Yu-Ching Chen

Unit 802 - 7 Concorde Place
North York • Ontario • M3C 3N4
(416) 906-5378 • yuch.chen@mail.utoronto.ca
http://yuch7.github.io

SUMMARY

Completed computer science specialist program at University of Toronto focusing on computer systems. Interested in most aspects of computer science, but more interested in C and hardware programming. Have been using Linux operating system for 4 years and currently using Arch Linux. A member of University of Toronto's computer science student union; helping out on their events and students in computer science needing help.

PROFESSIONAL EXPERIENCE

Junior Developer — *June '11 – Sept '11*
411.ca

- Created PHP scripts running tests on a Selenium server.
- Back end web testing on test servers and on primary server
- Active part of the IT department, participated in scrum meetings and engaged in the sprints.
- Set up local server for localized controlled testing

TECHNICAL

- C/C++
- Java
- Python
- Verilog
- assembly language

EDUCATION

University of Toronto — Expected Graduation: June 2016
Computer Science Specialist (computer systems focus)
Honors B.Sc

Abstract

Data Structures and Analysis
Tradeoffs of performance to storage complexity of many different data structures. From basic data structures, able to augment to what is needed knowing tradeoffs.

Algorithm Design, Analysis and Complexity
Different approaches to solving different problems, from looking at a dynamic approach, greedy approach, graph (states) theory, divide and conquer, etc. Able to understand NP problems and provide verify algorithms to prove NP completeness.

Advanced Operating Systems
Task scheduling, resource allocation, concurrency, file systems, threads, kernel constraints (eg. virtual machine), resource sharing, interrupts, Inter-process communication and the tradeoffs of using different ones.

Design patterns
Different design patterns and why they are used, mainly from the gang of four. Creational, structural and behavioral design patterns.

Computer Graphics
Limitations in graphics simulation, how graphics are rasterized, object, world, camera views, image processing, video interpolation.

Networks
OSI model and the use of all layers. Routing protocols, network security attacks, from autonomous systems to hubs/routers, NATS, firewall, packet framing/multiplexing etc.

Hardware
Tradeoffs of space and/or power consumption, when to use hardware and when to use a CPU/microcontroller, use of failsafe mechanisms (eg. watchdog), hardware communication (I^{2}C, usb, Bluetooth, etc.)

Projects/Volunteer:

Fault tolerant chat server (2016)
UDP and TCP chat client that interfaces with the server given a specific http protocol as a distributed system. Client must connect to server by obtaining information from a static location server and exchange control packets using TCP while opening channels open UDP sockets for message exchange. Client must also be user friendly and fault tolerant to the point where users never experiences bugs or crashes. *Code provided on github. (CSC469)*

CPU microbenchmark (2016)
Purpose of the benchmark is to explore how Operating Systems delegate tasks and the time it takes, specifically x86 systems running Linux. Also to understand how numa nodes work and benchmarking the speed it takes for CPUs to read and write from the nodes. Able to explain what each benchmark tests and compare with other technologies. *Code provided on github. (CSC469)*

CSSU events (2015-2016)
For majority of the hack nights hosted this year, I mentored and supervised peers on any help they needed with any hacks to the best of my ability. Also helped set up and run other events.