Elixir Cross Referencer

```c
// SPDX-License-Identifier: GPL-2.0-only
/*
 * kretprobe_example.c
 *
 * Here's a sample kernel module showing the use of return probes to
 * report the return value and total time taken for probed function
 * to run.
 *
 * usage: insmod kretprobe_example.ko func=<func_name>
 *
 * If no func_name is specified, kernel_clone is instrumented
 *
 * For more information on theory of operation of kretprobes, see
 * Documentation/trace/kprobes.rst
 *
 * Build and insert the kernel module as done in the kprobe example.
 * You will see the trace data in /var/log/messages and on the console
 * whenever the probed function returns. (Some messages may be suppressed
 * if syslogd is configured to eliminate duplicate messages.)
 */

#include <linux/kernel.h>
#include <linux/module.h>
#include <linux/kprobes.h>
#include <linux/ktime.h>
#include <linux/limits.h>
#include <linux/sched.h>

static char func_name[NAME_MAX] = "kernel_clone";
module_param_string(func, func_name, NAME_MAX, S_IRUGO);
MODULE_PARM_DESC(func, "Function to kretprobe; this module will report the"
                        " function's execution time");

/* per-instance private data */
struct my_data {
        ktime_t entry_stamp;
};

```