Ahmad Rashid

647-212-2141 | ahmad.rashid@mail.utoronto.ca | linkedin.com/in/ahmadhrashid | github.com/ahmadhrashid

Education

University of Toronto, Mississauga — Mississauga, ON
Honours Bachelor of Science in Computer Science — *Sep. 2023 – Present*

Technical Projects

Redis Clone | *Go, Networking, Concurrency, Systems* | `github.com/ahmadhrashid/go-redis`

- Implemented a Redis-compatible in-memory KV server supporting **27** core commands, lists, streams, blocking reads, and transaction queuing; interoperable with `redis-cli`.
- Built a goroutine-per-connection TCP server with per-connection wait/wakeup queues and notifier channels to guarantee correct blocking semantics and avoid races/deadlocks.
- Benchmarked under **100** concurrent clients: **p50 = 0.7 ms**, **p95 = 0.7 ms**, throughput ≈ **350k ops/sec**.
- Added replication primitives and designed synchronization to ensure correctness under concurrent writes.

mysh – Unix Shell | *C, POSIX, Systems Programming, Networking* | `github.com/ahmadhrashid/mysh`

- Implemented a POSIX-style Unix shell in C supporting pipelines, background jobs, environment-variable expansion, builtin utilities, and optional TCP chat client/server integration.
- Built pipeline execution with dynamic pipes and I/O redirection, builtin commands, and a job-management table for background processes.
- Ensured robust job control and signal handling to avoid zombie processes; implemented I/O multiplexing so the shell could handle both user input and socket activity for an integrated chat feature.
- Produced a reliable shell that runs complex foreground/background pipelines correctly, prevents zombie processes via proper reaping, and supports concurrent socket-based communication.

Experience

Mathematics Instructor — February 2025 – Present
Mathnasium - The Math Learning Center — *Brampton, ON*

- Taught advanced math using problem-solving methods that parallel algorithmic thinking.
- Diagnosed learning gaps and tailored solutions, applying structured, data-driven approaches to improve performance.
- Translated complex concepts into clear explanations, strengthening technical communication and mentorship skills
- Reinforced logical reasoning and systematic problem-solving, directly applicable to software design.

Skills and Concepts

CSC263: Data Structures and Analysis | *Python* | *University of Toronto Mississauga*

- Implemented and optimized abstract data types (ADTs) such as graphs for network modeling, dictionaries for fast key-value lookups, priority queues for scheduling, and disjoint sets for union-find operations.
- Designed, implemented, and compared various data structures such as AVL trees, hashmaps, heaps, and forests.
- Conducted worst-case, average-case, and amortized complexity evaluations to improve computational efficiency.
- Enhanced problem-solving skills by designing algorithms tailored to specific use cases, optimizing memory usage, and improving runtime performance.

JavaScript Development | *Self-Taught (Eloquent JavaScript 4th Edition book)*

- Gained experience in JavaScript through OOP, higher-order functions, and scope management.
- Developed proficiency in asynchronous programming using promises and async/await, enabling efficient handling of background tasks and network requests.
- Applied functional programming techniques to create maintainable and scalable code.
- Developed an understanding of event-driven programming and DOM manipulation, enabling responsive and interactive web development.

Technical Skills

Languages: Go, C, Bash, JavaScript, Python, Java, Risc-V Assembly, HTML, CSS
Systems & Networking: TCP/IP, Sockets, Concurrency, Goroutines, POSIX APIs, Process Control, I/O Multiplexing
Developer Tools: Git, Linux, Docker, VS Code, GCC, Make, PyCharm, JGrasp, Figma
Concepts: Distributed Systems, Replication, Transactions, Synchronization, Streams, Pipelines, Job Control