張博堯 (Chang, Po-Yao)

Xinzhuang District, New Taipei City, Taiwan

☎ (+886) 917-899-495 ✉ poyaoc97@gmail.com linkedin.com/in/poyaoc97 github.com/poyaoc97

Experience

MediaTek Hsinchu, Taiwan
Software Engineer *August 2022 – Present*

- Halved the size of a data structure in Neuron, a **compiler for machine learning models**, part of MediaTek NeuroPilot SDK, and put it on the heap, which fixed random app crash induced by stack overflow.
- Optimized jump tables emitted by the Android Runtime (ART) in an **ARM binary translator**. Emulated the *execve* system call in a way that extended the support to wider range of apps. Mitigated leaky abstractions detected by translated apps by altering system calls' arguments and their return value, among other things. Developed dynamic launch of the zygote service when opening an app. Baked the technology into the in-house debugging flow and helped customers land it on Android 13 and 14.

Education

National Tsing Hua University *Hsinchu, Taiwan*
M.S. Computer Science; Advisor: 李政崑教授 (Professor Jenq-Kuen Lee) *July 2020 – July 2022*

National Central University *Taoyuan, Taiwan*
B.S. Computer Science and Information Engineering *September 2016 – June 2020*

Skills

Languages: C++ (most interested), Rust, C, Python, English (TOEIC 945, 指考 93.5), Mandarin Chinese (Native)
Tools: LLDB, GDB, Git, CMake, Linux, clangd, Clang-Tidy, Clang-Format, GoogleTest, AddressSanitizer, UBSan

C++ Open Source Contribution

https://github.com/llvm/llvm-project/commits?author=poyaoc97

LLVM Clang (a compiler) *June 2023*

- Implemented the resolution for the C++ defect report, CWG 2521 (https://wg21.link/cwg2521), related to the stricter requirement on literal suffix identifiers and the deprecation of a grammar production of *literal-operator-id*. The patch under review: https://reviews.llvm.org/D152632.

LLVM Clang *June 2023*

- Implemented the resolution for the C++ defect report: CWG1473 by not erring on the lack of whitespaces after `operator""`, which is confusing as the error contradicts to the deprecation in C++. The patch under review: https://reviews.llvm.org/D153156.

LLVM Clang *April 2022*

- Implemented the resolution for the C++ defect report: CWG 1394 so that Clang doesn't mistakenly reject well-formed programs, also in ISO C++ §9.5.1 [dcl.fct.def.general]p2. The commit: 50b1faf5c188.

LLVM Clang *March 2022*

- Clang, the widely-used compiler, crashed when handling an *ill-formed* class member's *default member initializer*. The name of the *declaration* (*FieldDecl* in Clang AST) is empty, triggering an assertion failure.
- Fixed by properly setting the identifier part of the *declarator* to its *unqualified-id*. Now the name clash of a class member and the *injected-class-name* can also be correctly diagnosed. The commit: 355f1c75aa66. Details: https://poyaoc97.github.io/posts/2022-03-24-clang/

LLVM LLDB (a debugger) *December 2021*

- Resolved an issue making dynamic function call symbols in LLDB's disassembly unreadable by fixing the bool semantics of a class dating back to Apple's open-sourcing LLDB in 2010. The commit: 633b002944b9. Details: https://poyaoc97.github.io/posts/2022-01-07-lldb/

LLVM libunwind *December 2021*

- Added install targets to install the LLVM libunwind headers. The commit: 1c4867e6fc50.

Blog Posts

- **ABI Incompatibility of libc++ and libstdc++ and C++ Inline Namespace** is on things I learned after running into linker errors a while back, mainly how C++11 *inline namespace* affects symbol names and hence ABI.
- **C++ Trivia;** delves into the latest ISO C++ working draft to find out a semicolon's grammar entities (plural).