P4 Compiler Documentation Site

Date: 2nd April, 2024
Author: Farhan Sadeek
Academic Affiliation: Stanford University & The Ohio State University
Major: Computer Science and Mathematics (Theoretical Track)
Minor: Physics and Quantitative Economics
Professional Affiliation: Spectrum *(NYSE:CHTR)*
Email: sadeek.1@osu.edu
Github profile: SadeekFarhan21
Linkedin: Farhan Sadeek
Personal Website: https://farhan-sadeek.netlify.app/
Project Length: 90 hours
Mentors: Davide Scano, Bili Dong, Fabian Ruffy
Timezone: Eastern Day Time (UTC - 4:00)

Motivation

Last year, when I was in high school, I got extremely interested in working with Arduino and Raspberry Pi. Together with me and my cousin, a network engineer, built a real-radar system with the microcontroller system. And that embarks the journey of working with embedded systems.

While I was browning through the list of organizations participating in GSoc, I made it my goal because it is one of the closest projects that I could work with. Along with that, I am currently working as a solutions engineer for Spectrum **(NYSE:CHTR)** and a Computer Science student at the Ohio State University. The first one taught me technical writing and the second gave me the knowledge to understand how computers and electronic devices work. And the best part about the project is that I can start writing anytime I can. Because of all of the above-mentioned reasons, I would like to be part of this program.

Use case

Both p4lang and tutorials pages don't have a well-organized documentation page which causes issues for beginners when looking for a one-stop solution for all the necessary information for them to get started. On top of that, it enhances collaboration and troubleshooting for further use cases even for experienced developers. So, creating well-formatted documentation is necessary for both experienced and beginner p4 learners.

This project aims to add more well-formatted documentation both on GitHub and a website. The website will be created using a Python framework called Sphinx allowing faster and easier rendering.

Initial Research

- Emailed mentors about potential solutions for the documentation of the project.
- Browsed through the three recommended formats for the documentation site.
- Familiarized with the Sphinx framework for better workflow.
- Created framework/roadmap on the way of implementing this scalable documentation site.

Goals

- Collect all the existing documentation and merge all of the documentation inside the docs folder.
- Inside the docs folder separate the files based on their type (getting started, install, tutorial, etc).
- Create and host a website using a Python Framework called Sphinx.

Deliverables

The following are the deliverables for this project

- A well-formatted documentation website using Sphinx
- An updated documentation in Github for a better understanding of the project

Directory Structure

```
docs
│
├── _static
│   ├── css                    - css file for custom formatting
│   └── img                    - images for the website
│
├── index.rst                  - index of the website created by Sphinx
│
├── getting_started
│   ├── package_version        - installation from package version
│   │   ├── ubuntu.rst         - for ubuntu
│   │   └── debian.rst         - for debian
│   └── source                 - installation from source
│       └── source.rst
│
├── dependencies               - dependencies across different OS
│   ├── ubuntu.rst
│   ├── fedora.rst
│   ├── macOS.rst
│   ├── garbage_collector.rst  - details about garbage collector
│   └── crash_dumps.rst        - details about crash dumps
│
├── development_tools          - developer tools
│   ├── docker.rst             - for docker
│   └── bazel.rst              - for bazel
│
├── reference.rst              - reference for additional information
│
└── known_issues               - known issues inside of the repository
    ├── frontend               - front end issues
    │   ├── common.rst
    │   ├── p4-14.rst
    │   ├── p4.rst
    │   └── parsers
    └── backend                - backend issues
    │   ├── bmv2.rst
    │   ├── dpdk.rst
```

```
│   ├── ebpf.rst
│   ├── graphs.rst
│   ├── p4test.rst
│   ├── 5c.rst
│   └── ubpf.rst
├── specs
    └── specs.rst
```

This is a template of the project that I will work on. However, some of the folder structures can change in the future for example, my mentor may want to create a folder for reference instead of a single reStructuredText file.

Project Timeline

This document outlines the project timeline for organizing and documenting the P4 compiler project. The project is divided into 12 weeks, with each week focusing on specific tasks and deliverables.

Before:

- **Tasks:**
  - Read through different well-formatted documentation to get a better writing style
  - Get a good understanding of the problem.
  - Set up the development environment.
  - Consult with mentors.
  - Lurk and ask questions on Slack and the issues page.
- **Deliverables:**
  - A better understanding of the skills in the documentation
  - Initial review of project structure and requirements.

Week 1: Familiarization and Setup

- **Tasks:**
  - Review project requirements and documentation structure.
  - Familiarize yourself with p4c documentation tool.
- **Deliverables:**
  - Environment set up with Sphinx installed.
  - Initial review of project structure and requirements.

Week 2: Initial Documentation Organization

- **Tasks:**
  - Create a skeleton for the documentation.
  - Set up a basic structure for index, getting started, dependencies, development tools, reference, and known issues sections.
- **Deliverables:**
  - Initial directory structure and empty documentation files.

Week 3-4: Populate Getting Started and Dependencies Sections

- **Tasks:**
  - Populate getting started section with installation instructions.
  - Fill in details about dependencies for different operating systems.
- **Deliverables:**
  - Complete installation guides for both package version and source installations.
  - Detailed documentation on dependencies for Ubuntu, Fedora, and macOS.

Week 5-6: Development Tools and Reference

- **Tasks:**
  - Document developer tools such as Docker and Bazel.
  - Compile a reference section with additional information.
- **Deliverables:**
  - Complete documentation for Docker and Bazel usage.
  - Initial reference section with relevant details.

Week 7-8: Known Issues

- **Tasks:**
  - Identify known issues in the repository.
  - Document frontend and backend issues separately.
- **Deliverables:**
  - Detailed documentation of known frontend and backend issues.
  - Separate section for specifications.

Week 9-10: Styling and Formatting

- **Tasks:**
  - Add CSS files for custom formatting.
  - Insert images for enhanced visualization.
- **Deliverables:**
  - Styled and visually appealing documentation.
  - Images embedded where necessary for clarity.

Week 11: Review and Revision

- **Tasks:**
  - Review the entire documentation for consistency and completeness.
  - Make necessary revisions and improvements.
- **Deliverables:**
  - Finalized version of the documentation ready for deployment.

Week 12: Final Touches and Deployment

- **Tasks:**
  - Perform final checks and testing.
  - Prepare documentation for deployment.

- **Deliverables:**
  - Deployed and accessible documentation website.
  - Project completion report summarizing the process.

Related Pre-proposal works

Pull Request #545: I updated the list of details for the tutorials in the tutorials repository

Post Google Summer of Code Plans

I plan to harvest the experience and knowledge I will gain from GSoC and move forward to making regular contributions to open source projects. I am fairly new to the concept. I have been doing projects and uploading them publicly for a year now but wasn't aware of the big community behind many of the tools I have been using. I would like to show people my work and ask them to join in on the mission whilst also contributing to other projects to gain experience and pass it on. Also, I plan to help my juniors prepare for next year's GSoC, introduce them to open source, and help them get a head start on projects my club mates and I have worked on in GSoC.