eBPF ABI

Want to collaborate?

What *is* an ABI?

Everyone *knows* that, *obviously!*

What *is* an ABI?

Everyone *knows* that, *obviously!*

What *is* an ABI?

Everyone *knows* that, *obviously!*

An application binary interface (ABI) defines the requirements that one or more binary software objects must meet in order to guarantee that they can interoperate and/or use the resources provided by operating systems/hardware combinations.

Overview

- Goal

- Who Am I? Why Am I Qualified?

- Early Effort and Collaboration Hub

- Going Forward

Goal

Create a *generic* ABI for eBPF, with support for processor-specific supplements for different *implementations*

- E.g., the Kernel, Windows, FreeBSD, Agilio CX

The scenario would resemble the setup for the SystemV ABI.

Who Am I? Why Am I Qualified?

- I am *not* qualified! But,

- I *love* obscure things that are heavy on details.

- I am a computer science professor and co-maintainer of the uBPF implementation with Alan Jowett from Microsoft.

Who Am I? Why Am I Qualified?

- I am *not* qualified! But,

- I *love* obscure things that are heavy on details.

- I am a computer science professor and co-maintainer of the uBPF implementation with Alan Jowett from Microsoft.

Early Effort and Collaboration Hub

https://github.com/hawkinsw/ebpf-system-ad-abi

The current text has been scraped together in a few weeks – please be gentle!

Open Question: Is this setup an acceptable means of collaboration?