James Nicholson Thomas Jr.

jthomas8@gatech.edu

Education

Georgia Institute of Technology **Atlanta, GA**
Bachelor of Science in Computer Engineering GPA: 2.62 (3.90 since return)
Spring 2019 - Fall 2022 (64 Credit Hours) Major GPA: 3.59

Georgia State University **Atlanta, GA**
Studied Bachelor of Science in Actuarial Science GPA: 4.03
Fall 2010, Summer 2012 - Fall 2012, Summer 2018 - Fall 2018 (29 credit hours)

Georgia Institute of Technology **Atlanta, GA**
Bachelor of Science in Computer Engineering GPA: 1.92
Fall 2010 - Spring 2016 (68 credit hours)

Skills

Software: Linux, Microsoft Windows, macOS/OSX, Mathematica, Matlab, GCC/LLVM, Bash, Packet Filter/iptables, Unix CLI Tools, vim, MYSQL Server, MS SQL Server, Version Control (SVN, GIT), Servers (Apache/Windows), Django, Lean Methodology/Agile Programming, Quartus II, MATLAB, Google Test Suite, Eclipse, HSpice, Cadence, Virtuoso, Virtual Box, VMWare, Qemu
Programming: Python, C, x86, aarch64, C++, Java, Rust, VB, VHDL, NumPy, SciPy, CMake & Make, sed & awk, Jupyter, regular expressions, libvirt, OpenMP, OpenMPI, gRPC, pytorch, Karas, LaTex, SQL
Instrumentation: Oscilloscope, Multimeter, Google Test Suite, IDA pro, DTrace
Business: Macroeconomics, Inventory Management Techniques, Preparation and Procedures of Financial Statements (Statement of Cash Flows, Income Statement, Balance Sheet, and other Financial Reports), Managerial Reports, Discounted Cash Flows, Project Options under Uncertainty
Languages: English (native), German (Beginner)
Communication: Technical writing and reading of white papers, Textbook editing of Probability Theory from a Measure Theory Perspective
Certifications: Outdoor Leadership Instructor (NCOBS)
Interests: Inquiry into the nature and causes of startups, Guitar and Bass, Science Fiction Literature and Movies

Experience

NAND Development LLC **Atlanta, GA**
Co-Founder *January 2013 - January 2020*
Digital Systems Architects, Consultants, and Engineers - One stop shop for clients on a per need basis including software and hardware design decisions and implementation.

- Refactored database architecture, increasing client's ability to accept payments by reducing major inaccuracies in data using regular expressions.
- Successfully integrated several unique databases under one interface resulting in an increase in accepted payments and revenue of 150% for our client.
- Found and removed malware and hardened clients' pre-existing websites so they could be hosted and resume taking payments.

Georgia Institute of Technology **Atlanta, GA**
Undergraduate Teaching Assistant - School of Electrical and Computer Engineering
August 2019 - December 2019

- *Held office hours debugging and communicating basic organizational principles of the major components of a processor (core, memory hierarchy, I/O subsystem, heat) and basic operating system constructs that utilize them. Included debugging of students' implementations in VHDL as well as C++.*

July 2019 - August 2019

- *Held office hours debugging and communicating C / assembly basics including hash tables and performance optimizations.*

SECME **Atlanta, GA**
Web Database Developer *May 2011 - August 2012*
SECME provides opportunities for students, teachers, and parents to engage in STEM activities with local colleges, universities, and engineering partners with the mission of promoting diversity in STEM Education.

- Planned, refactored, and developed a database system spanning over 100,000 lines of code from Lockheed resulting in data requisition and presentation needed for grants.

Projects

Azure Zero Trust IoT - Senior Design

- Using Nordic IoT devices integrated into Azure Cloud, designed, implemented, secured and attempted to penetration test an industry IoT solution with Boeing.

Hardened OpenBSD

- Hardened a distribution of OpenBSD at the level of Absolute OpenBSD allowing me to understand how to better harden many unix servers.

Project Euler Problems

- Worked on Project Euler problems in python with an emphasis on the mathematical solutions

Coursework

Introduction to Digital Signal Processing: Introduction to discrete-time signal processing and linear systems. Sampling theorem. Filtering. Frequency response. Discrete Fourier Transform. Z Transform. Laboratory emphasizes computer-based signal processing.
Programming for Hardware/Software Systems: Creation of complex execution and storage mechanisms, based on instruction set architecture, for software design including high-level programming languages and operating systems.
High-Performance Computer Architecture: Built simulations and studied fundamental structures in modern microprocessor and computer system architecture design, covering computer organization, instruction set design, memory system design, pipelining, cache coherence protocols, and other techniques exploring instruction level parallelism. Also covered system level topics such as storage subsystems as well as GPU architectures and interconnects/routers.
Compilers & Interpreters: Built a compiler in Java optimizing Tiger-IR and translating it into MIPS. Optimizations included reaching definitions, constant and copy propagation, live sets, register allocation and spilling algorithms. Studied finite state automata, regular expressions, and context free grammars, and several Front End software development packages.
Design - Operating Systems - experimental: Built an operating system from ground up for the Raspberry pi 3b+ (embedded/ARM) in Rust, course was based off of Stanfords cs140e but including additional Labs. Built a kernel module for FreeBSD on the Raspberry pi 3b+ in Rust. Learned and built systems such as boot loaders, file systems, interrupts, schedulers, and virtual memory and how to create a basic shell to take advantage of it.
VLSI and Advanced Digital Design: Built and optimized a 16-bit adder using custom cells. Analyzed and optimized designs using logical effort and Elmore delay. Learned about 6T XOR and alternative technologies to CMOS as well as some compilation and optimizations performed by Cadence. Learned about abstractions in the design process and how the circuits are built up and why the rules exist for each technology (eg. 45nm).
Machine Learning: Studied the math for supervised, unsupervised and reinforcement learning algorithms. implemented algorithms for random Fourier features, regression algorithms, k-means clustering, and Gaussian mixture methods using array broadcasting and log function to optimize computation time in python. Implemented and trained a generative adversarial network (ProGAN/StyleGAN2) that created vaporwave art aided by transfer learning.
Deep Learning: Introduction to the basics of Neural Networks (NNs) and some cutting-edge research. Topics include Convolutional NNs, Recurrent NNs, Deep Reinforcement Learning, Deep Structured Prediction and the mathematics and papers behind them. Project on MIR and NLP resulting in generated lyrics given an instrumental sample and an initial token.
Advanced Operating Systems: Studied topics such as operating systems structuring, synchronization, communication and scheduling in parallel systems, distributed systems, their communication mechanisms, distributed objects and middleware, failures and recovery management, system support for Internet-scale computing and security. Projects included load balancing CPU usage and Memory allocation via balloon driver for linux guests. Implementation and analysis of various synchronization barriers for high speed computer clusters in OpenMP and MPI. Implemented a shopping cart backend to multiple inventory servers using GRPC. Implemented MapReduce.
Introduction to Malware Reverse Engineering: Exposure to immersive, hands-on experience in the dissection and analysis of the code, structure, and functionality of malicious software.
Introduction to Quantum Computing: Superposition, entanglement, quantum gates, teleportation, and famous algorithms (Simon's, Grover's, Shor's, and the QFT). Implementation and analysis of quantum algorithms. Quantum Error Correct Circuit and optimizing quantum code to solve real-world problems in the NISQ-era. Built and modified compiler and transpiler parts of qiskit based on current papers.
Engineering Optimization: Topics include modeling with networks and graphs; linear, nonlinear, and integer programming; construction of models employing modern modeling languages; and general solution strategies.
Capital Investment Analysis: Core concepts and techniques for capital investment analysis, as well as the basic terminology, fundamental economic ideas and practical issues relevant to corporate finance, financial management and financial engineering.
Stochastic Manufacturing and Service Systems: Stochastic modeling techniques and managerial insights for design and control or manufacturing and service systems.
Signals and Systems: Continuous-time linear systems and signals, their mathematical representations, and computational tools; Fourier and Laplace transforms, convolutions, input-output responses, stability.
Introduction to Modern Physics: Ability to solve relativist problems using Lorentz Transformations, solve problems in elementary quantum mechanics, and solve problems in nuclear instability originating from radioactivity, nuclear fission and fusion.