Siddharth **Patel**

(856)-426-6113
siddharthpatel
spatelak@purdue.edu
siddharthpatel

EDUCATION

Purdue University | B.S. in Computer Science | **GPA: 3.80** Jan 2016 – Dec 2018

- **Related Coursework**: Data Structures & Algorithms | Systems Programming | Design & Analysis of Algorithms | Computer Architecture | Operating Systems (Fall'17) | Web Application Development (Fall'17)

Rutgers University | B.S. in Computer Science | **GPA: 3.68** Jan 2014 – Dec 2015

- Classes and Programs audited at **Princeton University:** Introduction to Programming Systems (COS 217) | Data Structures and Algorithms (COS 226) | Program for Algorithmic & Combinatorial Thinking (PACT)

WORK EXPERIENCE

Teaching Assistant Purdue University Jan 2017 – May 2017

- Lab mentor and grader for course: CS 251 - Algorithms and Data Structures

Software Verification Engineer Delphi Aug 2016 – Nov 2016

- **Developed plugins** with a toolset that monitored and sent CAN serial messages through USB ports
- Verified message requirements from the Product Definition Specification

Peer Tutor Rutgers University Jan 2015 – Aug 2015

- **Tutored** introductory **Python Programming, Discrete Math**, Physics and all Math courses up to Calculus 2 courses to the college undergraduates.

INDEPENDENT PROJECTS

1) Chorus | **Python, Flask**

- Developed a **web application** that allows people to vote on next song to be played. Using Spotify and Facebook auth, we get user's playlist and relieve the DJ stress of selecting songs.
- Designed backend architecture for the application and made it integrate with **database**, **UI** and **Spotify APIs** using **Flask**. Learnt **HTML, CSS** and **JavaScript**.

2) EDF Scheduler and Extended Message Passing System for XINU | **C**

- Modified XINU to use **Earliest Deadline First (EDF) scheduler** which is a dynamic scheduling policy, instead of Static Priority based cyclic execution.
- Developed an extended IPC message passing interface for XINU that allows multiple messages to be outstanding using **constant extra space**. The interface provides sending/receiving of messages between processes in **constant time**.

3) Simplified Linux Shell | **C, C++, LEX, YACC**

- Implemented scanner and parser for the shell with **LEX** and **YACC.**
- Implemented simplified Linux shell from scratch that provides similar functionality as Bash, like, **IO redirection, execution of simple commands, file redirection**.

4) Web Server | **C, C++**

- Developed a web server application which users can use to host their website on any computer using different concurrency modes.
- Learnt about **HTTP requests**, **socket programming** and **concurrency modes**.

5) Burrows-Wheeler | **Java**

- Developed an application using **Burrows-Wheeler** data compression algorithm which transforms a piece of text in which sequences of same characters occur near each other many times thus, making it easier to compress. The implementation **reduces compression size by a factor of 3** as compared to PKZIP and gzip.

SKILLS AND TECHNOLOGIES

Java, C, Python, Flask, Shell, x86 Assembly, ARM, Git, HTML (beginner), CSS (beginner), JavaScript (beginner)