Lirong Yuan

lirongy@andrew.cmu.edu lirongyuan.github.io/

RESEARCH INTERESTS

Operating Systems, Programming Languages / Compilers, Software Engineering.

EDUCATION

Carnegie Mellon University **2023 - present**
PhD Student
Purdue University **2010 - 2014**
Bachelor's Degree in Computer Science and Mathematics GPA: 3.88/4.0

WORK EXPERIENCE

Software Engineer, Google **Sep 2014 - Jul 2023**
C++ Toolchain Team **Dec 2019 - Jul 2023**

- Developed a state-of-the-art gdb stub that supports modern multi-threaded programs, module loading, fault detection, expression evaluation, and so on.
- Resolved technical blockers that required in-depth debugging at the intersection of multiple modules: LLVM, LLDB, GNU C Library, Quick Emulator (QEMU).

Search Ranking Team **Nov 2017 - Dec 2019**

- Analyzed ranking experiments, came up with and executed plans to train and deploy machine learning models that increase content quality of results by a launchable margin.
- Redesigned the topic map pipeline, which reduced running time by 75% (from 8 hours to 2 hours). This allowed faster quality iteration and stable output tables.

RESEARCH EXPERIENCE

Research Assistant, Dept. of Mathematics, Purdue University **May 2012 - Aug 2012**
Roots of polynomials with random integer coefficients.

AWARDS

NSF CSGrad4US Fellowship (an annual stipend of $34,000 for three years) **2022**
Patent, Display screen with an animated graphical user interface. **2017**

NOTABLE COURSE PROJECTS

Operating Systems - Graduate Level (CS 50300) **Spring 2014**
Modified the source for XINU Operating System to: extend the messaging interface to allow sending multiple messages to a process; extend the process table to suspend waiting processes.
Computer Security - Graduate Level (CS 52600) **Spring 2013**
Built a software in Java that could hide information, verify integrity, and do forensic analysis.
System Programming (CS 25200) **Spring 2012**
Built a shell interpreter like csh that could execute simple commands, redirect files, do the piping and wildcarding, ignore ctrl-c, and eliminate zombie processes.
Computer Architecture (CS 25000) **Spring 2012**
Constructed a compiler using Yacc and Lex for the Simple C language that could generate x86-64 assembly language that can be assembled to produce an executable file.

TECHNICAL SKILLS & INTERESTS

Languages: C++, C, Python, LaTeX, Swift, Objective C, Java, Go, R, HTML, and CSS.
Tools: QEMU, LLVM, LLDB, PyTorch, Xcode, Vim, Matlab, and Mathematica.