Name

bpftool-link − tool for inspection and simple manipulation of eBPF links

SYNOPSIS

bpftool [*OPTIONS*] **link** *COMMAND*

OPTIONS := { { **−j** | **−−json** } [{ **−p** | **−−pretty** }] | { **−d** | **−−debug** } | { **−f** | **−−bpffs** } | { **−n** | **−−nomount** } }

COMMANDS := { **show** | **list** | **pin** | **help** }

LINK COMMANDS

bpftool link { show | list } [*LINK*]
bpftool link pin *LINK FILE*
bpftool link detach *LINK*
bpftool link help

LINK := { **id** *LINK_ID* | **pinned** *FILE* }

DESCRIPTION

bpftool link { show | list } [*LINK***]**
Show information about active links. If *LINK* is specified show information only about given link, otherwise list all links currently active on the system.

Output will start with link ID followed by link type and zero or more named attributes, some of which depend on type of link.

Since Linux 5.8 bpftool is able to discover information about processes that hold open file descriptors (FDs) against BPF links. On such kernels bpftool will automatically emit this information as well.

bpftool link pin *LINK FILE*
Pin link *LINK* as *FILE*.

Note: *FILE* must be located in *bpffs* mount. It must not contain a dot character ('.'), which is reserved for future extensions of *bpffs*.

bpftool link detach *LINK*
Force−detach link *LINK*. BPF link and its underlying BPF program will stay valid, but they will be detached from the respective BPF hook and BPF link will transition into a defunct state until last open file descriptor for that link is closed.

bpftool link help
Print short help message.

OPTIONS

−h, **−−help**
Print short help message (similar to **bpftool help**).

−V, **−−version**
Print bpftool's version number (similar to **bpftool version**), the number of the libbpf version in use, and optional features that were included when bpftool was compiled. Optional features include linking against LLVM or libbfd to provide the disassembler for JIT−ted programs (**bpftool prog dump jited**) and usage of BPF skeletons (some features like **bpftool prog profile** or showing pids associated to BPF objects may rely on it).

−j, **−−json**
Generate JSON output. For commands that cannot produce JSON, this option has no effect.

−p, **−−pretty**
Generate human−readable JSON output. Implies **−j**.

−d, **−−debug**
Print all logs available, even debug−level information. This includes logs from libbpf as well as from the verifier, when attempting to load programs.

−f, **−−bpffs**
When showing BPF links, show file names of pinned links.

−n, **−−nomount**
Do not automatically attempt to mount any virtual file system (such as tracefs or BPF virtual file system) when necessary.

EXAMPLES

 # bpftool link show

```
10: cgroup  prog 25
        cgroup_id 614  attach_type egress
        pids test_progs(223)
```

 # bpftool −−json −−pretty link show

```
[{
        "type": "cgroup",
        "prog_id": 25,
        "cgroup_id": 614,
        "attach_type": "egress",
        "pids": [{
                "pid": 223,
                "comm": "test_progs"
            }
        ]
    }
]
```

 # bpftool link pin id 10 /sys/fs/bpf/link
 # ls −l /sys/fs/bpf/

```
−rw−−−−−−− 1 root root 0 Apr 23 21:39 link
```

SEE ALSO

bpf(2), **bpf−helpers**(7), **bpftool**(8), **bpftool−btf**(8), **bpftool−cgroup**(8), **bpftool−feature**(8), **bpftool−gen**(8), **bpftool−iter**(8), **bpftool−map**(8), **bpftool−net**(8), **bpftool−perf**(8), **bpftool−prog**(8), **bpftool−struct_ops**(8), **bpftool−token**(8)