

Nashe Mncube

mncubenashe@gmail.com · +44 7552230755 · nasherm.github.io

Warwick, UK

EXPERIENCE

- **Arm**

Software Engineer - UK

Worked on the Arm Forge suite of tools for debugging HPC applications.

- Was part of the development team working on providing support for Python debugging functionality in Forge.
- Was part of the development team delivering support for CUDA-11 on Arm hardware in Forge.
- Was responsible for delivering support for the NVIDIA HPC compiler v20.9 in Forge.

Open Source Contributor to LLVM

- Was part of the development team working on support for Arm's scalable-vector-extension (SVE2) hardware feature in LLVM.
- Made general improvements and bug-fixes to LLVM and it's suite of tools. and technical skills in open source compiler implementation and development.

September 2020 Onwards

EDUCATION

- **MEng Computer Science - 2:1**

University of Bristol

My final year dissertation, graded as 1st class, involved working on the compiler for a research programming language called EasyCrypt. I extended the type system to support type-classes, a concept from type theory. This involved rigorous theoretical work as well as practical application. [nasherm.github.io/./diss]
2020

SKILLS

- **Technologies**

C, C++17, Haskell, LLVM, Rust, Python, x86-64 assembly, aarch64 assembly

- **Patterns & Practices & Tools**

Linux OS Internals, Compiler Development, Modern Computer Architecture, Reverse Engineering, High-Performance Computing, Object Oriented Programming, Functional Programming, CI & CD, Jenkins, Ansible, Artifactory, Git

- **Project Management**

Agile, Scrum, Jira

PROJECTS

- **Plox & Rox** [Lox Box]

A two fold project in writing compilers. Plox is a compiler in the style of an interpreter for the Lox programming language, written in Python, and Rox is a compiler, written in Rust, based on the bytecode virtual machine approach to compiler design.

Python, Rust

- **ossify** [ossify]

An open source operating system built in Rust targeting aarch64. architectures.

Rust, aarch64 assembly