Damien Ruscoe

Senior C++ Software Engineer

Bristol, United Kingdom 🗺
damien.ruscoe@gmail.com @
(+44) 7473-493-513 ☎
linkedin.com/in/damien-ruscoe
github.com/damienruscoe

A passionate and technically rigorous Software Engineer with over 18 years of industry experience, specializing in complex software development challenges including low latency systems, compiler technologies, and embedded development. Expert in C++ with complementary Python expertise, I thrive when tackling intricate technical problems that demand innovative solutions. Currently seeking opportunities to apply my extensive experience to high-performance applications within the financial sector.
A dedicated technology enthusiast who remains actively engaged with the developer community through regular participation in professional conferences and meet-ups, including the ACCU C++ developer conference. Driven by intellectual challenge rather than routine, I bring a combination of deep technical knowledge and unwavering curiosity to every project.

Core Skillset

- **Advanced C++ Expertise** Expert in utilizing C++ with advanced utilization of STL data structures and algorithms.
- **Optimization Strategies** Thorough comprehension of C++ optimization strategies, enabling the creation of high-performance software solutions with optimal runtime efficiency.
- **Compiler Optimization Proficiency** Skilled in maximizing runtime performance through comprehensive knowledge of compiler optimizations and the understanding of the hardware memory model.
- **Network Protocol Proficiency** Experienced in working with a diverse range of network protocols, encompassing fundamental transmission protocols to application layer protocols.
- **Template Metaprogramming Expertise** Deep understanding and accomplished in the application of C++ templates for generic programming and also employing template metaprogramming techniques for compile-time computations.
- **Code Clarity and Maintainability** Committed to code clarity and maintainability, consistently applying Clean Code and SOLID principles, along with appropriate design patterns to facilitate long-term maintainability and scalability.
- **Linux Expertise Extensive** familiarity with Linux as the preferred OS, both personally and professionally, leading to a profound understanding of its intricacies.
- **GCC Toolchain Utilization** Proficient in utilizing the GCC toolchain, with a particular emphasis on leveraging GDB as the debugger of choice.
- **Continuous Learning and Integration** Maintains an avid interest in modern C++ advancements, integrating new features introduced in language standards to enhance software development practices.

Career

Qube Research & Technologies

https://qube-rt.com/

Sept-Nov 2024

Joined Qube Research, $28B AUM across 5 funds, to spearhead the modernization of their critical trading infrastructure. Led the architectural transition from web-based interfaces to high-performance native desktop applications using wxWidgets, leveraging C++20 coroutines to eliminate threading complications and establish a robust foundation for next-generation trading tools.

Responsibilities

- Architected thread-safe GUI framework using C++20 coroutines: Eliminated recurring threading synchronization crashes that plagued 6+ developers weekly by wrapping 30-40 critical RTT API calls in coroutine-based abstractions, enabling intuitive sequential code patterns while maintaining thread safety
- Designed scalable market data visualization system: Built backend pipeline integrating RTT APIs with live price streams, implementing virtual scrolling and on-the-fly filtering/sorting for real-time instrument pricing across multiple markets
- Led technical mentorship of development team: Served as wxWidgets technical lead, mentoring 6 developers directly and consulting for broader engineering organization, transitioning team from callback-based async patterns to coroutine-based solutions
- Delivered proof-of-concept trading applications: Developed specialized desktop tools including settlement date management with regulatory compliance visualization, drag-and-drop pivot table for live financial data, JSON editor for quant validation, and cross-platform instrument pricing dashboard
- Enabled cross-framework compatibility: Designed coroutine abstraction layer separating GUI logic from RTT API integration, facilitating future UI framework adoption and ensuring architectural flexibility

Technologies Used

Real-Time Trading Systems
Market Data APIs
Financial Instruments
C++ 20
STL
Python
C#
wxWidgets
ncurses

QA Systems ***(Formerly Information Processing Limited)***

https://qa-systems.com/

2018-2024

Joined QA Systems to enhance Cantata, a mission-critical C++ instrumentation product serving safety-critical industries (automotive, aerospace, medical, nuclear). Working with a 4M LOC legacy codebase built on EDG compiler technology, systematically modernized the toolchain to C++20 while resolving hundreds of complex template and instrumentation bugs. Developed custom testing infrastructure and debugging methodologies that increased personal productivity from 1-2 bug fixes per week to 3-4 fixes weekly, achieving a peak of 8 bug fixes in a single week.

Responsibilities

- Led comprehensive C++20 implementation: Analyzed entire C++20 specification and implemented support for concepts, ranges, coroutines, spaceship operator, constexpr/consteval/constinit, and advanced lambda captures, enabling compliance testing for modern C++ codebases
- Achieved 45% performance optimization: Identified and resolved critical build issue where release builds shipped without compiler optimizations (-O3), delivering immediate performance gains while establishing proper build practices
- Built high-performance testing infrastructure: Developed remote execution scripts reducing test cycles from 4-5 hours to 40 minutes, enabling rapid iteration across multiple platforms and compiler environments
- Established STL compatibility testing framework: Pioneered comprehensive STL template testing across multiple compiler environments, transforming previously “unsupported” functionality into fully validated capability for safety-critical applications
- Resolved hundreds of template bugs: Systematically debugged approximately 200 STL wrapping failures and template corner cases using advanced debugging techniques and creduce tooling in 20-year legacy codebase

Technologies Used

EDG (Edison Design Group) compiler technologies
C++11
C++20
Python
templates
C
concepts
GCC toolchain
MSVC toolchain
GDB
ranges
creduce
STL
Linux
git
constexpr

SN Systems *Sony PlayStation*

https://playstation.com/
https://uk.linkedin.com/company/sn-systems
2014-2018

In my capacity as a Software Engineer in Test, my principal responsibility revolved around ensuring that the Sony PlayStation toolchain being shipped met the desired professional standards of quality.

Responsibilities

- Working on the proprietary PlayStation platform toolchain, developing tests and internal tools.
- Spearheaded the validation of a compiler scheduling static analyser, later integrated into the LLVM compiler as LLVM-MCA, ensuring optimized performance of code generation.
- Development and automation of tests for software running on PlayStation development consumer gaming platforms.
- Developed a full automation library for the building, execution and play simulation of AAA games saving 400 developer hours per release cycle.

Technologies Used

C++ *Python* *Groovy* *Jenkins* *GCC toolchain* *Make* *Ninja* *Linux* *LLVM* *creduce* *clang* *Distributed Build Tools* *git* *VIM* *Subversion (SVN)*

Imagination Technologies *(Now MIPS Inc.)*

https://imaginationtech.com/
https://mips.com/
2006-2014

Joined as graduate developer on CodeScape, Imagination’s flagship debugging platform for heterogeneous SoC development, serving major semiconductor manufacturers debugging META, MIPS, and PowerVR processors. Advanced from junior developer to senior-level contributor within an 8-person team, mastering professional development practices including Test-Driven Development while contributing to a 6M+ line C++/Python codebase. Delivered critical features including comprehensive breakpoint systems, industry-leading VT100 terminal emulator, and automated GUI testing framework that necessitated hiring 2 additional test engineers.

Responsibilities

- Architected comprehensive breakpoint debugging system: Designed and implemented complete breakpoint infrastructure supporting hardware/software code breakpoints, data breakpoints, and advanced complex breakpoints (tuple, data watch, primed) across 20+ GUI regions including source, disassembly, memory, registers, and target tree views
- Developed industry-leading VT100 terminal emulator: Built near-complete VT100-compliant terminal capable of handling Linux sessions, movie streams, and complex ncurses applications (vim, btop), enabling SSH/Telnet connectivity and serving as reusable component for formatted text display across CodeScape
- Created automated GUI testing framework (wxTest): Pioneered comprehensive system testing infrastructure enabling programmatic control of entire debugging workflow from GUI through networking to hardware targets, generating hundreds of automated tests requiring 4+ hours execution time and leading to creation of 2 test engineer positions
- Built multi-threaded Find in Files system: Implemented threaded search engine with GUI synchronization supporting text/regex patterns, recursive directory scanning, find/replace operations with regex group support, and VT100-formatted results with clickable navigation to source locations
- Established technical mentorship and training programs: Trained 2 test engineers on wxTest framework development, mentored 3 graduate developers on core product features, and created feedback loops driving infrastructure improvements and library feature enhancements
- Contributed to multi-architecture debugging platform: Developed features across heterogeneous processor support including META, MIPS, PowerVR, and x86 architectures with real-time OS integration, simultaneous multi-core debugging, and hardware/emulator target connectivity
- Implemented cross-platform GUI components: Built configurable, dockable debugging regions using C++ backend with Python bindings, enabling drag-and-drop data manipulation between views and extensible plugin architecture for custom debugging displays

Technologies Used

C++ *Python* *wxWidgets* *Test-Driven Development (TDD)* *Threading Synchronization* *META Architecture* *MIPS Architecture* *ARM Architecture* *x86 Architecture* *PowerVR GPU* *Linux* *Real-Time Operating Systems (RTOS)* *ThreadX* *FreeRTOS* *MEOS* *VT100 Terminal Protocol* *Networking* *SSH/Telnet* *JTAG/EJTAG/cJTAG* *QEMU* *FPGA* *Embedded Systems* *Debug Probes* *Hardware Emulators* *GCC* *Git* *CVS* *Open Source Contributions*

Education

UMIST *University of Manchester Institute of Science and Technology*

https://manchester.ac.uk/
2002-2004

I chose to pursue a course in Mathematics with Software Engineering at university because I sought a more challenging academic experience that would keep me engaged. Having already acquired skills in computing and programming, I wanted to delve deeper into a field that would push my intellectual boundaries further. Mathematics, with its intricate problem-solving and analytical aspects, presented itself as the perfect fit.

Qualifications

BSc. Mathematics with Software Engineering

Eccles College

https://eccles.salfordcc.ac.uk/
2000-2002

Qualifications

Mathematics *Further Mathematics* *Computer Science* *Physics*