

Louis Jenkins

34 W Montgomery Ave, Ardmore, Pennsylvania 19003, USA
LouisJenkinsCS@hotmail.com • +1 (610) 931-1207

<https://www.linkedin.com/in/LouisJenkinsCS> • <http://github.com/LouisJenkinsCS> • <http://LouisJenkinsCS.github.io>

EDUCATION

Bloomsburg University of Pennsylvania, Bloomsburg, PA

Bachelor of Science (B.S.) in Computer Science

- Dean's List

May 2012 – May 2017
Spring 2014, Fall 2015, Fall 2016

RESEARCH EXPERIENCE

N.S.F SMART SPACES R.E.U., Lehigh University, PA

May 2016 – Jul 2016

- **Project:** Concurrent and Scalable Built-in Hash Table for the Go Programming Language
- **Advisor:** Michael F. Spear
- **Awards:**
 - Peer's Choice for Outstanding Project
 - Honorable Mention for Microsoft Research sponsored C.R.A's 2017 Outstanding Undergraduate Researchers
- **Publication:** L. Jenkins, and M. Spear, "Concurrent Map for Go", in Preparation
- **Summary:**
 - Designed and implemented a novel scalable lock-based concurrent map for Go's runtime and compiler.
 - Designed with compatibility with Go map syntax, supporting insert/lookup/removal and concurrent iteration.
 - Outperforms sequential map by up to 7x across diverse microbenchmarks, competitive against lock-free maps.

MISC. EXPERIENCE

INDEPENDENT STUDY - STUDENT CODE EVALUATION TOOL, Bloomsburg University Fall 2016

- Developed a free open source tool that helps automate the process of grading and leaving feedback for students.
- Supports templates and project-wide markups in 169 languages to leave feedback for the student.
- Presented application at a Mathematical and Digital Sciences department seminar meeting.

SKILLS

LANGUAGES

- **Proficient:** C, Java, Go
- **Familiar:** Haskell, C++, HTML/CSS/JavaScript

PERSONAL PROJECTS

MOLTAR-OS - HOBBY OPERATING SYSTEM

- Developing an operating system in C and Assembly for the 32-bit x86 architecture for academic purposes.
 - Implemented virtual memory, interrupts, basic VGA and keyboard driver, and uniprocessor multitasking.
 - Designed to take a higher-half approach to virtual memory with 4MB pages.
 - Building from the ground up with POSIX-compliance as a long-term goal.
- Planned to have process management, multiprocessor support, ELF Binary loading, and command-line interface.
 - Planned to run actual userspace programs in a command-line environment.

MINIMAL-JVM - BYTECODE INTERPRETER

- Developed a java virtual machine in Haskell to run very simple programs written for JVM for academic purposes.
 - Implemented support for load/store, push/pop, conditional jumps, and arithmetic instructions.
 - Loads a single '.class' files that adhere to a subset of the JVM 8 Specification.
 - Designed execution cycle to be as functionally pure as possible through monads.
- Planned to have dynamic classloading, SSA-conversion of bytecode, and full support for bytecode instructions.
 - Planned experiment to test if functionally pure code under a functionally pure JVM is as efficient as others.

ANDROID WINDOW MANAGER

- Created new window manager for Android, which allows the user to configure predefined Widgets
 - Allows user to drag, resize, minimize, maximize, and snap Widgets via touch.
 - Handles marshallng and unmarshallng to maintain state across multiple user sessions.
 - Widgets include a web browser, notepad, screen recorder, and Google maps.
- In-progress features include support to allow user creation of custom Widgets using a built-in WYSIWYG editor
 - Designing minimal interpreted language for user to create callbacks for certain actions

UTILITIES PACKAGE FOR THE C PROGRAMMING LANGUAGE

- Created a large collection of useful thread-safe data structures and threading abstractions
 - Includes Linked Lists, Blocking Queues, Hash Maps, and Lock-Free Queues and Stacks
 - Incorporated some support for generic iteration through opaque, data-structure specific callbacks
 - Developed threading abstractions, including a thread pool, events, and futures.
- Implemented memory reclamation tools for sane memory management outside of a Garbage Collected environment.
 - Developed Hazard Pointer implementation for lock-free data-structures and Reference Counting.