

eBPF ELF Profile Specification draft-thaler-bpf-elf

Dave Thaler <dthaler@microsoft.com>

WG Charter

The working group will produce one or more documents on the following work item topics (with intended document status annotations, e.g. [PS] Proposed Standard and [I] Informational)

...

- **[PS]** the BPF Type Format (BTF) that defines debug information and introspection capabilities for BPF programs,
- **[I]** one or more documents that recommend conventions and guidelines for producing portable BPF program binaries,

Executable and Linkable Format (ELF) files

- Specified in Chapter 4 of the System V Application Binary Interface, last dated 1997, published by SCO
- Also in Tool Interface Standard, dated 1993, published by Linux Foundation
- Used across industry (not just System V or Linux)
- Supports many different “machine” architectures
- BPF uses a profile of the general ELF format
 - BPF is another “machine” architecture

BPF ELF Profile

- Defines a new “machine” value (247 = BPF)
- Specifies:
 - Which existing values of other ELF header fields apply
 - `e_type`, `e_ident[EI_CLASS]`
 - Meaning of TEXT sections (BPF programs)
 - Meaning of DATA sections (maps)
- Discusses section naming conventions
- Contents of other special sections (version, license, etc.)

Relationship to BPF Type Format (BTF)

- In Linux kernel repo:
 - elf.rst – Contains some non-BTF ELF profile info
 - btf.rst – Contains
 - A) non-ELF BTF info
 - B) “4. ELF File Format Interface”
- Currently the draft is elf.rst + part B of btf.rst
 - (plus boilerplate, plus some things not yet upstreamed)
- Recall that the BTF WG item is [PS] but ELF is [I]
 - Which parts of btf.rst should be [PS] vs [I]
- Should all BPF ELF profile content be in one document?
 - Can still reference BTF document for things not ELF-specific

TEXT section naming conventions

- Mentions three different conventions in use today:
 - **Prefix Convention:** The section name is prefixed with a string that identifies the program type, so that the program type of any programs in the section can be determined by finding the longest substring match across all program type prefixes.
 - **Exact Match Convention:** The section name is a string that identifies the program type of any programs in the section.
 - **Arbitrary Convention:** The section name can be anything and the program type of any programs in the section must be determined without consulting the section name.

DATA sections

- Two maps section formats:
 - Classic (non-BTF) maps: “maps” or “maps/*map-name*”
 - Contents specified in the doc
 - BTF maps: “.maps” section
 - TODO in the doc

Other optional special sections

- “**license**”: SPDX identifier that a runtime can optionally use to restrict what program types and/or helper functions can be used
- “**version**”: runtime-specific value that a runtime can optionally use to determine whether program is meant for this runtime version
- “**.BTF**”: BTF type and string data
- “**.BTF.ext**”: source file and line info for the first BPF instruction (identified by offset into data section) per source line
- “**.BTF_ids**”: BTF ID values used by the runtime

Next steps?

- Whether/how factor document?
- Adopt as WG item?
- Co-editors?