(315) 278-8449
Syracuse, NY
wangxiaojobhunting@gmail.com

Wang Xiao

Software Developer

GitHub: WangXiaoShawn
Linked In: linkedin.com/in/xiao-w-582a88229

Education

Syracuse University college of Engineering & Computer Science **May 2023(expected)**
MS of Computer Science *Current GPA:3.813*
Bentley University & McCallum Graduate School of Business **May 2015**
MS of Marketing Analytics & Business Analytics certificate *GPA:3.59*

Key Skills

Programming Languages C++/C, Python, Haskell,Golang, Sql
Applications/Technologies Git,System Programming in Linux environment,Multi-Thread Programming,Functional Programming

Course-Works Computer Architecture(A),Operation System(A-),Multi-Thread Programming(A),System Programming (A)

Professional Experience

Position: Software Analyst **2015 Sep—2021 May**
SHAANXI HAINA ELECTRONIC TECHNOLOGY CO., LTD *China*

- Designed and developed a full stack dashboard application based on MVC to for data visualization of customer data analysis to provide data-driven reference for upper management.
- Created the front-end interface using React.js for the dashboard.
- Built back-end service using Node.js, Express.js framework, along with MongoDB as a database for RESTful API service.
- Collected customer behavior, applied time series and logistic regression, decision tree to predict customer purchase rate, allocated more company resources to high-potential customers and lowered operating costs.

Technical Experience

Remote System Process Manager

- This project involves the creation of a server-client model for remote system process management in a Linux network environment. UDP-Sockets are used to facilitate communication between the server and client.
- Server:
  - Create multiple processes for running programs provided by the client.
  - Monitor the processes statutes through multiple threads and detects any abnormal signals for each Process in real-time.
  - If an abnormal signal is detected, the information is sent back to the client.
- Client:
  - Control the program running statutes (Running, Waiting, Terminated) on the server.
  - Receive real-time information about the process signals.

File Share Platform

- Server:
  - Capable of parsing client requests to the Linux command, then storing and reading files from server port.
  - Deploy epoll to handle high levels of concurrency. (High-performance server)
- Browser:
  - Login to the File Share Platform with ID and passwords
  - Allows for viewing every level of the directory and Supports extraction, deletion, and storage of files.

Implement a simple Operation System

- Build A Tiny-shell for Linux
- Scheduling Process with Round-Robin
- Implement System Calls and Address Translation
- Implement System Calls for Multi-process
- Implement System Calls for File I/O

Mini-banking System

- Combining the OOD idea, use c++ to write a bank system. In addition to basic banking system operations(withdraw, transfer money between accounts, saving, etc.), this system also automatically generates customer IDs, monitors account status, and prevents illegal operations.
- The system is deployed to a local file system, which is automatically read every time the bank system runs. In addition, all account changes are automatically written to the database when the process is terminated.