МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №6

з дисципліни «Дослідження та проектування комп’ютерних систем - 2»

Виконав:

студент гр. ІО-91мн

Глушенок В.О.

Перевірив:

Каплунов А.В.

Київ 2020

**Завдання**

Завдання Basic 1:

Додайте BUG\_ON() замість друку повідомлення та повернення -EINVAL для неприпустимого значення параметра.

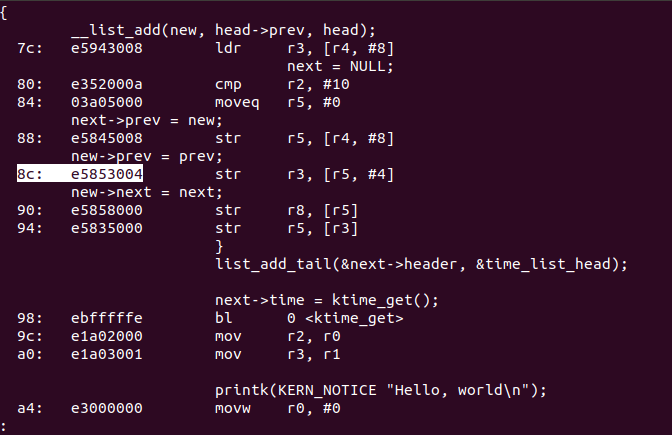
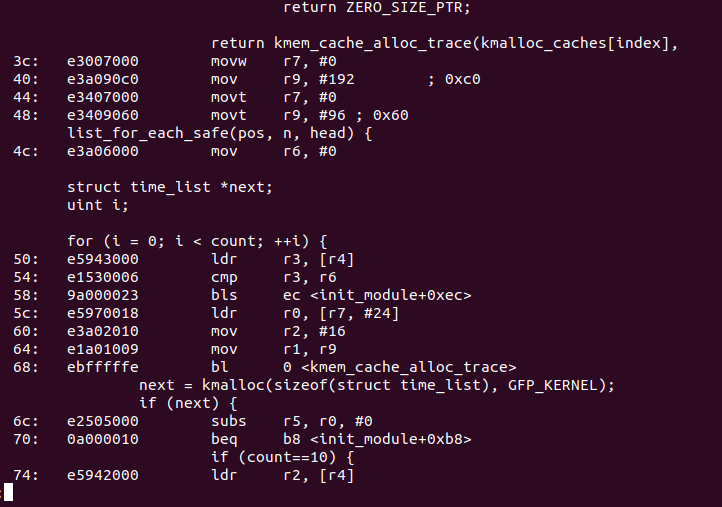
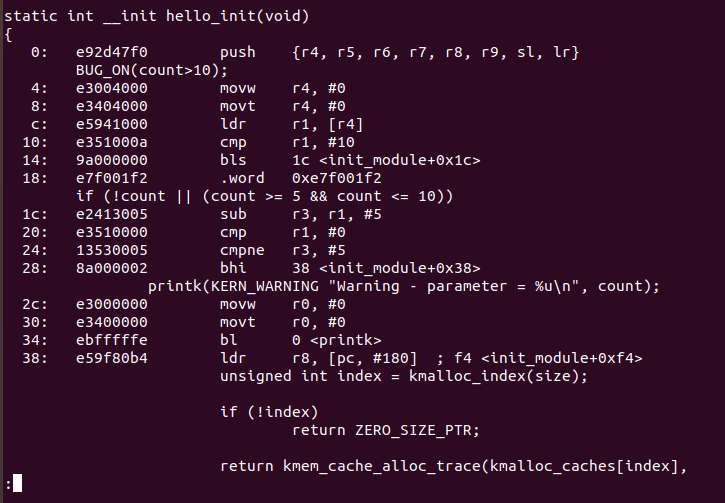
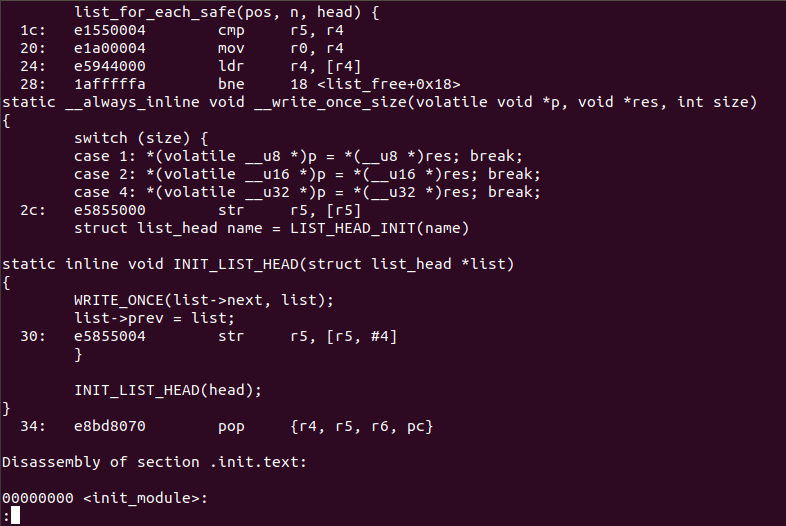
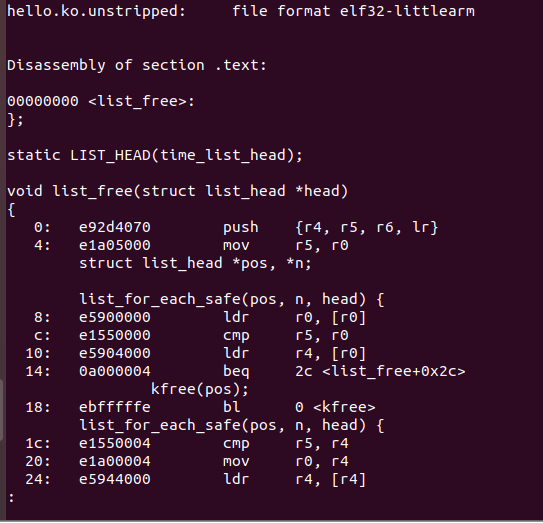
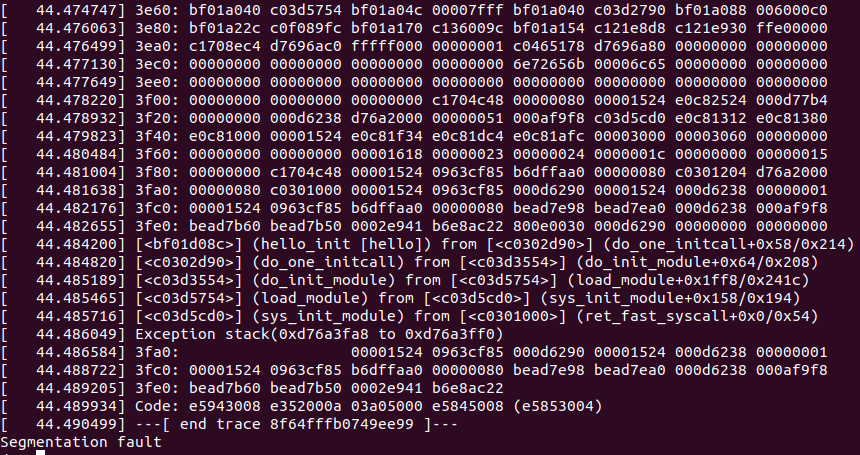
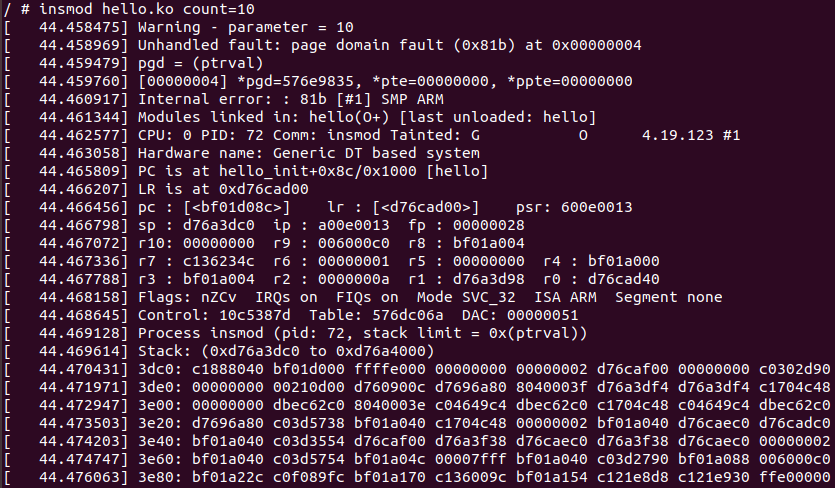
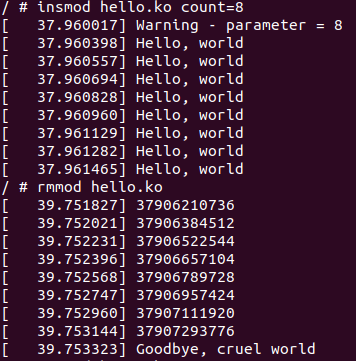
Додайте примусове внесення помилки “начебто kmalloc() повернув 0” під час формування елемента списку для якогось повідомлення (останнього із серії, 5-го, ... — на ваш вибір).

Модифікуйте Makefile аналогічно appendix1.

Отримайте обидва повідомлення, роздивіться їх та для одного з них виконайте пошук місця аварії аналогічно appendix1.

Зауважте, що при виконанні BUG\_ON() модуль буде “зайнятий”, і ви не зможете виконати rmmod.

**Приклад виконання програми**



**Програмний код**

**hello.c**

|  |
| --- |
| #include <linux/init.h> |
|  | #include <linux/module.h> |
|  | #include <linux/moduleparam.h> |
|  | #include <linux/errno.h> |
|  | #include <linux/ktime.h> |
|  | #include <linux/slab.h> |
|  |  |
|  | MODULE\_LICENSE("Dual BSD/GPL"); |
|  |  |
|  | static uint count = 1; |
|  | module\_param(count, uint, S\_IRUGO); |
|  | MODULE\_PARM\_DESC(count, "Here is parameter description"); |
|  |  |
|  | struct time\_list { |
|  | struct list\_head header; |
|  | ktime\_t time; |
|  | }; |
|  |  |
|  | static LIST\_HEAD(time\_list\_head); |
|  |  |
|  | void list\_free(struct list\_head \*head) |
|  | { |
|  | struct list\_head \*pos, \*n; |
|  |  |
|  | list\_for\_each\_safe(pos, n, head) { |
|  | kfree(pos); |
|  | } |
|  |  |
|  | INIT\_LIST\_HEAD(head); |
|  | } |
|  |  |
|  | static int \_\_init hello\_init(void) |
|  | { |
|  | BUG\_ON(count>10);  if (count > 10) { |
|  | printk(KERN\_ERR "Error - parameter > 10\n"); |
|  | return -EINVAL; |
|  | } |
|  |  |
|  | if (!count || (count >= 5 && count <= 10)) |
|  | printk(KERN\_WARNING "Warning - parameter = %u\n", count); |
|  |  |
|  | struct time\_list \*next; |
|  | uint i; |
|  |  |
|  | for (i = 0; i < count; ++i) { |
|  | next = kmalloc(sizeof(struct time\_list), GFP\_KERNEL); |
|  | if (next) {  if (count==10) {  next = NULL;  } |
|  | list\_add\_tail(&next->header, &time\_list\_head); |
|  |  |
|  | next->time = ktime\_get(); |
|  |  |
|  | printk(KERN\_NOTICE "Hello, world\n"); |
|  | } else { |
|  | list\_free(&time\_list\_head); |
|  | return -ENOMEM; |
|  | } |
|  | } |
|  |  |
|  | return 0; |
|  | } |
|  | module\_init(hello\_init); |
|  |  |
|  | static void \_\_exit hello\_exit(void) |
|  | { |
|  | struct list\_head \*pos; |
|  |  |
|  | list\_for\_each(pos, &time\_list\_head) { |
|  | printk(KERN\_NOTICE "%lld", ktime\_to\_ns(list\_entry(pos, struct time\_list, header)->time)); |
|  | } |
|  |  |
|  | list\_free(&time\_list\_head); |
|  |  |
|  | printk(KERN\_NOTICE "Goodbye, cruel world\n"); |
|  | } |
|  | module\_exit(hello\_exit); |

**Makefile**

|  |
| --- |
| # If KERNELRELEASE is defined, we've been invoked from the |
| # kernel build system and can use its language. | # kernel build system and can use its language. |
| ifneq ($(KERNELRELEASE),) | ifneq ($(KERNELRELEASE),) |
| obj-m := hello.o | obj-m := hello.o |
| ccflags-y += -g | ccflags-y += -g |
| # Otherwise we were called directly from the command | # Otherwise we were called directly from the command |
| # line; invoke the kernel build system. | # line; invoke the kernel build system. |
| else | else |
| KDIR ?= /lib/modules/$(shell uname -r)/build | KDIR ?= /lib/modules/$(shell uname -r)/build |
| PWD := $(shell pwd) | PWD := $(shell pwd) |
| default: | default: |
| $(MAKE) -C $(KDIR) M=$(PWD) modules | $(MAKE) -C $(KDIR) M=$(PWD) modules |