

## EDUCATION

---

|   |                                 |                                |
|---|---------------------------------|--------------------------------|
| <b>Seattle, WA</b>  | <b>University of Washington</b> | <b>Fall 2016 – Spring 2020</b> |
| 3.9 GPA (Phi Beta Kappa, Dean's List).  |                                 |                                |
| <ul style="list-style-type: none"><li>• B.S. in Computer Science (direct admission) and Mathematics (double major).</li><li>• <b>Upcoming Courses:</b> Real analysis; abstract algebra; topology.</li><li>• <b>Past Coursework:</b> Database Systems (grad); Computer Networks; Distributed Systems; Operating Systems; Theory of Computation; Compiler Construction; Computer Vision; Algorithms; FPGA Programming.</li><li>• <b>Interest:</b> Systems programming (databases, distributed systems, operating systems); C++.</li></ul> |                                 |                                |

## EXPERIENCE

---

|  |  |                                   |
|--|--|-----------------------------------|
| <b>Software Engineer Intern</b>  | <b>Microsoft – Azure Cosmos DB</b>         | <b>June 2019 – September 2019</b> |
| <ul style="list-style-type: none"><li>• Created and integrated a mutation proxy fuzzer system for Cosmos DB's Cassandra query processor.</li><li>• Integrated Azure Active Directory authentication to every Cosmos DB request sent and demoed feature to Fortune 500 customers.</li></ul>   |  |                                   |
| <b>Data Scientist Intern</b>   | <b>Microsoft – Surface</b>                 | <b>June 2018 – September 2018</b> |
| <ul style="list-style-type: none"><li>• Implemented a real-time statistical process control system for Microsoft devices, processing 200 gigabytes of data per day, with Azure and .NET tools – the first usage of real-time analytics within Microsoft devices manufacturing.</li><li>• Used to improve quality, avoid excess costs, and find root causes during quality failures significantly faster.</li></ul> |  |                                   |
| <b>Chair</b>   | <b>Association for Computing Machinery</b> | <b>September 2016 – June 2019</b> |
| <ul style="list-style-type: none"><li>• Elected to be the external face of ACM and represent over 1,200 CSE students.</li><li>• Coordinated and planned events with the school and industry affiliates.</li></ul>  |  |                                   |
| <b>Teaching Assistant</b>  | <b>University of Washington</b>            | <b>March 2017 – August 2017</b>   |
| <ul style="list-style-type: none"><li>• Head Grader for Software Design and Implementation (CSE 331).</li><li>• Taught a section of 20-25 students and answered content-related questions on forums.</li><li>• Graded theory-based code reasoning and project-based assignments.</li><li>• Held office hours for homework help and course questions.</li></ul>   |  |                                   |

## PROJECTS

---

- **Torgo** (June 2019): Anonymous overlay network based on the Tor protocol. Routes traffic from a browser through a randomized circuit of Tor routers before sending it to the web server. Other Tor routers are found using a peer discovery registration service. Can run (for multiple days or longer) in a heterogeneous environment without resource leaks or deadlock. Wrote around 1.5k lines of code in Golang.
- **Distributed Database System** (June 2019): A linearizable, Paxos replicated, sharded key-value store with multi-key updates and dynamic load balancing, similar in functionality to Amazon's DynamoDB or Google's Spanner.
- **Operating System** (March 2019): Created a working operating system in C that can run multiple processes efficiently and store file data reliably. Based on the Experimental Kernel (XK).
- **Java to x86-64 Compiler** (June 2018): Uses JFlex (lexical analyzer generator) and CUP (LALR parser generator) to generate a scanner and parser using context-free grammars, then transforms the program into an AST for static semantics checking, type checking, and symbol table generation. Finally, generates runnable optimized x86-64 code.
- **SimpleDB** (March 2018): A relational database management system in Java that handles queries (joins, aggregate functions, selections, etc.), ACID transactions, and a steal/no-force crash recovery (with a write-ahead redo/undo log + non-quiescent checkpoints). It can run in parallel or as a distributed system across multiple machines.

## RESEARCH EXPERIENCE

---

|   |  |                                  |
|---|--|----------------------------------|
| <b>Undergraduate Assistant</b>  | <b>UW Database Group</b>                       | <b>Spring 2018</b>               |
| <ul style="list-style-type: none"><li>• Developing a cost model for LightDB, a database system for virtual and augmented reality content at scale.</li></ul>        |  |                                  |
| <b>Undergraduate Assistant</b>  | <b>Taskar Center for Accessible Technology</b> | <b>Autumn 2016 – Winter 2017</b> |
| <ul style="list-style-type: none"><li>• Worked with Dr. Anat Caspi and Nick Bolton to Developed a tutorial module for the OpenSidewalks Project in Unity.</li></ul> |  |                                  |