Yen-Ju Tseng

LinkedIn | 858-729-3110 | Personal Website | tyj850916@gmail.com | GitHub

Skills

- C++ | C | Kotlin | Java | Go | JavaScript | HTML | CSS | Python
- Android Studio | Git | GitHub | Zookeeper | Kafka | MongoDB(NoSQL) | SQLite(SQL) | Visual Studio | VSCode | IntelliJ IDEA
- Android Development | Computer Networking | Distributed Systems | Frontend | Backend | OOP

Projects

MySQL-like Relational Database System in C++17 — 04/2022 - 06/2022

- Developed a MySQL-like relational database in **C++17**, proficient in interpreting, manipulating, querying, and presenting table data, delivering flawless performance while seamlessly managing 15,000+ data entries.
- Achieved a reduction in code maintenance efforts due to the **MVC** architectural pattern, leading to increased development efficiency.
- Reduced parsing errors through proficient scanning, tokenizing, and parsing techniques, ensuring more accurate query processing.
- Implemented the **chain-of-responsibility** design pattern to efficiently process user-provided commands.
- Employed the **factory** design pattern to seamlessly handle statements.
- Implemented data **encoding** to files and **decoding** from files to enable **persistent storage** and efficient data management.
- Reduced average query execution time by 20% through the implementation of **indexes** and **LRU Cache**, resulting in faster data retrieval.

Chat Android App (Client/Server) (Kotlin, Clean Architecture, Koin, Ktor, Hilt, MongoDB) — 06/2023 - 07/2023

- Engineered a robust, scalable chat server with **Ktor**, **Koin**, and **MongoDB**, harnessing the power of the **KMongo** toolkit.
- Managed 10+ concurrent connections through the implementation of a robust **WebSocket-based** communication system.
- Reduced code complexity by 30% via dependency injection with **Koin** and **Hilt**, enhancing code maintainability and scalability.
- Employs a customizable routing mechanism for precise request handling.
- Developed a robust **Android** chat app with **MVVM** and **Clean Architecture**, boasting a declarative UI with **Jetpack Compose**.
- Integrated **ViewModel** with **Flow**, ensuring real-time updates to the UI and enhancing user engagement.
- Ensured a secure and efficient connection between Android chat client and server using **HttpClient** and **WebSocketSession**.
- Achieved server-client interoperability, optimizing data exchange efficiency through serialization techniques with **Ktor**.

Fault-tolerance Scalable Cloud-Based File Storage service (Go, SQLite, API, Backend, gRPC) — 02/2023 - 04/2023

- Engineered a robust, Dropbox-inspired, fault-tolerant cloud-based file storage solution with both client and server components.
- Supported simultaneous access for 10+ users, effectively managing 100+ files.
- Achieved a reduction in data retrieval time by implementing file segmentation and **SHA-256 hashing** for faster access for users.
- Reduced update conflicts by 99% by implementing efficient **versioning** and hash list strategies, ensuring smoother synchronization.
- Enabled seamless data exchange, improved interoperability, and enhanced scalability by leveraging **protocol buffers** for **gRPC**.
- Optimized user experience by streamlining synchronization with an index.db file in the client's base directory.
- Achieved efficient block storage and server scalability by implementing a mapping approach based on **consistent hashing**.
- Ensured server reliability with **fault tolerance** mechanisms based on the **RAFT distributed consensus protocol**.

Simple Router in C — 02/2023 - 03/2023

- Constructed a streamlined router capable of receiving raw Ethernet frames and efficiently handling various packet types, including **ARP requests**, **ARP replies**, **ARP caching**, **ICMP** (returning messages to the sending host), **switching**, **longest prefix matching**, **IP sanity check** (ensuring minimum length and checksum), and other vital IP forwarding functionalities.
- Implemented **ping** and **traceroute** operations, and enabled file downloads using HTTP from designated application servers.
- Implemented **Trie-based Longest Prefix Match**, achieving a 90% improvement over the brute force method for 1000+ IPv4 addresses.

Sliding Window Protocol in C — 01/2023 - 02/2023

- Implemented communication between two or more hosts using a **sliding window protocol** (Window size = 8) that employed **selective repeat/retransmission** and cumulative ACK to guarantee reliable in-order delivery of frames between hosts. Each sender could only communicate with one receiver at a time, while a receiver must be able to handle frames from multiple senders concurrently.
- Established reliable communication through the segmentation of messages exceeding MAX_FRAME_SIZE (i.e., 64 bytes) into frames.
- Achieved a 99% accuracy rate in message reconstruction, ensuring the reliable retrieval of original messages.
- Ensured data integrity and effective communication between senders and receivers through the implementation of **CRC-8** error detection.

Nachos Operating System Implementation in Java — 09/2022 - 11/2022

- Executed the development of the Alarm class, implementing waitUntil, timerInterrupt, and cancel, as well as KThread.join.
- Employed interrupt disable and restore techniques to ensure atomicity while implementing condition variables.
- Implemented essential file system calls, including create, open, read, write, close, unlink, exec, join, exit, and halt.

Education

Master of Science — **University of California San Diego** — *San Diego, CA, USA* — **09/2021 - 06/2023**

- Major in Electrical and Computer Engineering (GPA: 3.5/4.0)
- Coursework: Software Foundations, Operating Systems, Computer Networks, Graduate Networked System, Advanced Data Structure

Bachelor of Science — **National Taipei University** — *Taipei, Taiwan* — **09/2015 - 06/2019**

- Major in Communication Engineering (GPA: 3.46/4.0)
- Coursework: Data Structure, Advanced Computer Programming, Database System