HOME ENGINEERING TRAINING DOCS COMMUNITY COMPANY









Linux debugging, tracing, profiling & perf. analysis

Check our new training course with Creative Commons CC-BY-SA lecture and lab materials

linux **№** *v5.15*

Elixir Cross Referencer

```
/ samples / kprobes / kretprobe example.c
                     Search Identifier
  All symbols
 1
     // SPDX-License-Identifier: GPL-2.0-only
 2
     /*
 3
      * kretprobe_example.c
 4
 5
       * Here's a sample kernel module showing the use of return probes to
       * report the return value and total time taken for probed function
 6
 7
       * to run.
 8
 9
        usage: insmod kretprobe_example.ko func=<func_name>
10
       * If no func_name is specified, kernel_clone is instrumented
11
12
13
       * For more information on theory of operation of kretprobes, see
14
       * Documentation/trace/kprobes.rst
15
      * Build and insert the kernel module as done in the kprobe example.
16
       * You will see the trace data in /var/log/messages and on the console
17
       * whenever the probed function returns. (Some messages may be suppressed
18
       * if syslogd is configured to eliminate duplicate messages.)
19
20
       */
21
     #include <linux/kernel.h>
22
     #include <linux/module.h>
23
     #include linux/kprobes.h>
24
     #include <linux/ktime.h>
25
     #include <linux/limits.h>
26
27
     #include <linux/sched.h>
28
29
     static char func_name[NAME_MAX] = "kernel_clone";
     module param string(func, func name, NAME MAX, S IRUGO);
30
     MODULE PARM DESC(func, "Function to kretprobe; this module will report the"
31
                               " function's execution time");
32
33
     /* per-instance private data */
34
35
     struct my data {
36
              ktime t entry stamp;
37
     };
38
```