Minseo Kim

minseo@mskim.org • kimminss0 • blog.mskim.org

Education

B.S. in Computer Science and Engineering, POSTECH, South Korea — Feb 2022 – Present

Skills

Programming Languages	C, C++, Python, Go, Java, TypeScript
Systems	Linux, FreeBSD, GDB
Infrastructure	Docker, Docker Compose
Web	React.js, Next.js, Express.js
Databases	PostgreSQL
Natural Languages	Korean (Native), English

Job Experience

Wordbricks LLC — Jun 2025 – Aug 2025

Intern Full-stack Engineer

- Led and developed an LLM-based web crawling API and dashboard, owning the full product lifecycle including backend architecture and frontend implementation
- Skills: TypeScript, Next.js, Hono, Vercel AI SDK, Cloudflare Workers, Prisma, Supabase

POSTECH Data Systems Lab — Jan 2023 – Jan 2024

Undergraduate Researcher (Full-stack Web Developer)

- Developed a website serving as an integrated database for PVLDB, an academic conference
- Full-time during vacation; part-time during semester
- Key Contributions:
  - Data engineering for data integration and automation of data pipelines
  - Full-stack development, including client-side, server-side, and database design/optimization
- Skills: JavaScript, React.js, Express.js, PostgreSQL, Airflow, Docker, Docker Compose

Projects

Pintos — Sep 2025 – Dec 2025

Educational operating system for 80x86 architecture (based on Stanford CS140)

- Written in C and x86 assembly
- Key Implementations:
  - Kernel-level threading with priority scheduling and synchronization primitives (locks, semaphores)
  - System call handling for user program execution, including process management and file operations
  - Virtual memory system featuring demand paging, stack growth, memory mapped files

RISC-V CPU — Mar 2025 – May 2025

5-stage pipelined RISC-V CPU implementing a subset of the RV32I ISA

- Collaborated in a 2-person team
- Implemented in Verilog HDL
- Key Implementations:
  - Hazard detection unit with data forwarding

  - 2-Bit saturation counter branch predictor with pattern history table and globally shared history buffer (gshare)
  - Direct-mapped cache

Tiny ML (TML) Compiler

May 2025

Implemented a compiler backend that translates TML programs into machine code for a virtual machine (Mach)

- Functional programming language based on Standard ML (SML)
- Compiler written in OCaml
- Supports closures, higher-order and recursive functions, recursive datatypes and pattern matching

Sponge

Mar 2025 – May 2025

Educational network stack development project (based on Stanford CS144)

- Written in C++
- Key Implementations:
  - TCP flow control
  - Routing table lookup & packet forwarding
  - ARP & IP packet processing

BaroKey

Nov 2023

Submitted to the 3rd UniDThon Hackathon (Team Project)

- Developed a web service that provides users with real-time emergency keywords, such as safety threats or natural disasters, based on their location
- Role: Team Leader, Full-stack Developer (Led Backend & Infrastructure)