Mikołaj Bartosz Lenczewski https://lenczewski.org
Email: mikolaj@lenczewski.org Mobile: +44 7512 734 132 https://git.lenczewski.org

Education

University of Manchester Manchester, UK
Computer Science, B.Sc. (Hons.) | 1st Class with Honours *Oct. 2020 – July 2023*

- Modules included: System Architecture, Advanced Distributed Systems, Algorithms and Data Structures, and Microcontrollers
- Dissertation: "A Generic Framework for Distributed Computing", covering compilers and distributed systems

Experience

Arm Ltd. Cambridge, UK
Engineer III | C++, C, Architecture Specification, ACPI, Linux Kernel *Jan. 2025 – Current*

- Researched and helped design armv9 architectural changes with focus on cache behavior, and TLB invalidations
- Prototyped extensions to ACPI firmware to support new architectural features
- Upstreamed Linux kernel patches leveraging contiguous page hints, reducing TLB pressure and improving memory access latency
- Refactored legacy C++ perf. model generator to modern C++20, reducing allocations and improving parsing throughput
- Working on RDMA-based networking to reduce E2E latency and increase throughput for multi-instance LLM partitioning

Engineer II | C++, C, DynamoRIO, SystemVerilog *Sept. 2023 – Jan. 2025*

- Modelled cache performance using DynamoRIO and C++, analyzing cache misses, and instruction-level behavior
- Presented cache performance modeling results at internal company conference
- Designed, implemented, and verified functional gzip decompression engine components for FPGAs using SystemVerilog
- Implemented Linux memory management patches improving TLBI performance and correctness in high-pressure allocation paths

Engineer I | C++, Make, Batch, CMake, LLVM, Zephyr *July 2021 – Mar. 2023*

- Part of the PTUG Social Committee, organising in-person, online, and hybrid events for other PTUGs
- Ported the Zephyr build system to support the LLVM and Armclang toolchains
- Ported existing Make and Batch build scripts to use CMake
- Participated in daily standups, bi-weekly sprint reviews, and code review

Projects

Networking Projects using io_uring | C++, C, Linux Kernel, io_uring, Parsing, Networking *Sept. 2024 – Current*

- Learning io_uring for throughput-focused networking components, to minimize syscalls, copies, and scheduling overhead
- Implemented an efficient, as-close-to-zero-copy-as-possible http 1.1 server
- Wrote an HTTPS client for the Riot API using OpenSSL with non-copying in-memory BIOs
- Implemented hand-written parsers (JSON, HTTP, custom binary) optimized to avoid allocations and branch misprediction
- Implementing an efficient, zero-copy, low-latency web crawler using C++, io_uring, and a custom TLS library
- Re-writing common data structures (trie, rbtree, spsc/mpmc queue) for lock-free, cacheline-aligned, zero-allocation usage

A Generic Framework for Distributed Computing | C, Compilers, Networking, Make *Sept. 2022 – July 2023*

- Designed a custom compute kernel language influenced by C11 and OpenCL
- Wrote a bytecode compiler and interpreter for the aforementioned compute kernel language
- Designed a custom network protocol, and implemented various distributed algorithms (e.g. hash-table, memory, synchronisation)
- Wrote a mandelbrot set compute kernel as an example of the platform

RoboSoc Orchestra Team | C++, C, Embedded, ARM *Sept. 2022 – July 2023*

- Served on the Conductor team, architecting and implementing the control system for the orchestra
- Planned the protocol and interfaces for communicating between the conductor and different musicians
- Implemented a MIDI parser, the conductor user interface, and the conductor backend

Skills

Languages: C++ (C++17/20), C, SystemVerilog, Bash, Python
Systems: Linux kernel, io_uring, RDMA, Zephyr, LLVM toolchains
Focus: lock-free programming, cacheline alignment, memory ordering, NUMA, profiling (perf), CPU microarchitecture

Interests

I have quite a few ongoing personal projects: a 3D sandbox game using Vulkan, a custom linux distribution, a C raytracer, a simple aarch64 assembly kernel, and more. In the past, I have built and programmed maze-solving robots in FORTH. I want to work on more embedded projects (from-scratch drones, 3d printers, CNC machines).