Keerthi Gowda B S

keerthis@vt.edu | +1-540-449-8943 Github | LinkedIn

Objective

Looking for full time opportunities in software development.

Education

Virginia Tech, *Blacksburg, Virginia (Expected: May 2018)*
Master's in Computer Engineering **GPA: 4.0**

R V College of Engineering, *Bangalore, India (Aug 2009 – June 2013)*
Bachelor's in Electronics and Communication Engineering **GPA: 9/10**

Skills

- Java
- C
- Node.js
- Processing.js
- MongoDB
- Python
- C++
- Matlab

Professional Experience

Qualcomm, *San Diego (May 2017 – Aug 2017)*
Interim Engineering Intern

Developed smart watch emulator tool for ambient mode watch faces. This would assist the watch face designers to develop and visualize watch faces within the color and space constraints in ambient mode.

Virgina Tech, *Blacksburg, Virginia (Jan-2017 – present)*
IT Programmer

Created an interface to authenticate to a MongoDB database and export the excel data. Built a query language to bridge the user input and database to update database, import from database and generate reports.

Cypress Semiconductor Corporation, *India (July 2013 – July 2016)*
Senior Applications Engineer

- Designed an end system to remotely control a NERF blaster using an android phone over Bluetooth Low Energy (BLE). The system was automated to fire any object within a specified distance.
- Developed library for Single Edge Nibble Transmission (SENT) protocol, Society of Automotive Engineers (SAE) J2716 standard, using programmable logic and FIFO.
- Created code examples for interfacing external sensors like Inertial Measurement Unit (IMU), pressure sensor, ultrasonic sensor and gas sensor to PSoC over serial communication protocols such as I2C, SPI, UART, and CAN.

Projects

Lockfree Red-Black Tree based CFS Scheduler *(java)*

Developed and implemented an algorithm for lock free insertions and deletions in the red-black tree. This would allow multiple threads to concurrently operate on the tree and maintain data consistency. The throughput of lock free red-black tree was improved by approximately 20% compared to lock version. Implemented priority CFS scheduler using this lock free red-black tree.

Recursive Descent Parsing *(c++)*

Implemented recursive descent parsing algorithm for University Simple C (USCC) compiler, which is a simple version of LLVM compiler. The parser verifies the user written 'C' code to be syntactically and semantically correct. Finally, an abstract syntax tree is generated which is further used for optimization and code generation.

Priority scheduler in linux *(c)*

Modified the linux kernel to replace the conventional CFS scheduler with the max heap based priority real time scheduler. All the runnable tasks would be added to the max heap based on their priority and the operating system picks the next highest priority task from the heap.

Game Design *(processing.js)*

- **Soccer** - Developed soccer game with intelligence built into non-playing characters to play with or against the active player to strategically attack or defend based on the current game play.
- **Pathfinder** - Designed a tile-map game to find the path to the given destination using Breadth First Search (BFS) and Depth First Search (DFS) algorithms.
- **Basketball** – Game was designed to be realistic considering the initial velocity of the ball, gravity, wind velocity and direction, bounce from the ground and bump from the basket.

Awards

- *Best Breakout Session* award at Cypress Sales Training Conference - 2014, held in Seattle and Shanghai.
- *Best Project Award* at RVCE, ECE -2013, for the undergraduate final year project on Thin Film Transistors.
- *Awarded Best Debutant*, *Never Say Die* and $2000 in Student Unmanned Aerial Systems (SUAS), 2012 by AUVSI.