Ji-Xinyou (Jerry)

+86 186-6538-6390 | jerryji0414@outlook.com | Github: github.com/Ji-Xinyou

EDUCATION

Shanghai Jiao Tong University, School of Electrical Information and Electrical Engineering **Shanghai, China**
B.E. in Computer Science and Engineering *Expected Jun 2023*

- **GPA**: 4.09/4.30 | **Overall Score**: 93.70/100 | **Rank**: 2/120 | **TOEFL**: 108/120
- **Courses**: Data Structure & Algorithm (97, final 100), Computer Architecture (96), Computer Network (100), Algorithm & Complexity (96), Computer Organization (93), The Mathematical Foundation of Computer Science (100), Linear Algebra (97), Probability & Statistics (100), Discrete Math (100), Artificial Intelligence (98) etc.
- **Online Courses:** {CS144(Computer Network), CS231n(Visual Recognition)}@Stanford, {6.S081 (O/S), 6.824(Distributed System) @MIT

SKILLS

- **Programming Languages:** C/C++, Python, Rust(novice), Go(novice), Verilog(novice)
- **Others:** GNU/Linux, Vim, PyTorch, Latex, git, gdb

PROJECTS

TCP/IP Stack, Network Interface & Router (cs144 course labs) – Modern C++

- Implemented a TCP/IP stack, consisting of Byte Stream, Stream Assembler, TCP sender/receiver and TCP connection based on Stanford CS144's libsponge and framework codes, which is able to communicate with other TCP/IP stack through Linux TUN service
- Further, implemented an Ethernet Interface with ARP, which wraps the datagram from upper self-implemented protocol stack into Ethernet Frame and send them out through TAP
- Finally, based on the Network Interface, implemented a router with multiple NIs, making the whole stack capable to communicate with other servers on its own

MapReduce & Fault-Tolerant KV Server on Raft (6.824 course labs) – Go

- Implemented a MapReduce framework, including the master and workers. The framework uses RPC as a simulation of network interactions and json parsing as a simulation of remote read by workers
- Build a Fault-Tolerant distributed KV server based on Raft consensus algorithm, which uses snapshotting for log compaction and is fault-tolerant under n/2n+1 machine failures and network failures

MIPS-32 Five-Stage Pipelined CPU – Verilog

- Implemented a five-stage (Fetch, Decode, Exec, Mem, WB) pipelined CPU based on MIPS-32 architecture design, which is emulated in Vivado
- The CPU takes the assembly codes in binary scheme (supports 16 instructions) as input, decodes the binary, obtain the opcode, executes and finally either writing/reading to/from memory or registers
- Implemented stall and forwarding mechanisms to better pipeline the CPU execution by avoid hazards causing the CPU to waste clock cycles

Modification in xv6, a unix-like O/S (6.828 course labs) - C

- Implemented some kernel-level modern optimization features in xv6, including large files based on two-level inode, copy-on-write fork and memory-mapped files based on page fault mechanism. Also, optimized the performance of buffer cache and kernel memory allocator by using a more fine-grained locking scheme
- Build some user-level utilities in xv6, including backtrace, xargs. symbolic link and a network driver for e1000 NIC

Neural Style Transfer – Python

- Implemented a demo based on paper *Gatys L A, Ecker A S, Bethge M. A neural algorithm of artistic style*, which is able to transform the artistic style of the input content image from the input style image
- The algorithm uses intermediate features of the pretrained network to compute Gram matrix, using the matrix as the sign of style similarity

EXTRACURRICULAR

- Kaggle Lux AI Challenge: Rank 88/1178 (Top 8%), Bronze Medal
- 2019-2020 School B-Level Scholarship
- Basketball (10 yrs+)