Qingqian(Oliver) Wang

Email: qw116@duke.edu
LinkedIn:Qingqian Wang | LinkedIn

Education

Duke University Durham, NC
Master of Engineering in Electrical and Computer Engineering Aug 2022 - May 2024
Software Development Engineering Track

- GPA: 4.0/4.0
- Courses: Data structure & Algorithm in c/c++; Fundamentals of Computer Systems and Engineering; Systems Programming & Engineering; Software Engineering; Computer security; Computer network

China University of Mining and Technology(CUMT) Jiangsu, China
Bachelor of Engineering in Civil Engineering Sep 2018 - Jul 2022

- GPA: 3.78/4.0(Rank: 11/135. Top 4%)
- Honor: China National Scholarship, Ministry of Education of the People's Republic of China (Top 1%); First-Class Outstanding Student Scholarship, CUMT (Top 5%)

Professional Experience

Didi May 2023 - Aug 2023
Software Development Engineer Intern

- Overview: Deeply engaged in the development of foundational software for Didi's intelligent vehicle project, including the development of fault/configuration management modules and contribution to the development of heterogeneous communication modules.
- Built module for parsing, storing, and transmitting fault/config data streams using SQLite, ensuring security through serialization and parity checks. Addressed SQLite concurrency with permission management.
- Supported heterogeneous communication across Linux, Windows, QNX systems. Led Linux thread management using a publish-subscribe model, epoll, notifier, and job queue.

Project Experience

Malloc Library, Dynamic Memory Allocation Tool Jan 2023 - Feb 2023

- Implemented malloc library with **first-fit** and **best-fit** methods using system calls sbrk()
- Organized memory space using **implicit linked list**.
- Optimized first-fit and best-fit by maintaining **explicit linked list**, accelerating original speed by 420%

B+ tree/ Buffer pool design Jun 2023 – Jul 2023

- Overview: This project involved the creation of a B+ tree and the application of an **LRU-K** buffer pool. Program correctness was verified using **Gtest**.
- Implemented a B+ tree data structure supporting insertion, deletion, and retrieval operations. Utilized read-write locks as latches to ensure **concurrent safety**.
- Employed an **extendable hash** structure as the index for the buffer pool, facilitating page_id to frame_id queries.
- Employed a **dual-queue approach** to manually implement the principles of LRU-K frame replacement.

RISC Game Design (Java Full-stack) Mar 2023 -May 2023

- Overview: Led a team in utilizing the Jira platform for Scrum **Agile** development. The project involved designing a territory-conquering game using the **MVC** (Model-View-Controller) architecture. The front end was implemented using **JavaFX**, while **SQLite** was employed for data storage.
- Utilized sockets for communication between clients and servers. Players could request game room creation from the server and await the entry of other players.
- In the game, players employed soldiers to attack the territories of other players. The gameplay included basic interactions like soldier **movement and upgrades.** Additionally, players could engage in optional activities such as upgrading technology, training spies, and forming alliances.
- After the game concluded, scores were calculated and stored in a database (db) file. Users could **log back** in using their account credentials.

Technical Skills

- **Language:** C/C++, Java, Python, Verilog, Shell(Bash), HTML, Linux, MySQL, SQLite
- **Tools:** VSCode, Vim, Emacs, IntelliJ IDEA, CLion, AWS
- **Collaboration & Development:** GitHub, GitLab, Agile(Jira), CI/CD, DevOps
- **Security:** Kali Linux, Nmap, Wireshark, Hashcat, John the Ripper