Raymond Chang

raymondkenchang@gmail.com rkchang linkedin.com/in/rkhchang raymondchang.ca

Skills

- Programming Languages: Python, C++, C, Java, SQL, TypeScript
- Technologies: Git, SQLite, GNU Make, Spring, CMake, PostgreSQL, Flask, GDB, VIM, Node.js
- Selected course work: Design & Analysis of Algorithms, Performance Engineering, Information Retrieval, Compiler Construction, Operating Systems, Databases, Artificial Intelligence

Education

University of Waterloo 09/2022 to Now

Masters of Mathematics in Computer Science (Thesis) (94.67%)

Teaching Assistant for: Databases, Operating Systems, Algorithms, Software Analysis and Testing

University of Ottawa 09/2016 to 04/2021

Honours Bachelor of Science in Computer Science (CO-OP) (8.35/10) - Deans Honours List

Honours project - Modelling and verifying distributed leader election algorithms with TLA+.

Work Experience

Research Assistant Waterloo, Canada — 09/2022 to Now

University of Waterloo

- Software analysis and testing research

CoreOS Software Development Student Ottawa, Canada — 06/2020 to 08/2020

Blackberry QNX

- Developed a binary analysis tool in Python to scan and detect issues in the QNX kernel
- Wrote Unit tests in C for the QNX Neutrino kernel and hypervisor, boosting system reliability
- Used Git, GNU Make, Ghidra

CoreOS Software Development Student Ottawa, Canada — 09/2019 to 12/2019

Blackberry QNX

- Worked on a Python program to test and track new commits to Review Board thus streamlining the code review process
- Setup a QNX Hypervisor System with an Ubuntu guest to demonstrate its abilities to other teams
- Wrote Unit tests in C for the QNX Neutrino kernel and hypervisor, boosting system reliability
- Used Git, GNU Make

Telematics Control Unit Software Developer Ottawa, Canada — 01/2019 to 04/2019

Ford Motor Company

- Developed features and Unit tests for C and C++ multi-threaded Linux applications
- Fixed bugs and race conditions/deadlocks found by static analysis tools such as Thread Sanitizer and Clang Static Analyzer
- Used GNU Make, Git, Jenkins and the Google repo tool

Junior Programmer Ottawa, Canada — 05/2018 to 08/2018

Canadian Border Services Agency

- Worked in a team of three to create a web application for the management of software development using Java, JSP, Hibernate, jQuery, SQL and Spring MVC
- Developed functional UI Mock-ups for the web application using HTML, CSS and JavaScript
- Gathered requirements for the web application and participated in weekly progress meetings
- Improved Unit test coverage of a seperate reporting application
- Acquired the Enhanced Reliability Status level of security clearance

Selected Projects

Juice: Java Compiler - Course project for Compiler Construction 01/2023 to 04/2023

- Developed a Java compiler targeting x86 with two teammates
- Consisted of 28,753 lines of TypeScript, 628 Jest unit tests and passed the majority of test cases
- Ported and upgraded an existing Intermediate Representation (IR) interpreter from Java to TypeScript to improve the debugging process
- Wrote hundreds of Jest unit tests with final code coverage of 80%

Lettuce - New Programming Language and LLVM based compiler 07/2023 to 08/2023

- Worked with a partner to create a new programming language and corresponding compiler using the LLVM compiler infrastructure
- Developed compiler in C++ and added an interpreter mode
- Created a new IR using MLIR
- Source code: https://github.com/rkchang/mlidk

DNS Server 12/2021 to Now

- Wrote a DNS server with C++ and the ASIO networking library
- Used Bazel for the build system and dependency management
- Wrote Unit tests with GoogleTest
- Source code: https://github.com/rkchang/rcdns

Document Search Engine - Course project for Information Retrieval 01/2020 to 04/2020

- Worked with a partner to create a search engine and UI for searching through a collection of course listings and news articles by using the Flask web framework
- Source code: https://github.com/rkchang/search-engine, Final report available upon request

Open Source Contributions

The LLVM Compiler Infrastructure Project https://llvm.org/

- Wrote C++ to emit an error instead of crashing
- Wrote Unit test in C++ to test the new error handling code
- Pull request: https://github.com/llvm/llvm-project/pull/70233