking, Internet services and infrastructure, and I'm familiar with various protocols and server implementations. Both professionally, and from running services

privately, hacking on code, reading RFCs. Buzz: TCP/IP, IPv6, iproute2, iptables, arp, DNS & nameservers, HTTP(S), SMTP, DHCP, IRC, XMPP. SSH, OpenVPN, SSL/TLS certificates, CAs, OpenPGP, ...

I write text that is version-controlled and readable, often in Markdown. I enjoy Pandoc for converting to other formats – even built an artist portfolio using Pandoc templates some shell script (oh and the Photoswipe js-lib, and too much CSS tweaking): samanehroghani.com¹

Coming to think of it, I have recently become more interested in visualization, graphing and such. I'm looking at Prometheus for metrics collection, and Grafana for visualizing of the same. With some friends I'm helping out to run a mesh-network [Pjodd.se](http://pjodd.se)² that provides free WLAN Internet-access on the streets of Malmö. I built a small tool using D3.js to graph this mesh³, which has been very useful when building the network.

Mapping and GIS is also fun, and I recently dabbled with some Javascript, using Leaflet to map a sailing journey⁴ I took part in a few years ago. The former also involved some code in Go to dig out photo metadata from Digikam's Sqlite database.

In Go I have also had fun writing [twet](https://github.com/quite/twet)⁵, which is an interface to the decentralized, minimalist micro-blogging service called [twtxt](https://github.com/quite/twtxt) (participants tend to write their own clients, in their language of choice!). I find myself productive in this language, and will happily use it instead of C for various systems and problem domains.

I'm spending a little time on exploring Rust. There is a lot to learn about the language, but there is a lot of good documentation, and the compiler is very friendly. So far, I've used it to write a few lines of code⁶ to display the time on a 7-segment display hooked up over I2C to an old Raspberry Pi. I also tried using it to solve some of 2018's Advent of Code challenges⁷, before time got eaten.

Natural languages:

- English, Swedish – fluent written and spoken
- French, German, Spanish, Turkish – from beginner to basic conversational

Assignments and experience

Volvo Cars, Lund

November 2017–July 2018 (through Stickybit AB)

- Developing benchmarking tools in POSIX C to measure system capabilities and performance, like scheduling accuracy of short-interval timers. Visualization of results using Python Pandas, Seaborn.
- Setting up a continuous integration chain using GitLab CI for a handful of projects, involving code in C, Elixir/Erlang, Python and Rust. Heavy use of Docker; target platforms ARM and x86, storage in Artifactory. Later doing similar work using Gerrit and Jenkins.

¹<https://samanehroghani.com>

²<https://pjodd.se>

³<http://graph.pjodd.se>

⁴<https://lublin.se/frejus-marmaris/>

⁵<https://github.com/quite/twet>

⁶<https://github.com/quite/klockan>

⁷<https://github.com/quite/advent-of-code-2018>

Axis Communications, Lund

September 2016–August 2017 (through Stickybit AB)

- As part of company-wide Tools team, working on maintenance of a large build system based on OpenEmbedded, targeting misc devices and hardware platforms. Occasional⁸ upstream⁹ contributions¹⁰.
- Developing an extendable linting tool in Python for OpenEmbedded package recipes, triggered in Jenkins when developers upload relevant patchsets to Gerrit, and posting back feedback on syntax, template adherence, etc.
- Holding workshops on OpenEmbedded, bitbake, and particularly devtool for developers; working on solving their frustrations with this fairly new in-house environment.
- Prototyping and evaluating the use of QEMU for running test-suites in the CI chain.

Beijer Electronics, Malmö

June–August 2016 (through Stickybit AB)

- Doing bring-up of U-Boot & Linux on custom ARM-based board. Reading schematics, doing RAM configuration, pin-muxing and tailoring of the device tree and driver associations. Including reading up parameters from ROM and customizations. Distribution building using Yocto/OpenEmbedded.

Netnod Internet Exchange, Stockholm

April–November 2015

- Extending an invoicing system to support new DNS services, with formula-based invoice generation using customer data aggregated over time. From the Django systems' web and CLI front-ends, over the ORM, down to the Postgresql and Mysql databases.
- Built a test client to simulate customers, speaking the REST-API being developed for new services. Small tools for generating test zone-files. Minor extensions, endpoints, to the Django system.
- Working in a small, remote team with tight communication, daily stand-ups online.

During 2003–2011 I undertook, among other things, several longer journeys. Large parts of these by hitchhiking, but that's way too many stories for a cv.

I was previously a C++-hacker and general factotum for an online RPG at Mindark in Göteborg. Before that I was hacking at early web-server and CMS company Roxen, Linköping. I began running Slackware as my primary operating system back when Linux was at 2.0.x.

⁸<http://git.openembedded.org/bitbake/commit/?id=1679188f9c55c615cae780f2b5e6852dea9cf2ec>

⁹<http://git.openembedded.org/openembedded-core/commit/?id=db54c9a22a9b66c673df8e8336de5e47fc9edda0b>

¹⁰<http://git.openembedded.org/openembedded-core/commit/?id=404e8e3661469175e1ea087ebfaf3a7867bf4df2>