Kevin R Kelly

☎ (267) 644-9370 | ✉ kevink2019@gmail.com | kevink2022.github.io/**blog** | in /in/**kevink2019** | /**kevink2022**

Education

Pennsylvania State University August 2019 – December 2022
Bachelor of Science in Computer Engineering *University Park, PA*

- GPA: **3.85**/4

Experience

FAST Enterprises July 2023 – Present
Implementation Consultant *Sacramento, CA*

- Effectively communicate with clients to understand their business requests, discuss requests to ensure they address root business problems, and created tailored solutions within the FAST system, adhering to **object oriented** and internal best practices.
- Develop and refactor **SQL** queries with high standards of readability, maintainability, and performance. Ensure SQL query performance through Execution Plan and cardinality statistics analysis.
- Deliver confident and impactful client demonstrations and technical presentations, effectively communicating complex technical concepts to audiences of varying levels of technical expertise.
- Initiate discussions about and tirelessly advocate for stronger software oriented organizational principles. Examples include adopting style guides for the languages we use, adding unit and integration testing to our testing procedures, and refactoring for future maintainability while making changes.

RTD Embedded Technologies, Inc. May 2022 – August 2022
Engineering Intern *University Park, PA*

- Maintained and updated Legacy **Linux Kernel Drivers**, adeptly diagnosing issues and analyzing Linux Kernel source code to resolve bugs and implement driver updates.
- Developed a **Python** tool for automated driver retrieval and testing, effectively identifying and addressing widespread bugs resulting from Kernel updates.
- Diligently adhered to the comprehensive document control protocols required by **ISO 9001** and **AS9100** standards.
- Successfully ported legacy ROM DOS manufacturing tests to Python scripts for Linux, modernizing the PCB production testing process.

Projects

Boomic Music | *Swift, SwiftUI, Combine, async/await* January 2024 – Present

- iOS music playback app for high resolution audio codecs (e.g. FLAC) and music library manager.
- Automatically populates the music library through the iOS Files app, allowing for drag and drop. Organizes music into albums and artists based on metadata and displays associated album art.
- Built in **Swift** with **SwiftUI**, utilizing **persistent data structures** and **functional programming principles** while still taking advantage of SwiftUI's powerful observation libraries. Uses **Combine queues** for efficient, decoupled, and reactive communication.
- Utilizes a transaction based persistence system, allowing for users to roll back erroneous changes.
- Thorough **unit testing**, often following **test driven design**, writing tests before the code.
- Architecture and design decisions discussed in detail on my blog at kevink2022.github.io/blog/tags/boomic/

Channels | *C, GDB, Valgrind, pthread, semaphores* January 2022 – May 2022

- Multi-thread safe buffered channel system enabling threads to send or receive data.
- Includes a 'select' function enabling threads to submit multiple send/receive requests, where only the first available request is completed.
- Implemented in **C**. Employed linked lists for queue management, **mutex locks** for thread safety, and **semaphores** to control access to resources (no polling used).
- Utilized **GDB** and **Valgrind** for debugging and memory safety.