

NAME

tslog - Boot-time event tracing facility

SYNOPSIS

To compile this boot-time event tracing facility into the kernel, place the following line in the kernel configuration file:

option TSLOG

DESCRIPTION

tslog is a boot-time event tracing facility. It is suitable for tracing recursive events based on function entries and exits. Its purpose is to ease pinpointing and reducing the overall FreeBSD boot time by generating detailed timing information.

tslog is able to trace the boot loader, kernel initialization, and userland processes.

In userland, it records the following details for each process ID:

- The timestamp of the `fork(2)` which creates the given process ID and the parent process ID.
- The path passed to `execve(2)`, if any.
- The first path resolved by `namei(9)`, if any.
- The timestamp of the `exit(3)` which terminates the process.

SYSCTL VARIABLES

The following `sysctl(8)` variables are available:

debug.tslog

Dump the **tslog** buffer of recorded loader and kernel event timestamps.

debug.tslog_user

Dump the **tslog** buffer of recorded userland event timestamps.

FLAMEGRAPHS

The **tslog** buffer dumps can be used to generate flamegraphs of the FreeBSD boot process for visual analysis. See <https://github.com/cperciva/freebsd-boot-profiling> for more information.

SEE ALSO

dtrace(1), boottrace(4), ktr(4)

HISTORY

tslog first appeared in FreeBSD 12.0. Support for tracing boot loaders and userland process was added in FreeBSD 14.0.

TSLOG vs. Boottrace

tslog is oriented towards system developers while `boottrace(4)` is meant to be easy to use by system administrators. Both facilities provide an overview of timing and resource usage of the boot process.

TSLOG vs. DTrace

`dtrace(1)` is not always the right tool for profiling early kernel initialization. The reason is it requires some kernel subroutines which are not yet available early in the boot process, e.g.: traps, memory allocation, or thread scheduling. **tslog** depends on fewer kernel subroutines than `dtrace(1)` and because of that can trace early kernel initialization.

TSLOG vs. KTR

`ktr(4)` has a couple of limitations which prevent it from being able to run at the start of the boot process. In contrast, **tslog** is designed for logging timestamped events for boot profiling.

AUTHORS

tslog was written by Colin Percival <cperciva@FreeBSD.org>.

This manual page was written by Mateusz Piotrowski <0mp@FreeBSD.org>.