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 Programming*

- Built a Redis-compatible in-memory key–value server in Go with a custom RESP encoder/decoder and a goroutine-based TCP server to serve concurrent clients.
- Implemented core Redis APIs, list operations, blocking reads, streams and transaction queuing.
- Solved concurrency and blocking-read correctness by designing per-connection goroutines, wait/wakeup queues and notifier channels, plus synchronization using Go concurrency primitives to prevent races.
- Delivered an interoperable Redis-like server with low-latency command paths and replication primitives.

mysh — Unix Shell | *C, POSIX, Systems Programming, Networking*

- 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.

Core Programming Knowledge

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, hashing techniques, heaps, and disjoint forests.
- Conducted worst-case, average-case, and amortized complexity evaluations to improve computational efficiency and scalability.
- 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 object oriented programming, 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 such as reduce, filter, and map 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