Rushil Patel

Email : rushil.patel20@imperial.ac.uk
LinkedIn: https://www.linkedin.com/in/rushil791/
Mobile : +44 7376 110058
Website: https://rushil.vercel.app
GitHub: https://github.com/r-ushil

Education

Imperial College London London, UK
Masters in Computer Science; (On track) 1st Class with Honours *Oct 2020 - July 2024*
Courses: Algorithms, Compilers, Operating Systems, Probability and Statistics, Software Engineering Design, Computer Architecture.

St Wilfrid's Catholic Comprehensive School Crawley, West Sussex
*A-Levels; A*A*A*A** *Sept 2018 - Aug 2020*
Courses: Maths, Computer Science, Physics, EPQ.

Technical Skills

Languages: Rust, C, Typescript, Java, Haskell, Python, HCL, Go
Tools: Kubernetes, Docker, GIT, JIRA, Terraform, GDB, Make

Experience

Global Payments Atlanta, GA
Enterprise Architecture Intern *July 2022 - Oct 2022*

- **Multi-Cluster Kubernetes Terraform Module**: Created a Terraform module that allowed for the deployment of multi-cluster GKE infrastructure in multi-regions for increased reliability. Used Anthos Service Mesh with a custom load balancer.
- **CI/CD as a Service - Using Go, Terraform and Kubernetes**: Worked on an internal CI/CDaaS project to allow for abstraction, ease of deployment and uniformity of CI/CD pipelines across the company. Primarily used Go for its concurrency and spun up Kubernetes clusters using a pub/sub-style listener.

CUB3 Inc London, UK
Contractor - Software Developer *June 2022 - July 2022*

- **Digital Asset Viewer**: Built a web-app allowing users to view digitalQ assets stored in their Solana wallets, using data that conformed to Metaplex standards. Helped implement a challenge system that allowed for users to unlock off-chain digital assets once proof of some behaviour was verified (e.g. posting a tweet).
- **Protocol Buffer API Integration**: Worked on integrating protocol buffer APIs using bufbuild, using a gRPC-web layer to translate requests in a format suitable for the gRPC-based backend.
- **Next.js Frontend Developement**: Used Next.js to build a web-app to allow for server-side rendering, improving client-side performance and page SEO.

Academic Projects

Compiler - Built With Rust
A compiler for a simple while-based language called WACC, with added features like pattern-matching and anonymous functions.

- **Syntactic and Semantic Analysis Using Parser Combinators**: Used a parser combinator framework called Nom and leveraged Rust's strong typing and memory safety to build a fast and correct parser. Traversed parse-tree for semantic checking.
- **Code Optimisation and Generation**: Transformed AST into Control Flow Graph to perform liveness analysis and Chaitin's Graph Colouring Algorithm. Converted back to a linear format to generate assembly code.
- **Compiled to WASM**: Compiled the project to WASM for use in a web-app, allowing for blazingly fast WACC compilation from the web. Used with Next.js.

Pintos - OS Kernel Implementation Written in C
A project to implement significant parts of a miniature UNIX-style kernel, debugged extensively with GDB.

- **Priority Scheduling and Donation**: Implemented priority scheduling using multi-level feedback queues and ensured that priority donations were implemented to avoid deadlocks, and relevant priorities were propagated down.
- **File System and User Program Implementation**: Added processes and their relevant system calls using file descriptors. Ensured that processes were properly waiting, orphaned and exited using semaphores as state-machines.
- **Adding Virtual Memory**: Implemented a second-chance algorithm for page eviction, and developed relevant data structures for the frame and swap tables to allow for virtual memory.

Hobbies

Cricket: Play both club and university level cricket, and have held multiple committee positions - and love every season!
Music: Enjoy picking up new instruments and teaching myself to play the songs I love.
Google DSC Technical Director: Deliver lectures on frontend development as part of Google's DSC program.
Pub Quizzes: Love going to a pub quiz now and again!